

Cartesian Partition Manager Library
1.0.8

作成 : Doxygen 1.8.0

Wed Apr 3 2013 09:59:18

Contents

1	ネームスペース索引	1
1.1	ネームスペース一覧	1
2	構成索引	3
2.1	クラス階層	3
3	構成索引	5
3.1	構成	5
4	ファイル索引	7
4.1	ファイル一覧	7
5	ネームスペース	9
5.1	ネームスペース CES	9
5.1.1	関数	9
5.1.1.1	BaseName	9
5.1.1.2	DirName	9
5.1.1.3	OmitDots	9
5.2	ネームスペース CPM_ENDIAN	9
5.2.1	説明	10
5.2.2	列挙型	10
5.2.2.1	EMatchType	10
5.2.3	関数	10
5.2.3.1	BSWAP16	10
5.2.3.2	BSWAP32	10
5.2.3.3	BSWAP64	11
5.2.3.4	BSWAPVEC	11
5.2.3.5	DBSWAPVEC	11
5.2.3.6	SBSWAPVEC	11
5.3	ネームスペース CPM_PATH	11
5.3.1	関数	12
5.3.1.1	cpmPath_adjustDelim	12
5.3.1.2	cpmPath_concat	12

5.3.1.3	cpmPath_emitDrive	12
5.3.1.4	cpmPath_getDelimChar	12
5.3.1.5	cpmPath_hasDrive	12
5.3.1.6	cpmPath_isAbsolute	12
5.3.1.7	cpmPath_normalize	13
6	クラス	15
6.1	クラス cpm_ActiveSubdomainInfo	15
6.1.1	説明	16
6.1.2	コンストラクタとデストラクタ	16
6.1.2.1	cpm_ActiveSubdomainInfo	16
6.1.2.2	cpm_ActiveSubdomainInfo	16
6.1.2.3	~cpm_ActiveSubdomainInfo	16
6.1.3	関数	16
6.1.3.1	clear	16
6.1.3.2	GetPos	17
6.1.3.3	operator!=	17
6.1.3.4	operator==	17
6.1.3.5	SetPos	17
6.1.4	変数	18
6.1.4.1	m_pos	18
6.2	クラス cpm_Base	18
6.2.1	説明	19
6.2.2	コンストラクタとデストラクタ	19
6.2.2.1	cpm_Base	19
6.2.2.2	~cpm_Base	19
6.2.3	関数	19
6.2.3.1	cpm_strCompare	19
6.2.3.2	cpm_strCompareN	20
6.2.3.3	getCommNull	20
6.2.3.4	GetMemString	20
6.2.3.5	getRankNull	21
6.2.3.6	GetSpanTime	21
6.2.3.7	GetTime	21
6.2.3.8	GetWSpanTime	21
6.2.3.9	GetWTime	21
6.2.3.10	IsCommNull	22
6.2.3.11	IsRankNull	22
6.2.3.12	ReallIsDouble	22
6.2.3.13	VersionInfo	22

6.2.3.14	VersionInfo	23
6.3	クラス cpm_DomainInfo	23
6.3.1	説明	24
6.3.2	コンストラクタとデストラクタ	24
6.3.2.1	cpm_DomainInfo	24
6.3.2.2	~cpm_DomainInfo	24
6.3.3	関数	24
6.3.3.1	CheckData	24
6.3.3.2	clear	25
6.3.3.3	GetOrigin	25
6.3.3.4	GetPitch	25
6.3.3.5	GetRegion	25
6.3.3.6	GetVoxNum	26
6.3.3.7	SetOrigin	26
6.3.3.8	SetPitch	26
6.3.3.9	SetRegion	26
6.3.3.10	SetVoxNum	27
6.3.4	変数	27
6.3.4.1	m_origin	27
6.3.4.2	m_pitch	27
6.3.4.3	m_region	27
6.3.4.4	m_voxNum	27
6.4	クラス cpm_GlobalDomainInfo	27
6.4.1	説明	29
6.4.2	コンストラクタとデストラクタ	29
6.4.2.1	cpm_GlobalDomainInfo	29
6.4.2.2	~cpm_GlobalDomainInfo	29
6.4.3	関数	29
6.4.3.1	AddSubdomain	29
6.4.3.2	CheckData	30
6.4.3.3	clear	30
6.4.3.4	GetDivNum	30
6.4.3.5	GetSubdomainArraySize	30
6.4.3.6	GetSubdomainInfo	30
6.4.3.7	GetSubdomainNum	31
6.4.3.8	IsExistSubdomain	31
6.4.3.9	isMatchEndianSbdmMagick	31
6.4.3.10	ReadActiveSubdomainFile	32
6.4.3.11	ReadActiveSubdomainFile	32
6.4.3.12	SetDivNum	32

6.4.4	変数	33
6.4.4.1	m_divNum	33
6.4.4.2	m_subDomainInfo	33
6.5	クラス cpm_LocalDomainInfo	33
6.5.1	説明	34
6.5.2	コンストラクタとデストラクタ	34
6.5.2.1	cpm_LocalDomainInfo	34
6.5.2.2	~cpm_LocalDomainInfo	34
6.5.3	関数	34
6.5.3.1	clear	34
6.6	クラス テンプレート cpm_ObjList< T >	35
6.6.1	説明	36
6.6.2	型定義	36
6.6.2.1	DelKeyList	36
6.6.2.2	ObjectMap	36
6.6.3	コンストラクタとデストラクタ	36
6.6.3.1	cpm_ObjList	36
6.6.3.2	~cpm_ObjList	36
6.6.4	関数	36
6.6.4.1	Add	36
6.6.4.2	Create	37
6.6.4.3	Delete	37
6.6.4.4	Get	37
6.6.5	変数	37
6.6.5.1	m_DelKeyList	37
6.6.5.2	m_newKey	38
6.6.5.3	m_ObjectMap	38
6.7	クラス cpm_ParaManager	38
6.7.1	説明	44
6.7.2	コンストラクタとデストラクタ	45
6.7.2.1	cpm_ParaManager	45
6.7.2.2	~cpm_ParaManager	45
6.7.3	関数	45
6.7.3.1	Abort	45
6.7.3.2	Allgather	45
6.7.3.3	Allgather	46
6.7.3.4	Allgatherv	46
6.7.3.5	Allgatherv	46
6.7.3.6	AllocDoubleS3D	47
6.7.3.7	AllocDoubleS4D	47

6.7.3.8	AllocDoubleS4DEx	48
6.7.3.9	AllocDoubleV3D	48
6.7.3.10	AllocDoubleV3DEx	48
6.7.3.11	AllocFloatS3D	48
6.7.3.12	AllocFloatS4D	49
6.7.3.13	AllocFloatS4DEx	49
6.7.3.14	AllocFloatV3D	49
6.7.3.15	AllocFloatV3DEx	50
6.7.3.16	AllocIntS3D	50
6.7.3.17	AllocIntS4D	50
6.7.3.18	AllocIntS4DEx	51
6.7.3.19	AllocIntV3D	51
6.7.3.20	AllocIntV3DEx	51
6.7.3.21	Allreduce	52
6.7.3.22	Allreduce	52
6.7.3.23	Barrier	52
6.7.3.24	Bcast	53
6.7.3.25	Bcast	53
6.7.3.26	BndCommS3D	54
6.7.3.27	BndCommS3D	54
6.7.3.28	BndCommS3D_nowait	55
6.7.3.29	BndCommS3D_nowait	55
6.7.3.30	BndCommS4D	56
6.7.3.31	BndCommS4D	56
6.7.3.32	BndCommS4D_nowait	57
6.7.3.33	BndCommS4D_nowait	57
6.7.3.34	BndCommS4DEx	58
6.7.3.35	BndCommS4DEx	58
6.7.3.36	BndCommS4DEx_nowait	59
6.7.3.37	BndCommS4DEx_nowait	60
6.7.3.38	BndCommV3D	60
6.7.3.39	BndCommV3D	61
6.7.3.40	BndCommV3D_nowait	61
6.7.3.41	BndCommV3D_nowait	62
6.7.3.42	BndCommV3DEx	62
6.7.3.43	BndCommV3DEx	63
6.7.3.44	BndCommV3DEx_nowait	63
6.7.3.45	BndCommV3DEx_nowait	64
6.7.3.46	CalcCommSize	64
6.7.3.47	CopyArray	64

6.7.3.48	<code>cpm_BndCommS3D_nowait</code>	65
6.7.3.49	<code>cpm_BndCommS4D_nowait</code>	65
6.7.3.50	<code>cpm_BndCommS4DEx_nowait</code>	66
6.7.3.51	<code>cpm_BndCommV3D_nowait</code>	66
6.7.3.52	<code>cpm_BndCommV3DEx_nowait</code>	67
6.7.3.53	<code>cpm_Irecv</code>	67
6.7.3.54	<code>cpm_Isend</code>	68
6.7.3.55	<code>cpm_Wait</code>	68
6.7.3.56	<code>cpm_wait_BndCommS3D</code>	69
6.7.3.57	<code>cpm_wait_BndCommS4D</code>	69
6.7.3.58	<code>cpm_wait_BndCommS4DEx</code>	70
6.7.3.59	<code>cpm_wait_BndCommV3D</code>	70
6.7.3.60	<code>cpm_wait_BndCommV3DEx</code>	71
6.7.3.61	<code>cpm_Waitall</code>	72
6.7.3.62	<code>CreateProcessGroup</code>	72
6.7.3.63	<code>DecideDivPattern</code>	72
6.7.3.64	<code>FindVoxelInfo</code>	73
6.7.3.65	<code>flush</code>	73
6.7.3.66	<code>flush</code>	73
6.7.3.67	<code>Gather</code>	73
6.7.3.68	<code>Gather</code>	74
6.7.3.69	<code>Gatherv</code>	74
6.7.3.70	<code>Gatherv</code>	75
6.7.3.71	<code>get_instance</code>	75
6.7.3.72	<code>get_instance</code>	75
6.7.3.73	<code>GetBndCommBuffer</code>	76
6.7.3.74	<code>GetBndCommBufferSize</code>	76
6.7.3.75	<code>GetBndIndexExtGc</code>	76
6.7.3.76	<code>GetBndIndexExtGc</code>	77
6.7.3.77	<code>GetDivNum</code>	78
6.7.3.78	<code>GetDivPos</code>	78
6.7.3.79	<code>GetGlobalOrigin</code>	78
6.7.3.80	<code>GetGlobalRegion</code>	78
6.7.3.81	<code>GetGlobalVoxelSize</code>	79
6.7.3.82	<code>GetHostName</code>	79
6.7.3.83	<code>GetLocalOrigin</code>	79
6.7.3.84	<code>GetLocalRegion</code>	80
6.7.3.85	<code>GetLocalVoxelSize</code>	80
6.7.3.86	<code>GetMPI_Comm</code>	80
6.7.3.87	<code>GetMPI_Datatype</code>	81

6.7.3.88	GetMPI_Datatype	81
6.7.3.89	GetMPI_Op	81
6.7.3.90	GetMyRankID	82
6.7.3.91	GetNeighborRankID	82
6.7.3.92	GetNumRank	82
6.7.3.93	GetPeriodicRankID	83
6.7.3.94	GetPitch	83
6.7.3.95	GetVoxelHeadIndex	83
6.7.3.96	GetVoxelTailIndex	84
6.7.3.97	InitArray	84
6.7.3.98	Initialize	84
6.7.3.99	Initialize	85
6.7.3.100	lrecv	85
6.7.3.101	lrecv	85
6.7.3.102	lsend	86
6.7.3.103	lsend	86
6.7.3.104	IsInnerBoundary	87
6.7.3.105	IsOuterBoundary	87
6.7.3.106	IsParallel	87
6.7.3.107	IsParallel	88
6.7.3.108	packX	88
6.7.3.109	packXEx	88
6.7.3.110	packY	89
6.7.3.111	packYEx	89
6.7.3.112	packZ	90
6.7.3.113	packZEx	90
6.7.3.114	PeriodicCommS3D	91
6.7.3.115	PeriodicCommS3D	91
6.7.3.116	PeriodicCommS4D	92
6.7.3.117	PeriodicCommS4D	93
6.7.3.118	PeriodicCommS4DEx	93
6.7.3.119	PeriodicCommS4DEx	94
6.7.3.120	PeriodicCommV3D	94
6.7.3.121	PeriodicCommV3D	95
6.7.3.122	PeriodicCommV3DEx	95
6.7.3.123	PeriodicCommV3DEx	96
6.7.3.124	Recv	96
6.7.3.125	Recv	97
6.7.3.126	Send	97
6.7.3.127	Send	97

6.7.3.128	sendrecv	98
6.7.3.129	SetBndCommBuffer	98
6.7.3.130	unpackX	99
6.7.3.131	unpackXEx	99
6.7.3.132	unpackY	100
6.7.3.133	unpackYEx	100
6.7.3.134	unpackZ	101
6.7.3.135	unpackZEx	101
6.7.3.136	VoxelInit	102
6.7.3.137	VoxelInit	102
6.7.3.138	VoxelInit	103
6.7.3.139	VoxelInit_Subdomain	103
6.7.3.140	VoxelInit_Subdomain	104
6.7.3.141	Wait	104
6.7.3.142	wait_BndCommS3D	105
6.7.3.143	wait_BndCommS3D	105
6.7.3.144	wait_BndCommS4D	106
6.7.3.145	wait_BndCommS4D	106
6.7.3.146	wait_BndCommS4DEx	107
6.7.3.147	wait_BndCommS4DEx	107
6.7.3.148	wait_BndCommV3D	108
6.7.3.149	wait_BndCommV3D	108
6.7.3.150	wait_BndCommV3DEx	109
6.7.3.151	wait_BndCommV3DEx	109
6.7.3.152	Waitall	110
6.7.4	変数	110
6.7.4.1	m_bndCommInfoMap	110
6.7.4.2	m_nRank	110
6.7.4.3	m_procGrpList	110
6.7.4.4	m_rankNo	111
6.7.4.5	m_rankNoMap	111
6.7.4.6	m_reqList	111
6.7.4.7	m_voxelInfoMap	111
6.8	クラス cpm_TextParser	112
6.8.1	説明	113
6.8.2	コンストラクタとデストラクタ	113
6.8.2.1	cpm_TextParser	113
6.8.2.2	~cpm_TextParser	113
6.8.3	関数	113
6.8.3.1	Read	113

6.8.3.2	readVector	113
6.8.3.3	readVector	114
6.8.3.4	readVector	114
6.8.4	変数	115
6.8.4.1	m_tp	115
6.9	クラス <code>cpm_TextParserDomain</code>	115
6.9.1	説明	116
6.9.2	コンストラクタとデストラクタ	116
6.9.2.1	cpm_TextParserDomain	116
6.9.2.2	~cpm_TextParserDomain	116
6.9.3	関数	116
6.9.3.1	Read	116
6.9.3.2	ReadDomainInfo	117
6.9.3.3	ReadMain	117
6.9.3.4	ReadSubdomainInfo	117
6.10	クラス <code>cpm_VoxelInfo</code>	118
6.10.1	説明	120
6.10.2	コンストラクタとデストラクタ	120
6.10.2.1	cpm_VoxelInfo	120
6.10.2.2	~cpm_VoxelInfo	120
6.10.3	関数	120
6.10.3.1	CreateLocalDomainInfo	120
6.10.3.2	CreateNeighborRankInfo	120
6.10.3.3	CreateRankMap	121
6.10.3.4	GetDivNum	121
6.10.3.5	GetDivPos	121
6.10.3.6	GetGlobalOrigin	121
6.10.3.7	GetGlobalRegion	122
6.10.3.8	GetGlobalVoxelSize	122
6.10.3.9	GetLocalOrigin	122
6.10.3.10	GetLocalRegion	122
6.10.3.11	GetLocalVoxelSize	122
6.10.3.12	GetNeighborRankID	123
6.10.3.13	GetPeriodicRankID	123
6.10.3.14	GetPitch	123
6.10.3.15	GetVoxelHeadIndex	123
6.10.3.16	GetVoxelTailIndex	124
6.10.3.17	Init	124
6.10.3.18	IsInnerBoundary	124
6.10.3.19	IsOuterBoundary	125

6.10.4	フレンドと関連する関数	125
6.10.4.1	cpm_ParaManager	125
6.10.5	変数	125
6.10.5.1	m_comm	125
6.10.5.2	m_globalDomainInfo	125
6.10.5.3	m_localDomainInfo	126
6.10.5.4	m_neighborRankID	126
6.10.5.5	m_nRank	126
6.10.5.6	m_periodicRankID	126
6.10.5.7	m_rankMap	126
6.10.5.8	m_rankNo	126
6.10.5.9	m_voxelHeadIndex	126
6.10.5.10	m_voxelTailIndex	127
6.11	構造体 S_BNDCOMM_BUFFER	127
6.11.1	説明	127
6.11.2	コンストラクタとデストラクタ	127
6.11.2.1	S_BNDCOMM_BUFFER	127
6.11.2.2	~S_BNDCOMM_BUFFER	128
6.11.3	変数	128
6.11.3.1	m_bufX	128
6.11.3.2	m_bufY	128
6.11.3.3	m_bufZ	128
6.11.3.4	m_maxN	128
6.11.3.5	m_maxVC	128
6.11.3.6	m_nwX	129
6.11.3.7	m_nwY	129
6.11.3.8	m_nwZ	129
7	ファイル	131
7.1	cpm_Base.h	131
7.1.1	説明	132
7.1.2	マクロ定義	132
7.1.2.1	CPM_INLINE	132
7.2	cpm_Define.h	132
7.2.1	説明	134
7.2.2	マクロ定義	134
7.2.2.1	_IDX_S3D	134
7.2.2.2	_IDX_S4D	135
7.2.2.3	_IDX_S4DEX	135
7.2.2.4	_IDX_V3D	136

7.2.2.5	<code>_IDX_V3DEX</code>	136
7.2.2.6	<code>REAL_BUF_TYPE</code>	137
7.2.3	列挙型	137
7.2.3.1	<code>CPM_Datatype</code>	137
7.2.3.2	<code>cpm_DirFlag</code>	137
7.2.3.3	<code>cpm_ErrorCode</code>	138
7.2.3.4	<code>cpm_FaceFlag</code>	140
7.2.3.5	<code>CPM_Op</code>	140
7.2.3.6	<code>cpm_PMFlag</code>	140
7.3	<code>cpm_DomainInfo.cpp</code>	141
7.3.1	説明	141
7.4	<code>cpm_DomainInfo.h</code>	141
7.4.1	説明	142
7.5	<code>cpm_EndianUtil.h</code>	143
7.5.1	説明	144
7.6	<code>cpm_ObjList.h</code>	144
7.6.1	説明	145
7.6.2	型定義	145
7.6.2.1	<code>RankNoMap</code>	145
7.7	<code>cpm_ParaManager.cpp</code>	145
7.7.1	説明	145
7.8	<code>cpm_ParaManager.h</code>	146
7.8.1	説明	147
7.8.2	型定義	147
7.8.2.1	<code>BndCommInfoMap</code>	147
7.8.2.2	<code>RankNoMap</code>	147
7.8.2.3	<code>VoxelInfoMap</code>	147
7.9	<code>cpm_ParaManager_Alloc.cpp</code>	147
7.9.1	説明	148
7.10	<code>cpm_ParaManager_BndComm.h</code>	148
7.10.1	説明	148
7.10.2	マクロ定義	148
7.10.2.1	<code>_IDAFX</code>	148
7.10.2.2	<code>_IDXFY</code>	149
7.10.2.3	<code>_IDXFZ</code>	149
7.11	<code>cpm_ParaManager_BndCommEx.h</code>	149
7.11.1	説明	150
7.11.2	マクロ定義	150
7.11.2.1	<code>_IDAFX</code>	150
7.11.2.2	<code>_IDXFY</code>	150

7.11.2.3	_IDXFZ	150
7.12	cpm_ParaManager_frtIF.cpp	151
7.12.1	説明	154
7.12.2	マクロ定義	154
7.12.2.1	cpm_Abort_	154
7.12.2.2	cpm_Allgather_	154
7.12.2.3	cpm_Allgatherv_	154
7.12.2.4	cpm_Allreduce_	154
7.12.2.5	cpm_Barrier_	154
7.12.2.6	cpm_Bcast_	154
7.12.2.7	cpm_BndCommS3D_	154
7.12.2.8	cpm_BndCommS3D_nowait_	154
7.12.2.9	cpm_BndCommS4D_	155
7.12.2.10	cpm_BndCommS4D_nowait_	155
7.12.2.11	cpm_BndCommS4DEx_	155
7.12.2.12	cpm_BndCommS4DEx_nowait_	155
7.12.2.13	cpm_BndCommV3D_	155
7.12.2.14	cpm_BndCommV3D_nowait_	155
7.12.2.15	cpm_BndCommV3DEx_	155
7.12.2.16	cpm_BndCommV3DEx_nowait_	155
7.12.2.17	CPM_EXTERN	155
7.12.2.18	cpm_Gather_	155
7.12.2.19	cpm_Gatherv_	155
7.12.2.20	cpm_GetDivNum_	156
7.12.2.21	cpm_GetDivPos_	156
7.12.2.22	cpm_GetGlobalOrigin_	156
7.12.2.23	cpm_GetGlobalRegion_	156
7.12.2.24	cpm_GetGlobalVoxelSize_	156
7.12.2.25	cpm_GetLocalOrigin_	156
7.12.2.26	cpm_GetLocalRegion_	156
7.12.2.27	cpm_GetLocalVoxelSize_	156
7.12.2.28	cpm_GetMyRankID_	156
7.12.2.29	cpm_GetNeighborRankID_	156
7.12.2.30	cpm_GetNumRank_	156
7.12.2.31	cpm_GetPeriodicRankID_	156
7.12.2.32	cpm_GetPitch_	157
7.12.2.33	cpm_GetVoxelHeadIndex_	157
7.12.2.34	cpm_GetVoxelTailIndex_	157
7.12.2.35	cpm_Initialize_	157
7.12.2.36	cpm_Irecv_	157

7.12.2.37	<code>cpm_Isend_</code>	157
7.12.2.38	<code>cpm_IsParallel_</code>	157
7.12.2.39	<code>cpm_PeriodicCommS3D</code>	157
7.12.2.40	<code>cpm_PeriodicCommS4D</code>	157
7.12.2.41	<code>cpm_PeriodicCommS4DEx</code>	157
7.12.2.42	<code>cpm_PeriodicCommV3D</code>	157
7.12.2.43	<code>cpm_PeriodicCommV3DEx</code>	157
7.12.2.44	<code>cpm_Recv_</code>	158
7.12.2.45	<code>cpm_Send_</code>	158
7.12.2.46	<code>cpm_SetBndCommBuffer_</code>	158
7.12.2.47	<code>cpm_Voxellnit_</code>	158
7.12.2.48	<code>cpm_Voxellnit_nodiv_</code>	158
7.12.2.49	<code>cpm_Wait_</code>	158
7.12.2.50	<code>cpm_wait_BndCommsS3D_</code>	158
7.12.2.51	<code>cpm_wait_BndCommsS4D_</code>	158
7.12.2.52	<code>cpm_wait_BndCommsS4DEx_</code>	158
7.12.2.53	<code>cpm_wait_BndCommV3D_</code>	158
7.12.2.54	<code>cpm_wait_BndCommV3DEx_</code>	158
7.12.2.55	<code>cpm_Waitall_</code>	158
7.12.3	関数	159
7.12.3.1	<code>cpm_Abort_</code>	159
7.12.3.2	<code>cpm_Allgather_</code>	159
7.12.3.3	<code>cpm_Allgatherv_</code>	159
7.12.3.4	<code>cpm_Allreduce_</code>	160
7.12.3.5	<code>cpm_Barrier_</code>	160
7.12.3.6	<code>cpm_Bcast_</code>	160
7.12.3.7	<code>cpm_BndCommS3D_</code>	161
7.12.3.8	<code>cpm_BndCommS3D_nowait_</code>	161
7.12.3.9	<code>cpm_BndCommS4D_</code>	162
7.12.3.10	<code>cpm_BndCommS4D_nowait_</code>	162
7.12.3.11	<code>cpm_BndCommS4DEx_</code>	163
7.12.3.12	<code>cpm_BndCommS4DEx_nowait_</code>	163
7.12.3.13	<code>cpm_BndCommV3D_</code>	164
7.12.3.14	<code>cpm_BndCommV3D_nowait_</code>	164
7.12.3.15	<code>cpm_BndCommV3DEx_</code>	164
7.12.3.16	<code>cpm_BndCommV3DEx_nowait_</code>	165
7.12.3.17	<code>cpm_Gather_</code>	165
7.12.3.18	<code>cpm_Gatherv_</code>	166
7.12.3.19	<code>cpm_GetDivNum_</code>	166
7.12.3.20	<code>cpm_GetDivPos_</code>	167

7.12.3.21	cpm_GetGlobalOrigin_	167
7.12.3.22	cpm_GetGlobalRegion_	167
7.12.3.23	cpm_GetGlobalVoxelSize_	167
7.12.3.24	cpm_GetLocalOrigin_	168
7.12.3.25	cpm_GetLocalRegion_	168
7.12.3.26	cpm_GetLocalVoxelSize_	168
7.12.3.27	cpm_GetMyRankID_	169
7.12.3.28	cpm_GetNeighborRankID_	169
7.12.3.29	cpm_GetNumRank_	169
7.12.3.30	cpm_GetPeriodicRankID_	169
7.12.3.31	cpm_GetPitch_	170
7.12.3.32	cpm_GetVoxelHeadIndex_	170
7.12.3.33	cpm_GetVoxelTailIndex_	170
7.12.3.34	cpm_Initialize_	171
7.12.3.35	cpm_Irecv_	171
7.12.3.36	cpm_Isend_	171
7.12.3.37	cpm_IsParallel_	172
7.12.3.38	cpm_PeriodicComms3D_	172
7.12.3.39	cpm_PeriodicComms4D_	173
7.12.3.40	cpm_PeriodicComms4DEx_	173
7.12.3.41	cpm_PeriodicCommV3D_	174
7.12.3.42	cpm_PeriodicCommV3DEx_	174
7.12.3.43	cpm_Recv_	175
7.12.3.44	cpm_Send_	175
7.12.3.45	cpm_SetBndCommBuffer_	175
7.12.3.46	cpm_Voxellnit_	176
7.12.3.47	cpm_Voxellnit_nodiv_	176
7.12.3.48	cpm_Wait_	177
7.12.3.49	cpm_wait_BndComms3D_	177
7.12.3.50	cpm_wait_BndComms4D_	177
7.12.3.51	cpm_wait_BndComms4DEx_	178
7.12.3.52	cpm_wait_BndCommV3D_	178
7.12.3.53	cpm_wait_BndCommV3DEx_	179
7.12.3.54	cpm_Waitall_	179
7.13	cpm_ParaManager_inline.h	180
7.13.1	説明	180
7.14	cpm_ParaManager_MPI.cpp	180
7.14.1	説明	180
7.15	cpm_PathUtil.h	181
7.15.1	説明	182

7.16	cpm_TextParser.cpp	182
7.16.1	説明	182
7.17	cpm_TextParser.h	183
7.17.1	説明	183
7.18	cpm_TextParserDomain.cpp	184
7.18.1	説明	184
7.19	cpm_TextParserDomain.h	184
7.19.1	説明	185
7.20	cpm_Version.h	186
7.20.1	説明	186
7.20.2	マクロ定義	186
7.20.2.1	CPM_REVISION	186
7.20.2.2	CPM_VERSION_NO	186
7.21	cpm_VoxelInfo.cpp	187
7.21.1	説明	187
7.22	cpm_VoxelInfo.h	188
7.22.1	説明	188

Chapter 1

ネームスペース索引

1.1 ネームスペース一覧

ネームスペースの一覧です。

CES	9
CPM_ENDIAN	9
CPM_PATH	11

Chapter 2

構成索引

2.1 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	18
cpm_ActiveSubdomainInfo	15
cpm_LocalDomainInfo	33
cpm_DomainInfo	23
cpm_GlobalDomainInfo	27
cpm_LocalDomainInfo	33
cpm_ObjList< T >	35
cpm_ParaManager	38
cpm_TextParser	112
cpm_TextParserDomain	115
cpm_VoxelInfo	118
S_BNDCOMM_BUFFER	127

Chapter 3

構成索引

3.1 構成

クラス、構造体、共用体、インタフェースの説明です。

cpm_ActiveSubdomainInfo	15
cpm_Base	18
cpm_DomainInfo	23
cpm_GlobalDomainInfo	27
cpm_LocalDomainInfo	33
cpm_ObjList< T >	35
cpm_ParaManager	38
cpm_TextParser	112
cpm_TextParserDomain	115
cpm_VoxelInfo	118
S_BNDCOMM_BUFFER	127

Chapter 4

ファイル索引

4.1 ファイル一覧

これはファイル一覧です。

cpm_Base.h	131
cpm_Define.h	132
cpm_DomainInfo.cpp	141
cpm_DomainInfo.h	141
cpm_EndianUtil.h	143
cpm_ObjList.h	144
cpm_ParaManager.cpp	145
cpm_ParaManager.h	146
cpm_ParaManager_Alloc.cpp	147
cpm_ParaManager_BndComm.h	148
cpm_ParaManager_BndCommEx.h	149
cpm_ParaManager_frtIF.cpp	151
cpm_ParaManager_inline.h	180
cpm_ParaManager_MPI.cpp	180
cpm_PathUtil.h	181
cpm_TextParser.cpp	182
cpm_TextParser.h	183
cpm_TextParserDomain.cpp	184
cpm_TextParserDomain.h	184
cpm_Version.h	186
cpm_VoxelInfo.cpp	187
cpm_VoxelInfo.h	188

Chapter 5

ネームスペース

5.1 ネームスペース CES

関数

- `std::string DirName` (`const std::string &path`, `const char dc= '/'`)
- `std::string BaseName` (`const std::string &path`, `const std::string &suffix=std::string("")`, `const char dc= '/'`)
- `std::string OmitDots` (`const std::string &path`, `const char dc= '/'`)

5.1.1 関数

5.1.1.1 `std::string CES::BaseName (const std::string & path, const std::string & suffix = std::string(""), const char dc = ' / ') [inline]`

`cpm_PathUtil.h` の 62 行で定義されています。

5.1.1.2 `std::string CES::DirName (const std::string & path, const char dc = ' / ') [inline]`

`cpm_PathUtil.h` の 22 行で定義されています。

参照元 `cpm_TextParserDomain::ReadSubdomainInfo()`.

5.1.1.3 `std::string CES::OmitDots (const std::string & path, const char dc = ' / ') [inline]`

`cpm_PathUtil.h` の 102 行で定義されています。

参照元 `CPM_PATH::cpmPath_normalize()`.

5.2 ネームスペース CPM_ENDIAN

列挙型

- `enum EMatchType { UnKnown = 0, Match = 1, UnMatch = 2 }`

関数

- `template<class X > CPM_INLINE void BSWAP16 (X &x)`

- `template<class X >`
`CPM_INLINE void BSWAP32 (X &x)`
- `template<class X >`
`CPM_INLINE void BSWAP64 (X &x)`
- `template<class X , class Y >`
`CPM_INLINE void SBSWAPVEC (X *a, Y n)`
- `template<class X , class Y >`
`CPM_INLINE void BSWAPVEC (X *a, Y n)`
- `template<class X , class Y >`
`CPM_INLINE void DBSWAPVEC (X *a, Y n)`

5.2.1 説明

CPM のエンディアンユーティリティ名前空間

5.2.2 列挙型

5.2.2.1 enum CPM_ENDIAN::EMatchType

エンディアンチェックフラグ

列挙型の値:

Unknown 未定 (フォーマット不明)

Match 一致

UnMatch 不一致

cpm_EndianUtil.h の 110 行で定義されています。

5.2.3 関数

5.2.3.1 template<class X > CPM_INLINE void CPM_ENDIAN::BSWAP16 (X & x)

16 ビット (2 バイト) 変数のエンディアン変換

引数

<i>inout]</i>	x 変換する変数
---------------	----------

cpm_EndianUtil.h の 27 行で定義されています。

参照元 SBSWAPVEC().

5.2.3.2 template<class X > CPM_INLINE void CPM_ENDIAN::BSWAP32 (X & x)

32 ビット (4 バイト) 変数のエンディアン変換

引数

<i>inout]</i>	x 変換する変数
---------------	----------

cpm_EndianUtil.h の 38 行で定義されています。

参照元 BSWAPVEC().

5.2.3.3 `template<class X> CPM_INLINE void CPM_ENDIAN::BSWAP64 (X & x)`

64 ビット (8 バイト) 変数のエンディアン変換

引数

<i>inout</i>	x 変換する変数
--------------	----------

cpm_EndianUtil.h の 52 行で定義されています。

参照元 DBSWAPVEC().

5.2.3.4 `template<class X, class Y> CPM_INLINE void CPM_ENDIAN::BSWAPVEC (X * a, Y n)`

32 ビット (4 バイト) 変数配列のエンディアン変換

引数

	<i>inout</i>	a 変換する変数配列
in	n	配列要素数

cpm_EndianUtil.h の 87 行で定義されています。

参照先 BSWAP32().

参照元 cpm_GlobalDomainInfo::ReadActiveSubdomainFile().

5.2.3.5 `template<class X, class Y> CPM_INLINE void CPM_ENDIAN::DBSWAPVEC (X * a, Y n)`

64 ビット (8 バイト) 変数配列のエンディアン変換

引数

	<i>inout</i>	a 変換する変数配列
in	n	配列要素数

cpm_EndianUtil.h の 100 行で定義されています。

参照先 BSWAP64().

5.2.3.6 `template<class X, class Y> CPM_INLINE void CPM_ENDIAN::SBSWAPVEC (X * a, Y n)`

16 ビット (2 バイト) 変数配列のエンディアン変換

引数

	<i>inout</i>	a 変換する変数配列
in	n	配列要素数

cpm_EndianUtil.h の 73 行で定義されています。

参照先 BSWAP16().

5.3 ネームスペース CPM_PATH

関数

- char `cpmPath_getDelimChar` ()
- void `cpmPath_adjustDelim` (std::string &path)
- bool `cpmPath_hasDrive` (const std::string &path)
- std::string `cpmPath_emitDrive` (std::string &path)
- bool `cpmPath_isAbsolute` (const std::string &path)
- std::string `cpmPath_concat` (const std::string &path1, const std::string &path2)
- std::string `cpmPath_normalize` (const std::string &path)

5.3.1 関数

5.3.1.1 void CPM_PATH::cpmPath_adjustDelim (std::string & path) [inline]

cpm_PathUtil.h の 169 行で定義されています。

参照元 `cpmPath_normalize()`.

5.3.1.2 std::string CPM_PATH::cpmPath_concat (const std::string & path1, const std::string & path2) [inline]

cpm_PathUtil.h の 211 行で定義されています。

参照先 `cpmPath_getDelimChar()`.

参照元 `cpm_TextParserDomain::ReadSubdomainInfo()`.

5.3.1.3 std::string CPM_PATH::cpmPath_emitDrive (std::string & path) [inline]

cpm_PathUtil.h の 193 行で定義されています。

参照先 `cpmPath_hasDrive()`.

参照元 `cpmPath_isAbsolute()`, と `cpmPath_normalize()`.

5.3.1.4 char CPM_PATH::cpmPath_getDelimChar () [inline]

cpm_PathUtil.h の 161 行で定義されています。

参照元 `cpmPath_concat()`, `cpmPath_isAbsolute()`, と `cpmPath_normalize()`.

5.3.1.5 bool CPM_PATH::cpmPath_hasDrive (const std::string & path) [inline]

cpm_PathUtil.h の 184 行で定義されています。

参照元 `cpmPath_emitDrive()`.

5.3.1.6 bool CPM_PATH::cpmPath_isAbsolute (const std::string & path) [inline]

cpm_PathUtil.h の 201 行で定義されています。

参照先 `cpmPath_emitDrive()`, と `cpmPath_getDelimChar()`.

参照元 `cpm_TextParserDomain::ReadSubdomainInfo()`.

5.3.1.7 `std::string CPM_PATH::cpmPath_normalize (const std::string & path) [inline]`

`cpm_PathUtil.h` の 222 行で定義されています。

参照先 `cpmPath_adjustDelim()`, `cpmPath_emitDrive()`, `cpmPath_getDelimChar()`, と `CES::OmitDots()`.

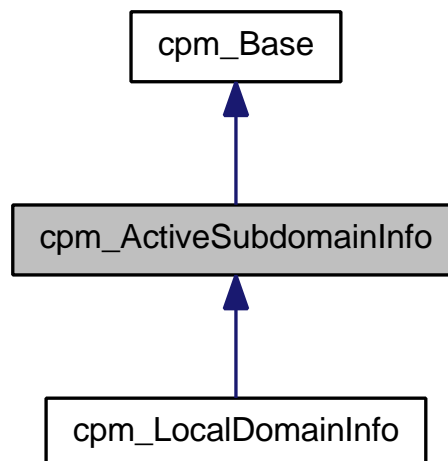
Chapter 6

クラス

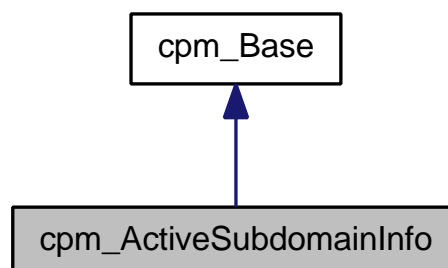
6.1 クラス cpm_ActiveSubdomainInfo

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

cpm_ActiveSubdomainInfo に対する継承グラフ



cpm_ActiveSubdomainInfo のコラボレーション図



Public メソッド

- [cpm_ActiveSubdomainInfo \(\)](#)
- [cpm_ActiveSubdomainInfo \(int pos\[3\]\)](#)

- virtual `~cpm_ActiveSubdomainInfo` ()
- virtual void `clear` ()
- void `SetPos` (int pos[3])
- const int * `GetPos` () const
- bool `operator==` (`cpm_ActiveSubdomainInfo` dom)
- bool `operator!=` (`cpm_ActiveSubdomainInfo` dom)

Private 変数

- int `m_pos` [3]
領域分割内での位置

6.1.1 説明

CPM のサブ領域情報クラス

`cpm_DomainInfo.h` の 103 行で定義されています。

6.1.2 コンストラクタとデストラクタ

6.1.2.1 `cpm_ActiveSubdomainInfo::cpm_ActiveSubdomainInfo` ()

デフォルトコンストラクタ

`cpm_DomainInfo.cpp` の 134 行で定義されています。

参照先 `clear()`.

6.1.2.2 `cpm_ActiveSubdomainInfo::cpm_ActiveSubdomainInfo` (int pos[3])

コンストラクタ

引数

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

`cpm_DomainInfo.cpp` の 142 行で定義されています。

参照先 `SetPos()`.

6.1.2.3 `cpm_ActiveSubdomainInfo::~cpm_ActiveSubdomainInfo` () [virtual]

デストラクタ

`cpm_DomainInfo.cpp` の 150 行で定義されています。

6.1.3 関数

6.1.3.1 `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 157 行で定義されています。

参照先 `m_pos`.

参照元 `cpm_ActiveSubdomainInfo()`.

6.1.3.2 `const int * cpm_ActiveSubdomainInfo::GetPos () const`

位置の取得

戻り値

位置情報整数配列のポインタ

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

参照先 `m_pos`.

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

6.1.3.3 `bool cpm_ActiveSubdomainInfo::operator!= (cpm_ActiveSubdomainInfo dom)`

比較演算子

引数

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

戻り値

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

`cpm_DomainInfo.cpp` の 196 行で定義されています。

参照先 `m_pos`.

6.1.3.4 `bool cpm_ActiveSubdomainInfo::operator==(cpm_ActiveSubdomainInfo dom)`

比較演算子

引数

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

戻り値

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

`cpm_DomainInfo.cpp` の 185 行で定義されています。

参照先 `m_pos`.

6.1.3.5 `void cpm_ActiveSubdomainInfo::SetPos (int pos[3])`

位置のセット

引数

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

cpm_DomainInfo.cpp の 167 行で定義されています。

参照先 `m_pos`.

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

6.1.4 変数

6.1.4.1 `int cpm_ActiveSubdomainInfo::m_pos[3]` `[private]`

領域分割内での位置

cpm_DomainInfo.h の 158 行で定義されています。

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

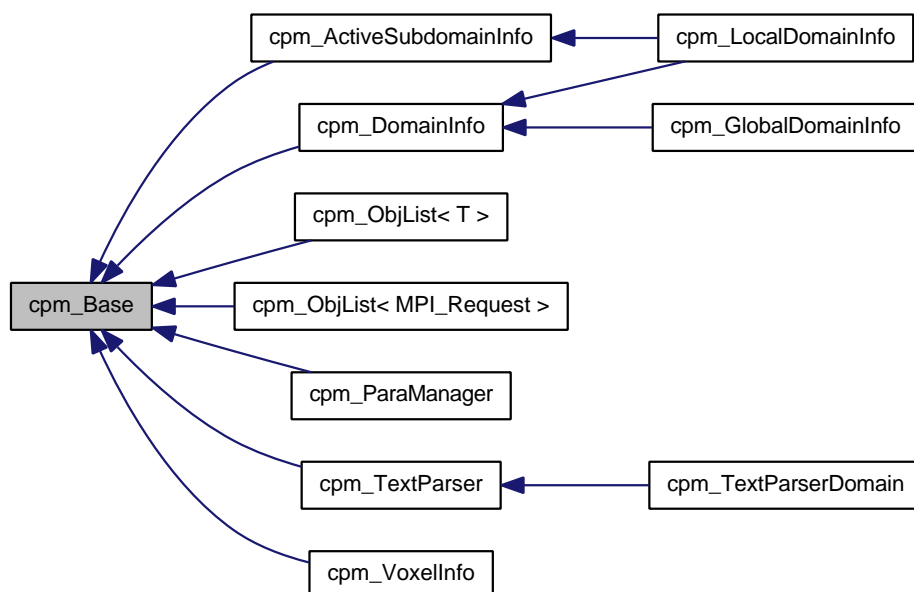
このクラスの説明は次のファイルから生成されました:

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

6.2 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` ()

6.2.1 説明

CPM のベースクラス

`cpm_Base.h` の 45 行で定義されています。

6.2.2 コンストラクタとデストラクタ

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

コンストラクタ

`cpm_Base.h` の 241 行で定義されています。

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

デストラクタ

`cpm_Base.h` の 244 行で定義されています。

6.2.3 関数

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

文字列の比較

引数

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

戻り値

0	一致する
0 以外	一致しない

cpm_Base.h の 207 行で定義されています。

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

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

文字列の比較 (文字数指定)

引数

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

戻り値

0	一致する
0 以外	一致しない

cpm_Base.h の 229 行で定義されています。

参照先 cpm_strCompare().

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

NULL の MPI_Comm を取得

戻り値

NULL の MPI_Comm

cpm_Base.h の 72 行で定義されています。

参照元 cpm_ParaManager::GetMPI_Comm().

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

メモリ量の文字列を返す

引数

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

戻り値

メモリ量の文字列

cpm_Base.h の 148 行で定義されています。

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

NULL のランク番号を取得

戻り値

NULL のランク番号

cpm_Base.h の 54 行で定義されています。

参照元 cpm_VoxelInfo::cpm_VoxelInfo(), cpm_VoxelInfo::CreateNeighborRankInfo(), cpm_VoxelInfo::CreateRankMap(), cpm_ParaManager::GetMyRankID(), cpm_ParaManager::PeriodicCommsS4D(), と cpm_ParaManager::PeriodicCommsS4DEx().

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

経過時刻の取得 (gettimeofday 版)

引数

in	before	計測開始時刻
----	--------	--------

戻り値

計測開始時刻からの経過時刻

cpm_Base.h の 119 行で定義されています。

参照先 GetTime().

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

時刻の取得 (gettimeofday 版) 時刻

cpm_Base.h の 104 行で定義されています。

参照元 GetSpanTime().

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

経過時刻の取得 (MPI_Wtime 版)

引数

in	before	計測開始時刻
----	--------	--------

戻り値

計測開始時刻からの経過時刻

cpm_Base.h の 138 行で定義されています。

参照先 GetWTime().

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

時刻の取得 (MPI_Wtime 版) 時刻

cpm_Base.h の 128 行で定義されています。

参照元 GetWSpanTime().

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

NULL の MPI_Comm かどうかを確認

戻り値

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

cpm_Base.h の 81 行で定義されています。

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

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

NULL のランクかどうかを確認

戻り値

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

cpm_Base.h の 63 行で定義されています。

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

6.2.3.12 `static CPM_INLINE bool cpm_Base::RealsIsDouble () [inline, static]`

fortan の実数型 (CPM_REAL) が倍精度かどうか確認

戻り値

<i>true</i>	倍精度
<i>false</i>	単精度

cpm_Base.h の 91 行で定義されています。

参照元 cpm_ParaManager::GetMPI_Datatype().

6.2.3.13 `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 182 行で定義されています。

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

バージョンを出力する

引数

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

cpm_Base.h の 191 行で定義されています。

参照先 CPM_VERSION_NO.

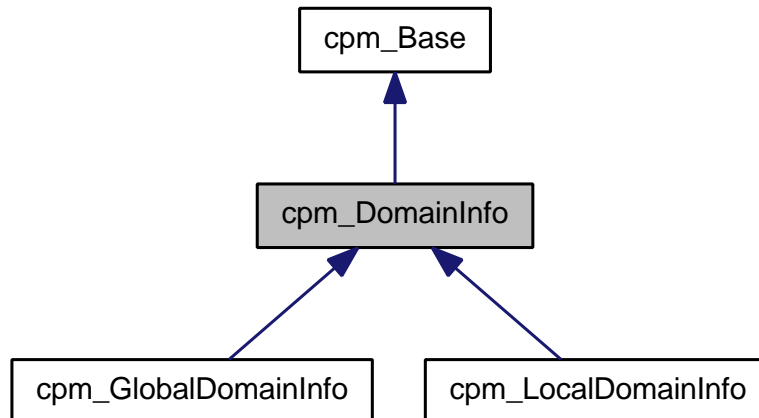
このクラスの説明は次のファイルから生成されました:

- [cpm_Base.h](#)

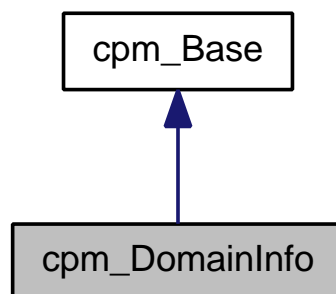
6.3 クラス cpm_DomainInfo

```
#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 (double org[3])`
- `const double * GetOrigin () const`
- `void SetPitch (double pch[3])`
- `const double * GetPitch () const`
- `void SetRegion (double rgn[3])`
- `const double * GetRegion () const`
- `void SetVoxNum (int vox[3])`
- `const int * GetVoxNum () const`
- `cpm_ErrorCode CheckData ()`

Private 変数

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

6.3.1 説明

CPM の領域情報クラス

`cpm_DomainInfo.h` の 23 行で定義されています。

6.3.2 コンストラクタとデストラクタ

6.3.2.1 `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()`.

6.3.2.2 `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。

6.3.3 関数

6.3.3.1 `cpm_ErrorCode cpm_DomainInfo::CheckData ()`

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_DomainInfo.cpp` の 118 行で定義されています。

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

参照元 `cpm_GlobalDomainInfo::CheckData()`.

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

情報のクリア

[cpm_LocalDomainInfo](#), と [cpm_GlobalDomainInfo](#) で再定義されています。

`cpm_DomainInfo.cpp` の 33 行で定義されています。

参照先 `m_origin`, `m_pitch`, `m_region`, と `m_voxNum`.

参照元 `cpm_DomainInfo()`.

6.3.3.3 `const double * cpm_DomainInfo::GetOrigin () const`

原点の取得

戻り値

原点情報実数配列のポインタ

`cpm_DomainInfo.cpp` の 57 行で定義されています。

参照先 `m_origin`.

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

6.3.3.4 `const double * cpm_DomainInfo::GetPitch () const`

ピッチの取得

戻り値

ピッチ情報実数配列のポインタ

`cpm_DomainInfo.cpp` の 75 行で定義されています。

参照先 `m_pitch`.

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

6.3.3.5 `const double * cpm_DomainInfo::GetRegion () const`

空間サイズの取得

戻り値

空間サイズ情報実数配列のポインタ

`cpm_DomainInfo.cpp` の 93 行で定義されています。

参照先 `m_region`.

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

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

VOXEL 数の取得

戻り値

VOXEL 数情報実数配列のポインタ

cpm_DomainInfo.cpp の 111 行で定義されています。

参照先 m_voxNum.

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

6.3.3.7 `void cpm_DomainInfo::SetOrigin (double 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。

参照先 m_origin.

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

6.3.3.8 `void cpm_DomainInfo::SetPitch (double 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。

参照先 m_pitch.

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

6.3.3.9 `void cpm_DomainInfo::SetRegion (double 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。

参照先 m_region.

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

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

VOXEL 数のセット

引数

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

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

参照先 `m_voxNum`.

参照元 `cpm_VoxellInfo::CreateLocalDomainInfo()`, `cpm_TextParserDomain::ReadDomainInfo()`, `cpm_ParaManager::VoxellInit()`, と `cpm_ParaManager::Voxellnit_Subdomain()`.

6.3.4 変数

6.3.4.1 `double cpm_DomainInfo::m_origin[3] [private]`

原点

`cpm_DomainInfo.h` の 94 行で定義されています。

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

6.3.4.2 `double cpm_DomainInfo::m_pitch[3] [private]`

ピッチ

`cpm_DomainInfo.h` の 96 行で定義されています。

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

6.3.4.3 `double cpm_DomainInfo::m_region[3] [private]`

空間サイズ

`cpm_DomainInfo.h` の 95 行で定義されています。

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

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

VOXEL 数

`cpm_DomainInfo.h` の 97 行で定義されています。

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

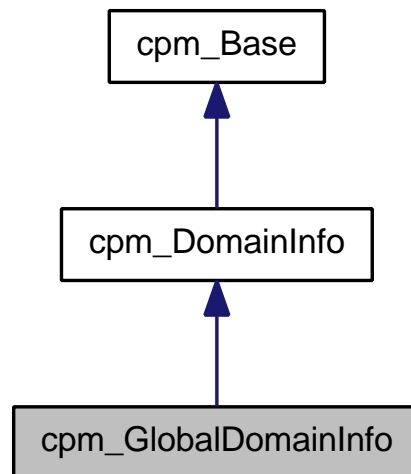
このクラスの説明は次のファイルから生成されました:

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

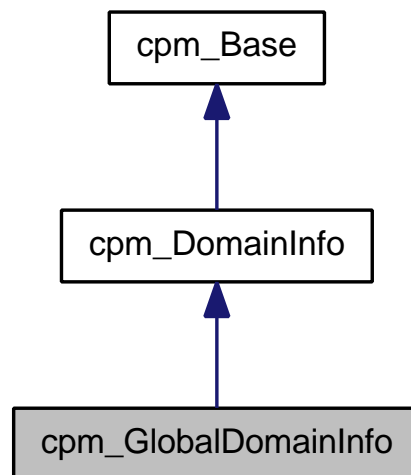
6.4 クラス `cpm_GlobalDomainInfo`

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

cpm_GlobalDomainInfo に対する継承グラフ



cpm_GlobalDomainInfo のコラボレーション図



Public メソッド

- [cpm_GlobalDomainInfo \(\)](#)
- virtual [~cpm_GlobalDomainInfo \(\)](#)
- virtual void [clear \(\)](#)
- void [SetDivNum \(int div\[3\]\)](#)
- const int * [GetDivNum \(\)](#) const
- bool [IsExistSubdomain \(cpm_ActiveSubdomainInfo subDomain\)](#)
- bool [AddSubdomain \(cpm_ActiveSubdomainInfo subDomain\)](#)
- int [GetSubdomainNum \(\)](#) const
- int [GetSubdomainArraySize \(\)](#) const
- const [cpm_ActiveSubdomainInfo](#) * [GetSubdomainInfo \(size_t idx\)](#) const
- [cpm_ErrorCode](#) [CheckData \(int nRank\)](#)
- [cpm_ErrorCode](#) [ReadActiveSubdomainFile \(std::string subDomainFile\)](#)

Static Public メソッド

- static `cpm_ErrorCode ReadActiveSubdomainFile` (`std::string subDomainFile`, `std::vector< cpm_ActiveSubdomainInfo > &subDomainInfo`, `int div[3]`)
- static `CPM_ENDIAN::EMatchType isMatchEndianSbdmMagick` (`int ident`)

Private 変数

- `int m_divNum [3]`
領域分割数
- `std::vector< cpm_ActiveSubdomainInfo > m_subDomainInfo`
活性サブドメイン情報

6.4.1 説明

CPM の全体領域情報クラス

`cpm_DomainInfo.h` の 164 行で定義されています。

6.4.2 コンストラクタとデストラクタ

6.4.2.1 `cpm_GlobalDomainInfo::cpm_GlobalDomainInfo ()`

コンストラクタ

`cpm_DomainInfo.cpp` の 206 行で定義されています。

参照先 `clear()`.

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

デストラクタ

`cpm_DomainInfo.cpp` の 214 行で定義されています。

6.4.3 関数

6.4.3.1 `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 265 行で定義されています。

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

参照元 `CheckData()`, と `cpm_ParaManager::Voxellnit_Subdomain()`.

6.4.3.2 `cpm_ErrorCode cpm_GlobalDomainInfo::CheckData (int nRank)`

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

引数

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

戻り値

終了コード (`CPM_SUCCESS`=正常終了)

`cpm_DomainInfo.cpp` の 310 行で定義されています。

参照先 `AddSubdomain()`, `cpm_DomainInfo::CheckData()`, `CPM_ERROR_INVALID_DIVNUM`, `CPM_ERROR_MISMATCH_NP_SUBDOMAIN`, `CPM_SUCCESS`, `m_divNum`, と `m_subDomainInfo`.

参照元 `cpm_ParaManager::Voxellnit()`.

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

情報のクリア

`cpm_DomainInfo` を再定義しています。

`cpm_DomainInfo.cpp` の 221 行で定義されています。

参照先 `m_divNum`, と `m_subDomainInfo`.

参照元 `cpm_GlobalDomainInfo()`.

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

領域分割数の取得

戻り値

領域分割数整数配列のポインタ

`cpm_DomainInfo.cpp` の 244 行で定義されています。

参照先 `m_divNum`.

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

6.4.3.5 `int cpm_GlobalDomainInfo::GetSubdomainArraySize () const`

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

`cpm_DomainInfo.cpp` の 294 行で定義されています。

参照先 `m_subDomainInfo`.

6.4.3.6 `const cpm_ActiveSubdomainInfo * cpm_GlobalDomainInfo::GetSubdomainInfo (size_t idx) const`

活性サブドメイン情報を取得

引数

<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 302 行で定義されています。

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

参照元 `cpm_VoxelInfo::CreateRankMap()`.

6.4.3.7 `int cpm_GlobalDomainInfo::GetSubdomainNum () const`

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

戻り値

活性サブドメインの数

`cpm_DomainInfo.cpp` の 278 行で定義されています。

参照先 `m_divNum`, と `m_subDomainInfo`.

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

6.4.3.8 `bool cpm_GlobalDomainInfo::IsExistSubdomain (cpm_ActiveSubdomainInfo subDomain)`

活性サブドメイン情報の存在チェック

引数

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

戻り値

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

`cpm_DomainInfo.cpp` の 252 行で定義されています。

参照先 `m_subDomainInfo`.

参照元 `AddSubdomain()`.

6.4.3.9 `CPM_ENDIAN::EMatchType cpm_GlobalDomainInfo::isMatchEndianSbdmMagick (int ident)` [static]

ActiveSubdomain ファイルのエンディアンチェック ActiveSubdomain ファイルのエンディアンをチェック

引数

<code>in</code>	<code>ident</code>	ActiveSubdomain ファイルのIdentifier
-----------------	--------------------	---------------------------------

戻り値

CPM_ENDIAN::Match	一致
CPM_ENDIAN::UnMatch	不一致
CPM_ENDIAN::UnKnown	フォーマットが異なる

cpm_DomainInfo.cpp の 357 行で定義されています。

参照先 CPM_ENDIAN::Match, CPM_ENDIAN::UnKnown, と CPM_ENDIAN::UnMatch.

6.4.3.10 cpm_ErrorCode cpm_GlobalDomainInfo::ReadActiveSubdomainFile (std::string subDomainFile)

ActiveSubdomain ファイルの読み込み ActiveSubdomain ファイルを読み込み、活性ドメイン情報を生成する

引数

in	subDomainFile	ActiveSubdomain ファイル名
----	---------------	-----------------------

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_DomainInfo.cpp の 384 行で定義されています。

参照先 CPM_ERROR_MISMATCH_DIV_SUBDOMAIN, と CPM_SUCCESS.

参照元 cpm_TextParserDomain::ReadSubdomainInfo(), と cpm_ParaManager::Voxellnit_Subdomain().

6.4.3.11 cpm_ErrorCode cpm_GlobalDomainInfo::ReadActiveSubdomainFile (std::string subDomainFile, std::vector< cpm_ActiveSubdomainInfo > & subDomainInfo, int div[3]) [static]

ActiveSubdomain ファイルの読み込み (static 関数) ActiveSubdomain ファイルを読み込み、活性ドメイン情報を生成する

引数

in	subDomainFile	ActiveSubdomain ファイル名
out	subDomainInfo	活性ドメイン情報
out	div	ActiveSubdiomain ファイル中の領域分割数

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_DomainInfo.cpp の 412 行で定義されています。

参照先 CPM_ENDIAN::BSWAPVEC(), CPM_ERROR_OPEN_SBDM, CPM_ERROR_READ_SBDM_CONTENTS, CPM_ERROR_READ_SBDM_DIV, CPM_ERROR_READ_SBDM_FORMAT, CPM_ERROR_READ_SBDM_HEADER, CPM_ERROR_SBDM_NUMDOMAIN_ZERO, CPM_SUCCESS, CPM_ENDIAN::UnKnown, と CPM_ENDIAN::UnMatch.

6.4.3.12 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 234 行で定義されています。

参照先 m_divNum.

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

6.4.4 変数

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

領域分割数

cpm_DomainInfo.h の 267 行で定義されています。

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

6.4.4.2 std::vector<cpm_ActiveSubdomainInfo> cpm_GlobalDomainInfo::m_subDomainInfo [private]

活性サブドメイン情報

cpm_DomainInfo.h の 268 行で定義されています。

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

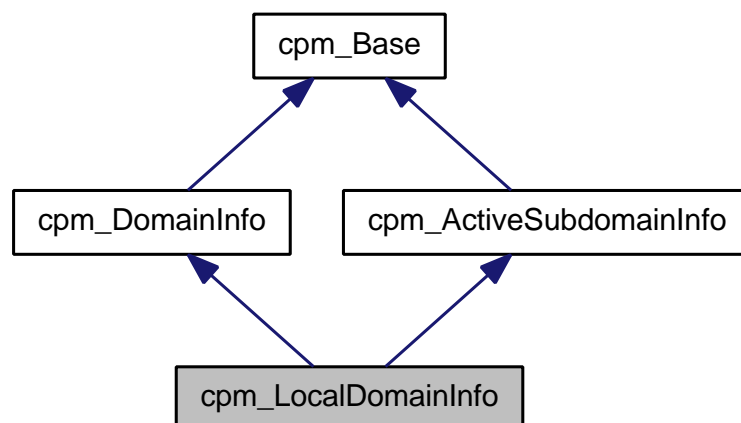
このクラスの説明は次のファイルから生成されました:

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

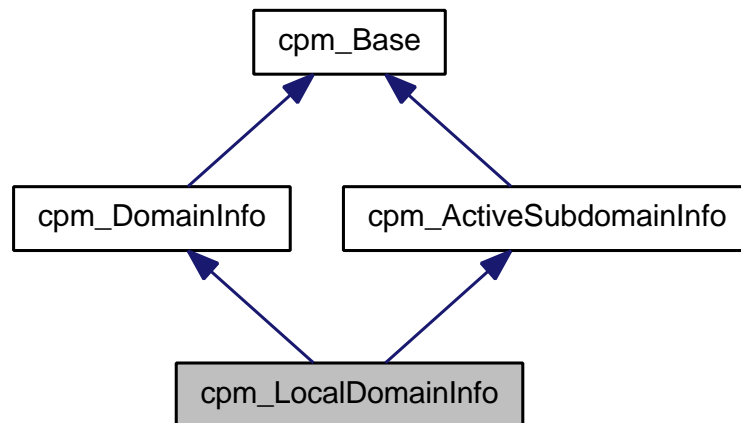
6.5 クラス cpm_LocalDomainInfo

```
#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 \(\)](#)

6.5.1 説明

CPM のローカル領域情報クラス

cpm_DomainInfo.h の 274 行で定義されています。

6.5.2 コンストラクタとデストラクタ

6.5.2.1 cpm_LocalDomainInfo::cpm_LocalDomainInfo ()

コンストラクタ

cpm_DomainInfo.cpp の 497 行で定義されています。

6.5.2.2 cpm_LocalDomainInfo::~~cpm_LocalDomainInfo () [virtual]

デストラクタ

cpm_DomainInfo.cpp の 504 行で定義されています。

6.5.3 関数

6.5.3.1 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 511 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:

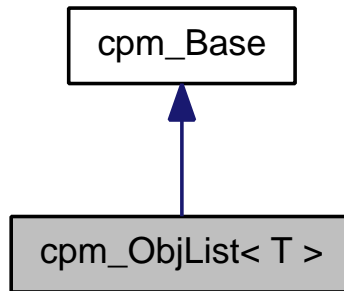
- [cpm_DomainInfo.h](#)

- [cpm_DomainInfo.cpp](#)

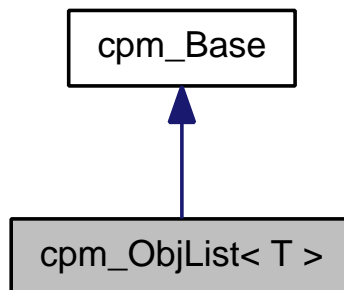
6.6 クラス テンプレート `cpm_ObjList< T >`

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

`cpm_ObjList< T >` に対する継承グラフ



`cpm_ObjList< T >` のコラボレーション図



Public メソッド

- [cpm_ObjList \(\)](#)
- [~cpm_ObjList \(\)](#)
- [T * Create \(\)](#)
- [int Add \(T *obj\)](#)
- [cpm_ErrorCode Delete \(int key\)](#)
- [T * Get \(int key\)](#)

Private 型

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

Private 変数

- [ObjectMap m_ObjectMap](#)
- [DelKeyList m_DelKeyList](#)
- [int m_newKey](#)

6.6.1 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。

6.6.2 型定義

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

削除済み登録番号のリスト

cpm_ObjList.h の 43 行で定義されています。

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

オブジェクトのマップ

cpm_ObjList.h の 39 行で定義されています。

6.6.3 コンストラクタとデストラクタ

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

コンストラクタ

cpm_ObjList.h の 56 行で定義されています。

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

デストラクタ

cpm_ObjList.h の 64 行で定義されています。

6.6.4 関数

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

オブジェクトの追加

引数

in	obj	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().

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

オブジェクトの生成 デフォルトコンストラクタが必要

戻り値

生成したオブジェクトのポインタ

`cpm_ObjList.h` の 79 行で定義されています。

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

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

オブジェクトの削除

引数

<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()`.

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

オブジェクトの取得

引数

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

6.6.5 変数

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

`cpm_ObjList.h` の 44 行で定義されています。

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

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

使用可能な登録番号

`cpm_ObjList.h` の 47 行で定義されています。

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

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

`cpm_ObjList.h` の 40 行で定義されています。

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

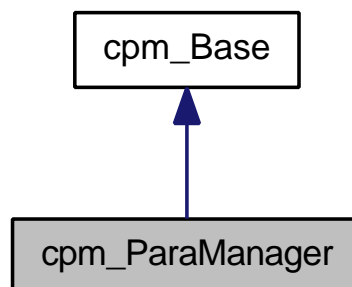
このクラスの説明は次のファイルから生成されました:

- [cpm_ObjList.h](#)

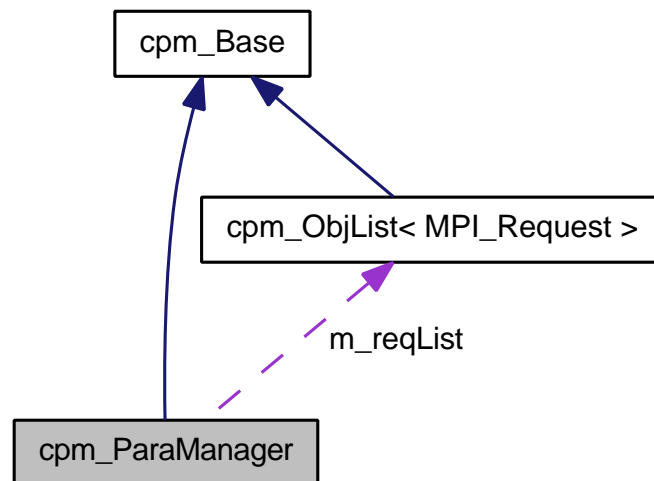
6.7 クラス `cpm_ParaManager`

`#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\[3\], double origin\[3\], double region\[3\], size_t maxVC=1, size_t maxN=3, int procGrpNo=0\)](#)
- [cpm_ErrorCode Voxellnit \(int vox\[3\], double origin\[3\], double region\[3\], size_t maxVC=1, size_t maxN=3, int procGrpNo=0\)](#)
- [cpm_ErrorCode Voxellnit_Subdomain \(int div\[3\], int vox\[3\], double origin\[3\], double region\[3\], std::string subDomainFile, size_t maxVC=1, size_t maxN=3, int procGrpNo=0\)](#)
- [cpm_ErrorCode Voxellnit_Subdomain \(int vox\[3\], double origin\[3\], double region\[3\], std::string subDomainFile, 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 double * GetPitch \(int procGrpNo=0\)](#)
- [const int * GetGlobalVoxelSize \(int procGrpNo=0\)](#)
- [const double * GetGlobalOrigin \(int procGrpNo=0\)](#)
- [const double * GetGlobalRegion \(int procGrpNo=0\)](#)
- [const int * GetLocalVoxelSize \(int procGrpNo=0\)](#)
- [const double * GetLocalOrigin \(int procGrpNo=0\)](#)
- [const double * GetLocalRegion \(int procGrpNo=0\)](#)
- [const int * GetDivPos \(int procGrpNo=0\)](#)
- [const int * GetVoxelHeadIndex \(int procGrpNo=0\)](#)
- [const int * GetVoxelTailIndex \(int procGrpNo=0\)](#)
- [const int * GetNeighborRankID \(int procGrpNo=0\)](#)
- [const int * GetPeriodicRankID \(int procGrpNo=0\)](#)
- [bool GetBndIndexExtGc \(int id, int *array, int vc, int &ista, int &jsta, int &ksta, int &ilen, int &jlen, int &klen, int procGrpNo=0\)](#)
- [bool GetBndIndexExtGc \(int id, int *array, int imax, int jmax, int kmax, int vc, int &ista, int &jsta, int &ksta, int &ilen, int &jlen, int &klen, int procGrpNo=0\)](#)
- [bool IsOuterBoundary \(cpm_FaceFlag face, int procGrpNo=0\)](#)

- bool `IsInnerBoundary` (`cpm_FaceFlag` face, int procGrpNo=0)
- int `GetMyRankID` (int procGrpNo=0)
- int `GetNumRank` (int procGrpNo=0)
- std::string `GetHostName` ()
- 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_BndCommsS3D_nowait` (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)

- `cpm_ErrorCode cpm_BndCommV3D_nowait` (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- `cpm_ErrorCode cpm_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 BndComms4D` (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo=0)
- `cpm_ErrorCode BndComms4D` (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 BndComms3D_nowait` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode 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)
- `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 BndComms4D_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 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)
- `template<class T >`
`CPM_INLINE cpm_ErrorCode wait_BndCommS3D` (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- `cpm_ErrorCode wait_BndCommS3D` (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)

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

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

Static Public メソッド

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

Private メソッド

- `cpm_ParaManager` ()
- `virtual ~cpm_ParaManager` ()
- `cpm_ErrorCode DecideDivPattern` (`int` `divNum`, `int` `voxSize[3]`, `int` `divPttn[3]`) `const`
- `unsigned long long CalcCommSize` (`unsigned long long` `iDiv`, `unsigned long long` `jDiv`, `unsigned long long` `kDiv`, `unsigned long long` `voxsize[3]`) `const`
- `CPM_INLINE S_BNDCOMM_BUFFER * GetBndCommBuffer` (`int` `procGrpNo=0`)
- `template<class T >`
`cpm_ErrorCode packX` (`T *array`, `int` `imax`, `int` `jmax`, `int` `kmax`, `int` `nmax`, `int` `vc`, `int` `vc_comm`, `T *sendm`, `T *sendp`, `int` `nIDm`, `int` `nIDp`)
- `template<class T >`
`cpm_ErrorCode unpackX` (`T *array`, `int` `imax`, `int` `jmax`, `int` `kmax`, `int` `nmax`, `int` `vc`, `int` `vc_comm`, `T *recvm`, `T *recvp`, `int` `nIDm`, `int` `nIDp`)

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

Private 変数

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

6.7.1 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。

6.7.2 コンストラクタとデストラクタ

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

コンストラクタ

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

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

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

デストラクタ

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

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

6.7.3 関数

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

Abort

- `MPI_Abort` のインターフェイス

引数

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

`cpm_ParaManager_MPI.cpp` の 161 行で定義されています。

参照元 `cpm_Abort_()`, と `VoxelInit()`.

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

Allgather

- `MPI_Allgather` のインターフェイス

引数

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

戻り値

終了コード (`CPM_SUCCESS`=正常終了)

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

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

参照元 `cpm_Allgather_()`.

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

Allgather

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 447 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().

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

Allgather

- MPI_Allgather のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Allgather_().

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

Allgather

- MPI_Allgather のインターフェイス

- `MPI_Datatype` を指定するバージョン

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_MPI.cpp` の 506 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()`.

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

配列確保 `double(imax,jmax,kmax)`

引数

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

戻り値

配列ポインタ

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

参照先 `AllocDoubleS4D()`.

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

配列確保 `double(imax,jmax,kmax,nmax)`

引数

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

戻り値

配列ポインタ

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

参照先 `GetLocalVoxelSize()`.

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

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

配列確保 `double(nmax,imax,jmax,kmax)`

引数

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

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 128 行で定義されています。

参照先 `AllocDoubleS4D()`.

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

配列確保 `double(imax,jmax,kmax,3)`

引数

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

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 80 行で定義されています。

参照先 `AllocDoubleS4D()`.

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

配列確保 `double(3,imax,jmax,kmax)`

引数

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

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 104 行で定義されています。

参照先 `AllocDoubleS4D()`.

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

配列確保 `float(imax,jmax,kmax)`

引数

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

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 64 行で定義されています。

参照先 `AllocFloatS4D()`.

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

配列確保 `float(imax,jmax,kmax,nmax)`

引数

<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 32 行で定義されています。

参照先 `GetLocalVoxelSize()`.

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

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

配列確保 `float(nmax,imax,jmax,kmax)`

引数

<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 136 行で定義されています。

参照先 `AllocFloatS4D()`.

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

配列確保 `float(imax,jmax,kmax,3)`

引数

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

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 88 行で定義されています。

参照先 AllocFloatS4D().

6.7.3.15 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 112 行で定義されています。

参照先 AllocFloatS4D().

6.7.3.16 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 72 行で定義されています。

参照先 AllocIntS4D().

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

配列確保 int(imax,jmax,kmax,nmax)

引数

in	vc	仮想セル数
in	procGrpNo	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 `GetLocalVoxelSize()`.

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

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

配列確保 `int(nmax,imax,jmax,kmax)`

引数

<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 144 行で定義されています。

参照先 `AllocIntS4D()`.

6.7.3.19 `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 96 行で定義されています。

参照先 `AllocIntS4D()`.

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

配列確保 `int(3,imax,jmax,kmax)`

引数

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

戻り値

配列ポインタ

`cpm_ParaManager_Alloc.cpp` の 120 行で定義されています。

参照先 `AllocIntS4D()`.

6.7.3.21 `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	<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_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_(), と GetBndIndexExtGc().

6.7.3.22 `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 391 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().

6.7.3.23 `cpm_ErrorCode cpm_ParaManager::Barrier (int procGrpNo = 0)`

Barrier

- MPI_Barrier のインターフェイス

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

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 171 行で定義されています。

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

参照元 cpm_Barrier_(), と Initialize().

6.7.3.24 `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_().

6.7.3.25 `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 249 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().

6.7.3.26 `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_().

6.7.3.27 `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 535 行で定義されています。

参照先 BndCommS4D().

6.7.3.28 `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().

6.7.3.29 `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 597 行で定義されています。

参照先 BndCommS4D_nowait().

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

袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

参照元 BndCommS3D(), BndCommS4D(), BndCommV3D(), と cpm_BndCommS4D().

6.7.3.31 `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 553 行で定義されています。

参照先 BndCommS4D(), と CPM_ERROR_MPI_INVALID_DATATYPE.

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

6.7.3.33 `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 615 行で定義されています。

参照先 BndCommsS4D_nowait(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.7.3.34 `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, GetBndCommBuffer(), GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, packXEx(), packYEx(), packZEx(), sendrecv(), unpackXEx(), unpackYEx(), unpackZEx(), Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

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

6.7.3.35 `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 792 行で定義されています。

参照先 BndCommS4DEx(), と CPM_ERROR_MPI_INVALID_DATATYPE.

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

非同期版袖通信 (Scalar4DEx 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

6.7.3.37 `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 845 行で定義されています。

参照先 BndCommS4DEx_nowait(), と CPM_ERROR_MPI_INVALID_DATATYPE.

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

袖通信 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BndCommS4D().

参照元 cpm_BndCommV3D_().

6.7.3.39 `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 544 行で定義されています。

参照先 BndCommS4D().

6.7.3.40 `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().

6.7.3.41 `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 606 行で定義されています。

参照先 BndComms4D_nowait().

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

袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BndComms4DEx().

参照元 cpm_BndCommV3DEx_().

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 783 行で定義されています。

参照先 BndCommS4DEx().

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_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().

6.7.3.45 `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 836 行で定義されています。

参照先 BndCommsS4DEx_nowait().

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

I,J,K 分割を行った時の通信点数の総数を取得する

引数

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

戻り値

袖通信点数

cpm_ParaManager.cpp の 569 行で定義されています。

参照元 DecideDivPattern().

6.7.3.47 `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。

6.7.3.48 `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 (<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 2647 行で定義されています。

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

参照元 `cpm_BndCommS3D_nowait()`.

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

`cpm_BndCommS4D_nowait`

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

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (<code>cpm_fparam.fi</code> 参照)

out	<i>reqNo</i>	リクエスト番号配列 (サイズ 12)
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2731 行で定義されています。

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

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

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

cpm_BndCommS4DEx_nowait

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

引数

in	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (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 2945 行で定義されています。

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

参照元 cpm_BndCommS4DEx_nowait_(), と cpm_BndCommV3DEx_nowait().

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

cpm_BndCommV3D_nowait

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 2689 行で定義されています。

参照先 `cpm_ObjList< T >::Add()`, `BndCommV3D_nowait()`, `cpm_BndCommsS4D_nowait()`, `CPM_ERROR_MPI_INVALID_DATATYPE`, `CPM_ERROR_REGIST_OBJKEY`, `CPM_SUCCESS`, `cpm_ObjList< T >::Create()`, `GetMPI_Datatype()`, と `m_reqList`.

参照元 `cpm_BndCommV3D_nowait()`.

6.7.3.52 `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 2903 行で定義されています。

参照先 `cpm_ObjList< T >::Add()`, `BndCommV3DEx_nowait()`, `cpm_BndCommsS4DEx_nowait()`, `CPM_ERROR_MPI_INVALID_DATATYPE`, `CPM_ERROR_REGIST_OBJKEY`, `CPM_SUCCESS`, `cpm_ObjList< T >::Create()`, `GetMPI_Datatype()`, と `m_reqList`.

参照元 `cpm_BndCommV3DEx_nowait()`.

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

`cpm_Irecv`

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2615 行で定義されています。

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

参照元 cpm_Irecv_().

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

cpm_Isend

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2578 行で定義されています。

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

参照元 cpm_Isend_().

6.7.3.55 `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	reqNo	リクエスト番号
----	-------	---------

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2514 行で定義されています。

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

参照元 cpm_Wait().

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

cpm_wait_BndCommS3D

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2769 行で定義されています。

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

参照元 cpm_wait_BndCommS3D().

6.7.3.57 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	<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 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 2861 行で定義されています。

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

参照元 cpm_wait_BndCommS3D(), cpm_wait_BndCommS4D_(), と cpm_wait_BndCommV3D().

6.7.3.58 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	<i>array</i>	袖通信をする配列の先頭ポインタ
in	<i>nmax</i>	配列サイズ (成分数)
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (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 3029 行で定義されています。

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

参照元 cpm_wait_BndCommS4DEx_(), と cpm_wait_BndCommV3DEx().

6.7.3.59 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	<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 2815 行で定義されています。

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

参照元 cpm_wait_BndCommV3D().

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

cpm_wait_BndCommV3DEx

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2983 行で定義されています。

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

参照元 cpm_wait_BndCommV3DEx().

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

`cpm_Waitall`

- `MPI_Waitall` のインターフェイス

引数

in	<i>count</i>	リクエストの数
in	<i>reqNoList</i>	リクエスト番号のリスト

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_frtIF.cpp` の 2537 行で定義されています。

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

参照元 `cpm_Waitall_()`.

6.7.3.62 `int cpm_ParaManager::CreateProcessGroup (int nproc, int * proclist, 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>proclist</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 597 行で定義されています。

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

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

並列プロセス数からI,J,K 方向の分割数を取得する

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager.cpp` の 445 行で定義されています。

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

参照元 `VoxelInit()`.

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

VOXEL 空間マップを検索

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

VOXEL 空間情報ポインタ

`cpm_ParaManager.cpp` の 639 行で定義されています。

参照先 `m_voxelInfoMap`.

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

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

flush

参照元 `Initialize()`.

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

flush

6.7.3.67 `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

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

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

Gather

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager_MPI.cpp` の 418 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()`.

6.7.3.69 `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_()`.

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

Gatherv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 476 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().

6.7.3.71 `cpm_ParaManager * cpm_ParaManager::get_instance () [static]`

唯一のインスタンスの取得

戻り値

インスタンスのポインタ

cpm_ParaManager.cpp の 19 行で定義されています。

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

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

唯一のインスタンスの取得 (initialize 処理も実行)

引数

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

戻り値

インスタンスのポインタ

cpm_ParaManager.cpp の 31 行で定義されています。

参照先 CPM_SUCCESS, と Initialize().

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

袖通信バッファの取得

- ・ 袖通信バッファ情報の取得

引数

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

戻り値

袖通信バッファ情報のポインタ

cpm_ParaManager.h の 1842 行で定義されています。

参照先 m_bndCommInfoMap.

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

6.7.3.74 `size_t cpm_ParaManager::GetBndCommBufferSize (int procGrpNo = 0)`

袖通信バッファサイズの取得

- ・ 袖通信バッファとして確保されている配列サイズ (byte) を返す

引数

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

戻り値

バッファサイズ (byte)

cpm_ParaManager.cpp の 990 行で定義されています。

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

6.7.3.75 `bool cpm_ParaManager::GetBndIndexExtGc (int id, int * array, int vc, int & ista, int & jsta, int & ksta, int & ilen, int & jlen, int & klen, int procGrpNo = 0)`

指定 id を含む全体ボクセル空間のインデクス範囲を取得

- 全体空間実セルのスタートインデックスを 0 としたときの, i,j,k 各方向の スタートインデックスと長さを取得する。

引数

in	<i>id</i>	判定する id
in	<i>array</i>	判定対象の配列ポインタ
in	<i>vc</i>	仮想セル数
out	<i>ista</i>	I 方向範囲のスタートインデックス
out	<i>jsta</i>	J 方向範囲のスタートインデックス
out	<i>ksta</i>	K 方向範囲のスタートインデックス
out	<i>ilen</i>	I 方向範囲の長さ
out	<i>jlen</i>	J 方向範囲の長さ
out	<i>klen</i>	K 方向範囲の長さ
in	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)

戻り値

<i>true</i>	指定 id を含むセルが存在した
<i>false</i>	指定 id を含むセルが存在しない

cpm_ParaManager.cpp の 805 行で定義されています。

参照先 GetLocalVoxelSize().

6.7.3.76 `bool cpm_ParaManager::GetBndIndexExtGc (int id, int * array, int imax, int jmax, int kmax, int vc, int & ista, int & jsta, int & ksta, int & ilen, int & jlen, int & klen, int procGrpNo = 0)`

指定 id を含む全体ボクセル空間のインデックス範囲を取得

- 全体空間実セルのスタートインデックスを 0 としたときの, i,j,k 各方向の スタートインデックスと長さを取得する。

引数

in	<i>id</i>	判定する id
in	<i>array</i>	判定対象の配列ポインタ
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
out	<i>ista</i>	I 方向範囲のスタートインデックス
out	<i>jsta</i>	J 方向範囲のスタートインデックス
out	<i>ksta</i>	K 方向範囲のスタートインデックス
out	<i>ilen</i>	I 方向範囲の長さ
out	<i>jlen</i>	J 方向範囲の長さ
out	<i>klen</i>	K 方向範囲の長さ
in	<i>procGrpNo</i>	プロセスグループ番号 (省略時=0)

戻り値

<i>true</i>	指定 id を含むセルが存在した
<i>false</i>	指定 id を含むセルが存在しない

cpm_ParaManager.cpp の 825 行で定義されています。

参照先 _IDX_S3D, Allreduce(), CPM_SUCCESS, GetGlobalVoxelSize(), と GetVoxelHeadIndex().

6.7.3.77 `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 649 行で定義されています。

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

参照元 cpm_GetDivNum_().

6.7.3.78 `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 745 行で定義されています。

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

参照元 cpm_GetDivPos_().

6.7.3.79 `const double * 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 685 行で定義されています。

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

参照元 cpm_GetGlobalOrigin_().

6.7.3.80 `const double * cpm_ParaManager::GetGlobalRegion (int procGrpNo = 0)`

全体空間サイズを取得

引数

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

戻り値

全体空間サイズ実数配列のポインタ

cpm_ParaManager.cpp の 697 行で定義されています。

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

参照元 cpm_GetGlobalRegion_().

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

全体ボクセル数を取得

引数

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

戻り値

全体ボクセル数の整数配列ポインタ (3word)

cpm_ParaManager.cpp の 673 行で定義されています。

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

参照元 cpm_GetGlobalVoxelSize_(), と GetBndIndexExtGc().

6.7.3.82 `std::string cpm_ParaManager::GetHostName ()`

ホスト名の取得

- ・ 自ランクのホスト名を取得

戻り値

ホスト名

cpm_ParaManager_MPI.cpp の 135 行で定義されています。

6.7.3.83 `const double * cpm_ParaManager::GetLocalOrigin (int procGrpNo = 0)`

自ランクの空間原点を取得

引数

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

戻り値

自ランクの空間原点実数配列のポインタ

cpm_ParaManager.cpp の 721 行で定義されています。

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

参照元 cpm_GetLocalOrigin_(), と cpm_GetLocalRegion_().

6.7.3.84 `const double * cpm_ParaManager::GetLocalRegion (int procGrpNo = 0)`

自ランクの空間サイズを取得

引数

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

戻り値

自ランクの空間サイズ実数配列のポインタ

cpm_ParaManager.cpp の 733 行で定義されています。

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

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

自ランクのボクセル数を取得

引数

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

戻り値

ローカルボクセル数の整数配列ポインタ (3word)

cpm_ParaManager.cpp の 709 行で定義されています。

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

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

6.7.3.86 `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

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

戻り値

MPI コミュニケーター

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

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

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

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

MPI_Datatype を取得

引数

in	<i>ptr</i>	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()`.

6.7.3.88 `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

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

戻り値

MPI_Datatype

`cpm_ParaManager_MPI.cpp` の 24 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()`.

6.7.3.89 `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

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

戻り値

MPI_Op

cpm_ParaManager_MPI.cpp の 58 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_().

6.7.3.90 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

<i>in</i>	<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 79 行で定義されています。

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

参照元 cpm_GetMyRankID_().

6.7.3.91 const int * cpm_ParaManager::GetNeighborRankID (int *procGrpNo* = 0)

自ランクの隣接ランク番号を取得

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

自ランクの隣接ランク番号整数配列のポインタ

cpm_ParaManager.cpp の 781 行で定義されています。

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

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

6.7.3.92 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_MPl.cpp の 107 行で定義されています。

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

参照元 cpm_GetNumRank_(), Voxellnit(), と Voxellnit_Subdomain().

6.7.3.93 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

引数

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

戻り値

自ランクの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 793 行で定義されています。

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

参照元 cpm_GetPeriodicRankID_(), PeriodicCommsS4D(), と PeriodicCommsS4DEx().

6.7.3.94 const double * 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 661 行で定義されています。

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

参照元 cpm_GetPitch_().

6.7.3.95 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

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

自ランクの始点インデクス整数配列のポインタ

cpm_ParaManager.cpp の 757 行で定義されています。

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

参照元 cpm_GetVoxelHeadIndex_(), と GetBndIndexExtGc().

6.7.3.96 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

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

自ランクの終点インデクス整数配列のポインタ

cpm_ParaManager.cpp の 769 行で定義されています。

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

参照元 cpm_GetVoxelTailIndex_().

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

配列の初期化処理

引数

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

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

6.7.3.98 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 Barrier(), CPM_ERROR_MPI, CPM_ERROR_NO_MPI_INIT, CPM_SUCCESS, flush(), IsParallel(), m_n-Rank, と m_rankNo.

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

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 167 行で定義されています。

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

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

lrecv

- MPI_lrecv のインターフェイス

引数

out	buf	受信データ
in	count	受信データのサイズ
in	source	送信元のランク番号 (procGrpNo 内でのランク番号)
out	request	リクエストハンドル
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 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_lrecv(), と sendrecv().

6.7.3.101 cpm_ErrorCode cpm_ParaManager::lrecv (MPI_Datatype dtype, void * buf, int count, int source, MPI_Request * request, int procGrpNo = 0)

lrecv

- MPI_lrecv 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 362 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().

6.7.3.102 `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(), と sendrecv().

6.7.3.103 `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 333 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()`.

6.7.3.104 `bool cpm_ParaManager::IsInnerBoundary (cpm_FaceFlag face, int procGrpNo = 0)`

自ランクの境界が内部境界 (隣が不活性ドメイン) かどうかを判定

引数

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

戻り値

<code>true</code>	内部境界
<code>false</code>	内部境界でない

`cpm_ParaManager.cpp` の 910 行で定義されています。

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

6.7.3.105 `bool cpm_ParaManager::IsOuterBoundary (cpm_FaceFlag face, int procGrpNo = 0)`

自ランクの境界が外部境界かどうかを判定

引数

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

戻り値

<code>true</code>	外部境界
<code>false</code>	外部境界でない

`cpm_ParaManager.cpp` の 897 行で定義されています。

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

6.7.3.106 `bool cpm_ParaManager::IsParallel ()`

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

戻り値

<code>true</code>	並列実行
<code>false</code>	逐次実行

`cpm_ParaManager.cpp` の 191 行で定義されています。

参照先 `m_nRank`.

参照元 `cpm_IsParallel()`, と `Initialize()`.

6.7.3.107 `bool cpm_ParaManager::IsParallel () const`

並列実行であるかチェックする (const)

- 並列実行であっても、並列数が 1 のときは false となる

戻り値

<i>true</i>	並列実行
<i>false</i>	逐次実行

`cpm_ParaManager.cpp` の 203 行で定義されています。

参照先 `m_nRank`.

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

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

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

引数

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

in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
out	sendm	マイナス方向の送信バッファ
out	sendp	プラス方向の送信バッファ
in	nIDm	マイナス方向の隣接ランク番号
in	nIDp	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

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

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

引数

in	array	袖通信をする配列の先頭ポインタ
in	nmax	配列サイズ (成分数)
in	imax	配列サイズ (I 方向)

in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
out	<i>sendm</i>	マイナス方向の送信バッファ
out	<i>sendp</i>	プラス方向の送信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 569 行で定義されています。

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

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

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 629 行で定義されています。

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

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

6.7.3.114 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicCommS3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar3D 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 `PeriodicCommS4D()`.

参照元 `cpm_PeriodicCommS3D_()`.

6.7.3.115 `cpm_ErrorCode cpm_ParaManager::PeriodicCommS3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo = 0)`

周期境界袖通信 (Scalar3D 版, MPI_Datatype 指定)

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

- MPI_Datatype を指定するバージョン

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 721 行で定義されています。

参照先 PeriodicCommS4D().

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

周期境界袖通信 (Scalar4D 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

6.7.3.117 **cpm_ErrorCode cpm_ParaManager::PeriodicComms4D** (MPI_Datatype *dtype*, void * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, cpm_DirFlag *dir*, cpm_PMFlag *pm*, int *procGrpNo* = 0)

周期境界袖通信 (Scalar4D 版, MPI_Datatype 指定)

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

引数

in	<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 739 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicComms4D().

6.7.3.118 **template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::PeriodicComms4DEx** (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, cpm_DirFlag *dir*, cpm_PMFlag *pm*, int *procGrpNo* = 0)

周期境界袖通信 (Scalar4DEx 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BOTH, CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_GET_PERIODIC_RANK, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), Get-

PeriodicRankID(), cpm_Base::getRankNull(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, MINUS2PLUS, packXEx(), packYEx(), packZEx(), PLUS2MINUS, sendrecv(), unpackXEx(), unpackYEx(), unpackZEx(), Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_PeriodicCommS4DEx(), PeriodicCommS4DEx(), と PeriodicCommV3DEx().

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

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 951 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4DEx().

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

周期境界袖通信 (Vector3D 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 PeriodicComms4D().

参照元 cpm_PeriodicCommV3D_().

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

周期境界袖通信 (Vector3D 版, MPI_Datatype 指定)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 730 行で定義されています。

参照先 PeriodicComms4D().

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

周期境界袖通信 (Vector3DEx 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 PeriodicComms4DEx().

参照元 cpm_PeriodicCommV3DEx_().

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

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 942 行で定義されています。

参照先 PeriodicComms4DEx().

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

Recv

- MPI_Recv のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Recv().

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

Recv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 304 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().

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

Send

- MPI_Send のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Send().

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

Send

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 276 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().

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

1 方向 (プラス、マイナス) の双方向袖通信処理

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_SUCCESS, lrecv(), lsend(), と cpm_Base::IsRankNull().

参照元 BndCommsS4D(), BndCommsS4D_nowait(), BndCommsS4DEx(), BndCommsS4DEx_nowait(), PeriodicCommS4D(), と PeriodicCommS4DEx().

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

袖通信バッファのセット

- 6face 分の送受信バッファを確保する

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 923 行で定義されています。

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

参照元 cpm_SetBndCommBuffer(), と VoxelInit().

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

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

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

引数

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

in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 539 行で定義されています。

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

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

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

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

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

引数

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

in	<i>recvm</i>	マイナス方向の受信バッファ
in	<i>recvp</i>	プラス方向の受信バッファ
in	<i>nIDm</i>	マイナス方向の隣接ランク番号
in	<i>nIDp</i>	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 599 行で定義されています。

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

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

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

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

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

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

引数

	<i>inout]</i>	array 袖通信をした配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>nmax</i>	配列サイズ (成分数)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 659 行で定義されています。

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

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

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

領域分割

- 既に作成済みの領域分割情報を用いた領域分割処理

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 215 行で定義されています。

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

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

6.7.3.137 `cpm_ErrorCode cpm_ParaManager::Voxellnit (int div[3], int vox[3], double origin[3], double region[3], size_t maxVC = 1, size_t maxN = 3, int procGrpNo = 0)`

領域分割

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

引数

<code>in</code>	<code>div</code>	領域分割数
<code>in</code>	<code>vox</code>	空間全体のボクセル数
<code>in</code>	<code>origin</code>	空間全体の原点
<code>in</code>	<code>region</code>	空間全体のサイズ
<code>in</code>	<code>maxVC</code>	最大の袖数 (袖通信用)
<code>in</code>	<code>maxN</code>	最大の成分数 (袖通信用)
<code>in</code>	<code>procGrpNo</code>	領域分割を行うプロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager.cpp` の 291 行で定義されています。

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

6.7.3.138 `cpm_ErrorCode cpm_ParaManager::Voxellnit (int vox[3], double origin[3], double region[3], size_t maxVC = 1, size_t maxN = 3, int procGrpNo = 0)`

領域分割

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

引数

<code>in</code>	<code>vox</code>	空間全体のボクセル数
<code>in</code>	<code>origin</code>	空間全体の原点
<code>in</code>	<code>region</code>	空間全体のサイズ
<code>in</code>	<code>maxVC</code>	最大の袖数 (袖通信用)
<code>in</code>	<code>maxN</code>	最大の成分数 (袖通信用)
<code>in</code>	<code>procGrpNo</code>	領域分割を行うプロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

`cpm_ParaManager.cpp` の 341 行で定義されています。

参照先 `Voxellnit()`.

6.7.3.139 `cpm_ErrorCode cpm_ParaManager::Voxellnit_Subdomain (int div[3], int vox[3], double origin[3], double region[3], std::string subDomainFile, size_t maxVC = 1, size_t maxN = 3, int procGrpNo = 0)`

領域分割 (ActiveSubdomain 指定)

- ・ 領域分割の各種情報を引数で渡して領域分割を行う
- ・ ActiveSubdomain ファイルで指定される領域分割位置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
- ・ 指定の領域分割数とActiveSubdomain ファイルで指定されている領域分割数が一致している必要がある
- ・ ActiveSubdomain 数と並列数が一致している必要がある

引数

in	<i>div</i>	領域分割数
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>region</i>	空間全体のサイズ
in	<i>subDomainFile</i>	ActiveSubdomain ファイル名
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 352 行で定義されています。

参照先 cpm_GlobalDomainInfo::AddSubdomain(), CPM_ERROR_INVALID_REGION, CPM_ERROR_INVALID_VOXELSIZE, CPM_ERROR_MISMATCH_DIV_SUBDOMAIN, CPM_ERROR_MISMATCH_NP_SUBDOMAIN, CPM_SUCCESS, GetNumRank(), cpm_GlobalDomainInfo::ReadActiveSubdomainFile(), cpm_GlobalDomainInfo::SetDivNum(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), と Voxellnit().

参照元 Voxellnit_Subdomain().

6.7.3.140 `cpm_ErrorCode cpm_ParaManager::Voxellnit_Subdomain (int vox[3], double origin[3], double region[3], std::string subDomainFile, size_t maxVC = 1, size_t maxN = 3, int procGrpNo = 0)`

領域分割 (ActiveSubdomain 指定)

- ・ 領域分割の各種情報を引数で渡して領域分割を行う
- ・ ActiveSubdomain ファイルで指定される領域分割位置のランクが活性ドメインになる
- ・ ActiveSubdomain ファイルで指定されている領域分割数で領域分割を行う
- ・ ActiveSubdomain 数と並列数が一致している必要がある

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>region</i>	空間全体のサイズ
in	<i>subDomainFile</i>	ActiveSubdomain ファイル名
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 432 行で定義されています。

参照先 Voxellnit_Subdomain().

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

Wait

- ・ MPI_Wait のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 192 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.

6.7.3.142 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndComms3D (T * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar3D 版)

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

引数

	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	req	MPI リクエスト
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 wait_BndComms4D().

参照元 cpm_wait_BndComms3D().

6.7.3.143 `cpm_ErrorCode cpm_ParaManager::wait_BndComms3D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

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

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 659 行で定義されています。

参照先 wait_BndComms4D().

6.7.3.144 `template<class T> CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndComms4D (T * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

非同期版袖通信の wait、展開 (Scalar4D 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

参照元 cpm_wait_BndComms4D(), wait_BndComms3D(), wait_BndComms4D(), と wait_BndCommV3D().

6.7.3.145 `cpm_ErrorCode cpm_ParaManager::wait_BndComms4D (MPI_Datatype dtype, void * array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo = 0)`

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

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

引数

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

in	req	MPI リクエスト
in	procGrpNo	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 677 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と wait_BndCommS4D().

6.7.3.146 `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

	inout]	array 袖通信をする配列の先頭ポインタ
in	nmax	配列サイズ (成分数)
in	imax	配列サイズ (I 方向)
in	jmax	配列サイズ (J 方向)
in	kmax	配列サイズ (K 方向)
in	vc	仮想セル数
in	vc_comm	通信する仮想セル数
in	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 270 行で定義されています。

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

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

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

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

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

引数

in	dtype	袖通信データのMPI_Datatype
	inout]	array 袖通信をする配列の先頭ポインタ
in	nmax	配列サイズ (成分数)

in	<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 898 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と wait_BndCommS4DEx().

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

非同期版袖通信の wait、展開 (Vector3D 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 wait_BndCommS4D().

参照元 cpm_wait_BndCommV3D().

6.7.3.149 `cpm_ErrorCode cpm_ParaManager::wait_BndCommV3D (MPI.Datatype dtype, void * array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI.Request req[12], int procGrpNo = 0)`

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

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

引数

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

in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>req</i>	MPI リクエスト
in	<i>procGrpNo</i>	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 668 行で定義されています。

参照先 wait_BndComms4D().

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

非同期版袖通信の wait、展開 (Vector3DEx 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 260 行で定義されています。

参照先 wait_BndComms4DEx().

参照元 cpm_wait_BndCommV3DEx().

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

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

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

引数

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

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 889 行で定義されています。

参照先 wait_BndComms4DEx().

6.7.3.152 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 216 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().

6.7.4 変数

6.7.4.1 BndCommInfoMap cpm_ParaManager::m_bndCommInfoMap [private]

プロセスグループ毎の袖通信バッファ情報

cpm_ParaManager.h の 2120 行で定義されています。

参照元 cpm_ParaManager(), GetBndCommBuffer(), GetBndCommBufferSize(), SetBndCommBuffer(), と ~cpm_ParaManager().

6.7.4.2 int cpm_ParaManager::m_nRank [private]

プロセス並列数

cpm_ParaManager.h の 2093 行で定義されています。

参照元 cpm_ParaManager(), Initialize(), と IsParallel().

6.7.4.3 std::vector<MPI_Comm> cpm_ParaManager::m_procGrpList [private]

プロセスグループのリスト

- VOXEL 空間番号をインデクスとしたVOXEL 空間のMPI コミュニケータを格納

- `vector` のインデクス=プロセスグループ番号とする
- [0] には必ず `MPI_COMM_WORLD` を格納
- 自ランクが含まれるプロセスグループのみを管理する (同じプロセスグループでもプロセス毎に異なるプロセスグループ番号になる場合もある)

`cpm_ParaManager.h` の 2105 行で定義されています。

参照元 `cpm_ParaManager()`, `CreateProcessGroup()`, `GetMPI_Comm()`, `GetMyRankID()`, `GetNumRank()`, `Voxel-Init()`, と `~cpm_ParaManager()`.

6.7.4.4 `int cpm_ParaManager::m_rankNo` [private]

`MPI_COMM_WORLD` での自ランク番号

`cpm_ParaManager.h` の 2096 行で定義されています。

参照元 `cpm_ParaManager()`, と `Initialize()`.

6.7.4.5 `RankNoMap cpm_ParaManager::m_rankNoMap` [private]

プロセスグループ毎のランク番号マップ

- VOXEL 空間番号をキーとしたランク番号マップ (未使用)

`cpm_ParaManager.h` の 2116 行で定義されています。

参照元 `cpm_ParaManager()`, と `~cpm_ParaManager()`.

6.7.4.6 `cpm_ObjList<MPI.Request> cpm_ParaManager::m_reqList` [private]

`MPI_Request` の管理マップ

- Fortran インターフェイス用

`cpm_ParaManager.h` の 2125 行で定義されています。

参照元 `cpm_BndCommsS3D_nowait()`, `cpm_BndCommsS4D_nowait()`, `cpm_BndCommsS4DEx_nowait()`, `cpm_BndCommV3D_nowait()`, `cpm_BndCommV3DEx_nowait()`, `cpm_lrecv()`, `cpm_lsend()`, `cpm_Wait()`, `cpm_wait_BndCommS3D()`, `cpm_wait_BndCommS4D()`, `cpm_wait_BndCommS4DEx()`, `cpm_wait_BndCommV3D()`, `cpm_wait_BndCommV3DEx()`, と `cpm_Waitall()`.

6.7.4.7 `VoxelInfoMap cpm_ParaManager::m_voxelInfoMap` [private]

プロセスグループ毎のVOXEL 空間情報マップ

- VOXEL 空間番号をキーとしたVOXEL 空間情報マップ
- 自ランクが含まれるVOXEL 空間のみを管理する

`cpm_ParaManager.h` の 2111 行で定義されています。

参照元 `cpm_ParaManager()`, `FindVoxelInfo()`, `VoxelInit()`, と `~cpm_ParaManager()`.

このクラスの説明は次のファイルから生成されました:

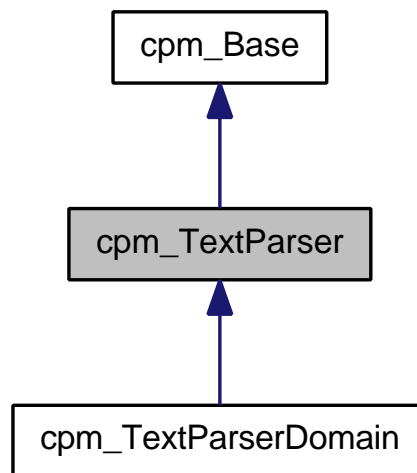
- [cpm_ParaManager.h](#)
- [cpm_ParaManager.cpp](#)

- [cpm_ParaManager_Alloc.cpp](#)
- [cpm_ParaManager_frtIF.cpp](#)
- [cpm_ParaManager_MPI.cpp](#)
- [cpm_ParaManager_BndComm.h](#)
- [cpm_ParaManager_BndCommEx.h](#)
- [cpm_ParaManager_inline.h](#)

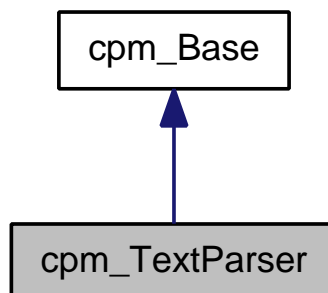
6.8 クラス cpm_TextParser

```
#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`

6.8.1 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。

6.8.2 コンストラクタとデストラクタ**6.8.2.1 `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`.

6.8.2.2 `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`.

6.8.3 関数**6.8.3.1 `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

<code>in</code>	<code>filename</code>	読み込むファイル名
-----------------	-----------------------	-----------

戻り値

`TextParser` クラスの終了コード

`cpm_TextParser.cpp` の 34 行で定義されています。

参照先 `m_tp`.

参照元 `cpm_TextParserDomain::ReadMain()`.

6.8.3.2 `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

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

戻り値

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

`cpm_TextParser.cpp` の 53 行で定義されています。

参照先 `m_tp`.

参照元 `cpm_TextParserDomain::ReadDomainInfo()`.

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

ベクトルデータの読み込み (倍精度実数版)

引数

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

戻り値

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

`cpm_TextParser.cpp` の 88 行で定義されています。

参照先 `CPM_ERROR_TP_NOVECTOR`, `CPM_ERROR_TP_VECTOR_SIZE`, と `m_tp`.

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

ベクトルデータの読み込み (整数版)

引数

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

戻り値

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

`cpm_TextParser.cpp` の 123 行で定義されています。

参照先 `m_tp`.

6.8.4 変数

6.8.4.1 TextParser* cpm_TextParser::m_tp [protected]

テキストパーサークラスのインスタンス

cpm_TextParser.h の 91 行で定義されています。

参照元 cpm_TextParser(), Read(), cpm_TextParserDomain::ReadDomainInfo(), cpm_TextParserDomain::ReadSubdomainInfo(), readVector(), と ~cpm_TextParser().

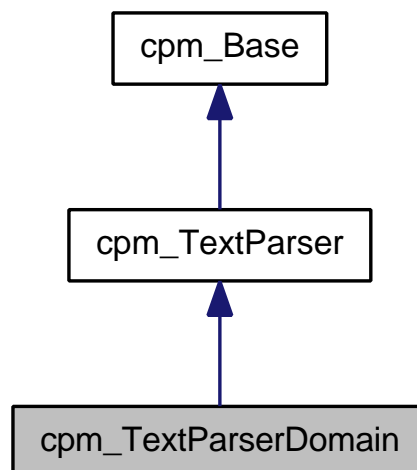
このクラスの説明は次のファイルから生成されました:

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

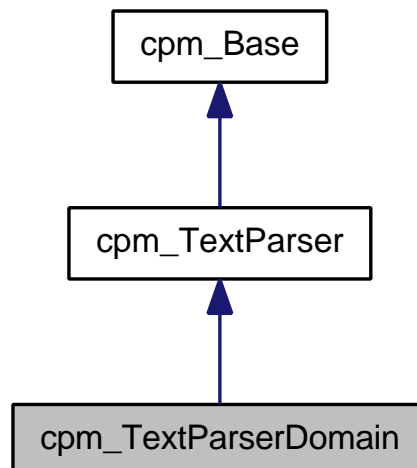
6.9 クラス cpm_TextParserDomain

```
#include <cpm_TextParserDomain.h>
```

cpm_TextParserDomain に対する継承グラフ



cpm_TextParserDomain のコラボレーション図



Public メソッド

- [cpm_TextParserDomain](#) ()
- virtual [~cpm_TextParserDomain](#) ()

Static Public メソッド

- static [cpm_GlobalDomainInfo](#) * [Read](#) (std::string filename, int &errorcode)

Private メソッド

- [cpm_GlobalDomainInfo](#) * [ReadMain](#) (std::string filename, int &errorcode)
- int [ReadDomainInfo](#) ([cpm_GlobalDomainInfo](#) *dInfo)
- int [ReadSubdomainInfo](#) ([cpm_GlobalDomainInfo](#) *dInfo, std::string tpfname)

6.9.1 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。

6.9.2 コンストラクタとデストラクタ

6.9.2.1 [cpm_TextParserDomain::cpm_TextParserDomain](#) ()

コンストラクタ

[cpm_TextParserDomain.cpp](#) の 19 行で定義されています。

6.9.2.2 [cpm_TextParserDomain::~~cpm_TextParserDomain](#) () [virtual]

デストラクタ

[cpm_TextParserDomain.cpp](#) の 26 行で定義されています。

6.9.3 関数

6.9.3.1 [cpm_GlobalDomainInfo](#) * [cpm_TextParserDomain::Read](#) (std::string *filename*, int & *errorcode*) [static]

読み込み処理

- TextParser クラスを用いて領域分割情報ファイルを読み込む
- TextParser クラスのインスタンスはクリア (remove) される

引数

in	<i>filename</i>	読み込むファイル名
out	<i>errorcode</i>	CPM エラーコード

戻り値

領域情報ポインタ

[cpm_TextParserDomain.cpp](#) の 33 行で定義されています。

参照先 `ReadMain()`.

6.9.3.2 `int cpm_TextParserDomain::ReadDomainInfo (cpm_GlobalDomainInfo * dInfo) [private]`

DomainInfo の読み込み

引数

<i>inout</i>	dInfo 領域情報
--------------	------------

戻り値

CPM エラーコード

`cpm_TextParserDomain.cpp` の 83 行で定義されています。

参照先 `CPM_ERROR_INVALID_PTR`, `CPM_ERROR_TP_INVALID_G_DIV`, `CPM_ERROR_TP_INVALID_G_ORG`, `CPM_ERROR_TP_INVALID_G_PITCH`, `CPM_ERROR_TP_INVALID_G_RGN`, `CPM_ERROR_TP_INVALID_G_VOXEL`, `cpm_Base::cpm_strCompare()`, `CPM_SUCCESS`, `cpm_TextParser::m_tp`, `cpm_TextParser::readVector()`, `cpm_GlobalDomainInfo::SetDivNum()`, `cpm_DomainInfo::SetOrigin()`, `cpm_DomainInfo::SetPitch()`, `cpm_DomainInfo::SetRegion()`, と `cpm_DomainInfo::SetVoxNum()`.

参照元 `ReadMain()`.

6.9.3.3 `cpm_GlobalDomainInfo * cpm_TextParserDomain::ReadMain (std::string filename, int & errorcode) [private]`

読み込み処理のメイン

- TextParser クラスを用いて領域分割情報ファイルを読み込む
- TextParser クラスのインスタンスはクリア (remove) される

引数

in	<i>filename</i>	読み込むファイル名
out	<i>errorcode</i>	CPM エラーコード

戻り値

領域情報ポインタ

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

参照先 `CPM_ERROR_INVALID_PTR`, `cpm_TextParser::Read()`, `ReadDomainInfo()`, と `ReadSubdomainInfo()`.

参照元 `Read()`.

6.9.3.4 `int cpm_TextParserDomain::ReadSubdomainInfo (cpm_GlobalDomainInfo * dInfo, std::string tpfname) [private]`

ActiveSubdomainInfo の読み込み

引数

	<i>inout</i>	dInfo 領域情報
in	<i>tpfname</i>	メインの領域分割情報ファイル名

戻り値

CPM エラーコード

cpm_TextParserDomain.cpp の 245 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_TP_INVALID_G_DIV, cpm_Base::cpm_strCompare(), CPM_SUCCESS, CPM_PATH::cpmPath_concat(), CPM_PATH::cpmPath_isAbsolute(), CES::DirName(), cpm_GlobalDomainInfo::GetDivNum(), cpm_TextParser::m_tp, と cpm_GlobalDomainInfo::ReadActiveSubdomainFile().

参照元 ReadMain().

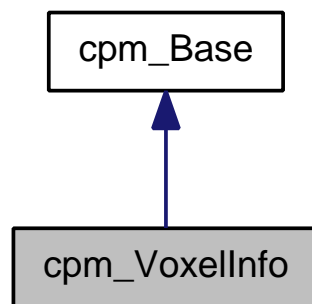
このクラスの説明は次のファイルから生成されました:

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

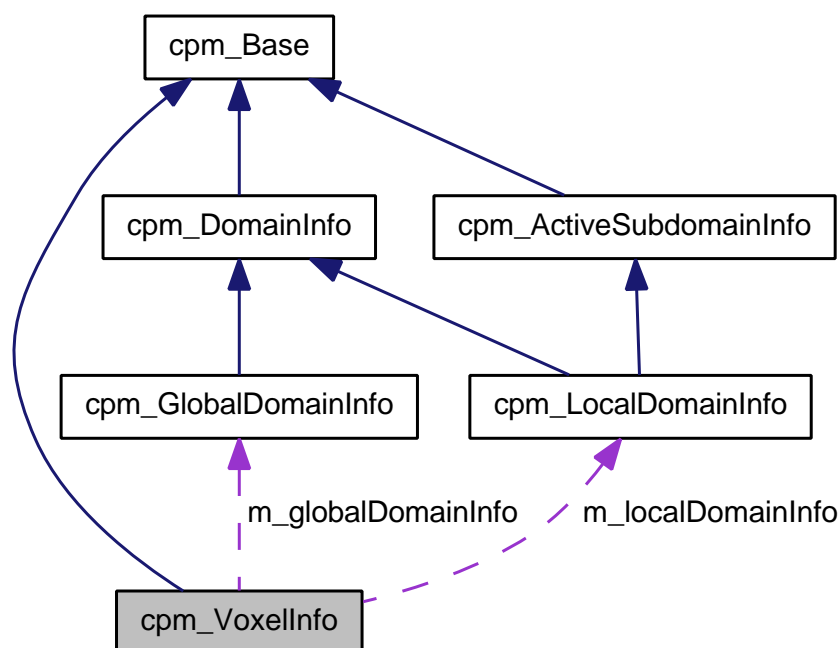
6.10 クラス cpm_VoxelInfo

```
#include <cpm_VoxelInfo.h>
```

cpm_VoxelInfo に対する継承グラフ



cpm_VoxelInfo のコラボレーション図



Private メソッド

- `cpm_VoxelInfo` ()
- `virtual ~cpm_VoxelInfo` ()
- `cpm_ErrorCode Init` (MPI_Comm comm, `cpm_GlobalDomainInfo` *dInfo)
- `bool CreateRankMap` ()
- `bool CreateNeighborRankInfo` ()
- `bool CreateLocalDomainInfo` ()
- `const int * GetDivNum` () const
- `const double * GetPitch` () const
- `const int * GetGlobalVoxelSize` () const
- `const double * GetGlobalOrigin` () const
- `const double * GetGlobalRegion` () const
- `const int * GetLocalVoxelSize` () const
- `const double * GetLocalOrigin` () const
- `const double * GetLocalRegion` () const
- `const int * GetDivPos` () const
- `const int * GetVoxelHeadIndex` () const
- `const int * GetVoxelTailIndex` () const
- `const int * GetNeighborRankID` () const
- `const int * GetPeriodicRankID` () const
- `bool IsOuterBoundary` (`cpm_FaceFlag` face) const
- `bool IsInnerBoundary` (`cpm_FaceFlag` face) const

Private 変数

- `cpm_GlobalDomainInfo m_globalDomainInfo`
空間全体の領域情報
- `cpm_LocalDomainInfo m_localDomainInfo`
自ランクの領域情報
- `int m_voxelHeadIndex` [3]
自ランクの始点ボクセルインデックス
- `int m_voxelTailIndex` [3]
自ランクの終点ボクセルインデックス
- `MPI_Comm m_comm`
MPI コミュニケータ
- `int m_nRank`
コミュニケータ内のランク数 (=プロセス並列数)
- `int m_rankNo`
コミュニケータ内でのランク番号
- `int m_neighborRankID` [6]
隣接ランク番号 (外部境界は負の値)
- `int m_periodicRankID` [6]
周期境界の隣接ランク番号
- `int * m_rankMap`
ランクマップ

フレンド

- `class cpm_ParaManager`

6.10.1 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。

6.10.2 コンストラクタとデストラクタ

6.10.2.1 cpm_VoxelInfo::cpm_VoxelInfo () [private]

コンストラクタ

cpm_VoxelInfo.cpp の 18 行で定義されています。

参照先 cpm_Base::getRankNull(), m_comm, m_neighborRankID, m_nRank, m_periodicRankID, m_rankMap, m_rankNo, m_voxelHeadIndex, と m_voxelTailIndex.

6.10.2.2 cpm_VoxelInfo::~cpm_VoxelInfo () [private, virtual]

デストラクタ

cpm_VoxelInfo.cpp の 43 行で定義されています。

参照先 m_rankMap.

6.10.3 関数

6.10.3.1 bool cpm_VoxelInfo::CreateLocalDomainInfo () [private]

ローカル領域情報を生成

戻り値

<i>true</i>	正常終了
<i>false</i>	エラー

cpm_VoxelInfo.cpp の 158 行で定義されています。

参照先 _IDX_S3D, cpm_GlobalDomainInfo::GetDivNum(), cpm_DomainInfo::GetOrigin(), cpm_DomainInfo::GetPitch(), cpm_DomainInfo::GetVoxNum(), m_globalDomainInfo, m_localDomainInfo, m_rankMap, m_rankNo, m_voxelHeadIndex, m_voxelTailIndex, cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_ActiveSubdomainInfo::SetPos(), cpm_DomainInfo::SetRegion(), と cpm_DomainInfo::SetVoxNum().

参照元 Init().

6.10.3.2 bool cpm_VoxelInfo::CreateNeighborRankInfo () [private]

隣接ランク情報を生成

戻り値

<i>true</i>	正常終了
<i>false</i>	エラー

cpm_VoxelInfo.cpp の 266 行で定義されています。

参照先 _IDX_S3D, cpm_GlobalDomainInfo::GetDivNum(), cpm_ActiveSubdomainInfo::GetPos(), cpm_Base::getRankNull(), m_globalDomainInfo, m_localDomainInfo, m_neighborRankID, m_periodicRankID, m_rankMap, m_rankNo, X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 `Init()`.

6.10.3.3 `bool cpm_VoxellInfo::CreateRankMap () [private]`

ランクマップを生成

戻り値

<code>true</code>	正常終了
<code>false</code>	エラー

`cpm_VoxellInfo.cpp` の 95 行で定義されています。

参照先 `_IDX_S3D`, `cpm_GlobalDomainInfo::GetDivNum()`, `cpm_ActiveSubdomainInfo::GetPos()`, `cpm_Base::getRankNull()`, `cpm_GlobalDomainInfo::GetSubdomainInfo()`, `cpm_GlobalDomainInfo::GetSubdomainNum()`, `m_globalDomainInfo`, と `m_rankMap`.

参照元 `Init()`.

6.10.3.4 `const int * cpm_VoxellInfo::GetDivNum () const [private]`

領域分割数を取得

戻り値

領域分割数整数配列のポインタ

`cpm_VoxellInfo.cpp` の 372 行で定義されています。

参照先 `cpm_GlobalDomainInfo::GetDivNum()`, と `m_globalDomainInfo`.

参照元 `cpm_ParaManager::GetDivNum()`, `IsInnerBoundary()`, と `IsOuterBoundary()`.

6.10.3.5 `const int * cpm_VoxellInfo::GetDivPos () const [private]`

自ランクの領域分割位置を取得

戻り値

自ランクの領域分割位置整数配列のポインタ

`cpm_VoxellInfo.cpp` の 436 行で定義されています。

参照先 `cpm_ActiveSubdomainInfo::GetPos()`, と `m_localDomainInfo`.

参照元 `cpm_ParaManager::GetDivPos()`, `IsInnerBoundary()`, と `IsOuterBoundary()`.

6.10.3.6 `const double * cpm_VoxellInfo::GetGlobalOrigin () const [private]`

全体空間の原点を取得

戻り値

全体空間の原点実数配列のポインタ

`cpm_VoxellInfo.cpp` の 396 行で定義されています。

参照先 `cpm_DomainInfo::GetOrigin()`, と `m_globalDomainInfo`.

参照元 `cpm_ParaManager::GetGlobalOrigin()`.

6.10.3.7 `const double * cpm_VoxelInfo::GetGlobalRegion () const` [private]

全体空間サイズを取得

戻り値

全体空間サイズ実数配列のポインタ

cpm_VoxelInfo.cpp の 404 行で定義されています。

参照先 cpm_DomainInfo::GetRegion(), と m_globalDomainInfo.

参照元 cpm_ParaManager::GetGlobalRegion().

6.10.3.8 `const int * cpm_VoxelInfo::GetGlobalVoxelSize () const` [private]

全体ボクセル数を取得

戻り値

全体ボクセル数整数配列のポインタ

cpm_VoxelInfo.cpp の 388 行で定義されています。

参照先 cpm_DomainInfo::GetVoxNum(), と m_globalDomainInfo.

参照元 cpm_ParaManager::GetGlobalVoxelSize().

6.10.3.9 `const double * cpm_VoxelInfo::GetLocalOrigin () const` [private]

自ランクの空間原点を取得

戻り値

自ランクの空間原点実数配列のポインタ

cpm_VoxelInfo.cpp の 420 行で定義されています。

参照先 cpm_DomainInfo::GetOrigin(), と m_localDomainInfo.

参照元 cpm_ParaManager::GetLocalOrigin().

6.10.3.10 `const double * cpm_VoxelInfo::GetLocalRegion () const` [private]

自ランクの空間サイズを取得

戻り値

自ランクの空間サイズ実数配列のポインタ

cpm_VoxelInfo.cpp の 428 行で定義されています。

参照先 cpm_DomainInfo::GetRegion(), と m_localDomainInfo.

参照元 cpm_ParaManager::GetLocalRegion().

6.10.3.11 `const int * cpm_VoxelInfo::GetLocalVoxelSize () const` [private]

自ランクのボクセル数を取得

戻り値

自ランクのボクセル数整数配列のポインタ

`cpm_VoxelInfo.cpp` の 412 行で定義されています。

参照先 `cpm_DomainInfo::GetVoxNum()`, と `m_localDomainInfo`.

参照元 `cpm_ParaManager::GetLocalVoxelSize()`.

6.10.3.12 `const int * cpm_VoxelInfo::GetNeighborRankID () const` `[private]`

自ランクの隣接ランク番号を取得

戻り値

自ランクの隣接ランク番号整数配列のポインタ

`cpm_VoxelInfo.cpp` の 460 行で定義されています。

参照先 `m_neighborRankID`.

参照元 `cpm_ParaManager::GetNeighborRankID()`.

6.10.3.13 `const int * cpm_VoxelInfo::GetPeriodicRankID () const` `[private]`

自ランクの周期境界の隣接ランク番号を取得

戻り値

自ランクの周期境界の隣接ランク番号整数配列のポインタ

`cpm_VoxelInfo.cpp` の 468 行で定義されています。

参照先 `m_periodicRankID`.

参照元 `cpm_ParaManager::GetPeriodicRankID()`.

6.10.3.14 `const double * cpm_VoxelInfo::GetPitch () const` `[private]`

ピッチを取得

戻り値

ピッチ実数配列のポインタ

`cpm_VoxelInfo.cpp` の 380 行で定義されています。

参照先 `cpm_DomainInfo::GetPitch()`, と `m_globalDomainInfo`.

参照元 `cpm_ParaManager::GetPitch()`.

6.10.3.15 `const int * cpm_VoxelInfo::GetVoxelHeadIndex () const` `[private]`

自ランクの始点VOXELの全体空間でのインデクスを取得

戻り値

自ランクの始点インデクス整数配列のポインタ

cpm_VoxelInfo.cpp の 444 行で定義されています。

参照先 m_voxelHeadIndex.

参照元 cpm_ParaManager::GetVoxelHeadIndex().

6.10.3.16 `const int * cpm_VoxelInfo::GetVoxelTailIndex () const` [private]

自ランクの終点VOXEL の全体空間でのインデクスを取得

戻り値

自ランクの終点インデクス整数配列のポインタ

cpm_VoxelInfo.cpp の 452 行で定義されています。

参照先 m_voxelTailIndex.

参照元 cpm_ParaManager::GetVoxelTailIndex().

6.10.3.17 `cpm_ErrorCode cpm_VoxelInfo::Init (MPI_Comm comm, cpm_GlobalDomainInfo * dInfo)`
[private]

CPM 領域分割情報の生成

- MPI_COMM_WORLD を使用した領域を生成する。

引数

in	<i>comm</i>	MPI コミュニケータ
in	<i>dInfo</i>	領域分割情報
in	<i>maxVC</i>	最大の袖数 (袖通信用)
in	<i>maxN</i>	最大の成分数 (袖通信用)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_VoxelInfo.cpp の 51 行で定義されています。

参照先 CPM_ERROR_CREATE_LOCALDOMAIN, CPM_ERROR_CREATE_NEIGHBOR, CPM_ERROR_CREATE_RANKMAP, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, CreateLocalDomainInfo(), CreateNeighborRankInfo(), CreateRankMap(), cpm_Base::IsCommNull(), m_comm, m_globalDomainInfo, m_nRank, と m_rankNo.

参照元 cpm_ParaManager::VoxelInit().

6.10.3.18 `bool cpm_VoxelInfo::IsInnerBoundary (cpm_FaceFlag face) const` [private]

自ランクの境界が内部境界 (隣が不活性ドメイン) かどうかを判定

引数

in	<i>face</i>	面方向
----	-------------	-----

戻り値

<i>true</i>	内部境界
<i>false</i>	内部境界でない

`cpm_VoxellInfo.cpp` の 503 行で定義されています。

参照先 `GetDivNum()`, `GetDivPos()`, `cpm_Base::IsRankNull()`, `m_neighborRankID`, `X_MINUS`, `X_PLUS`, `Y_MINUS`, `Y_PLUS`, `Z_MINUS`, と `Z_PLUS`.

参照元 `cpm_ParaManager::IsInnerBoundary()`.

6.10.3.19 `bool cpm_VoxellInfo::IsOuterBoundary (cpm_FaceFlag face) const` `[private]`

自ランクの境界が外部境界かどうかを判定

引数

<i>in</i>	<i>face</i>	面方向
-----------	-------------	-----

戻り値

<i>true</i>	外部境界
<i>false</i>	外部境界でない

`cpm_VoxellInfo.cpp` の 476 行で定義されています。

参照先 `GetDivNum()`, `GetDivPos()`, `cpm_Base::IsRankNull()`, `m_neighborRankID`, `X_MINUS`, `X_PLUS`, `Y_MINUS`, `Y_PLUS`, `Z_MINUS`, と `Z_PLUS`.

参照元 `cpm_ParaManager::IsOuterBoundary()`.

6.10.4 フレンドと関連する関数

6.10.4.1 `friend class cpm_ParaManager` `[friend]`

`cpm_VoxellInfo.h` の 25 行で定義されています。

6.10.5 変数

6.10.5.1 `MPI_Comm cpm_VoxellInfo::m_comm` `[private]`

MPI コミュニケータ

`cpm_VoxellInfo.h` の 164 行で定義されています。

参照元 `cpm_VoxellInfo()`, と `Init()`.

6.10.5.2 `cpm_GlobalDomainInfo cpm_VoxellInfo::m_globalDomainInfo` `[private]`

空間全体の領域情報

`cpm_VoxellInfo.h` の 156 行で定義されています。

参照元 `CreateLocalDomainInfo()`, `CreateNeighborRankInfo()`, `CreateRankMap()`, `GetDivNum()`, `GetGlobalOrigin()`, `GetGlobalRegion()`, `GetGlobalVoxelSize()`, `GetPitch()`, と `Init()`.

6.10.5.3 `cpm_LocalDomainInfo cpm_VoxelInfo::m_localDomainInfo` [private]

自ランクの領域情報

`cpm_VoxelInfo.h` の 159 行で定義されています。

参照元 `CreateLocalDomainInfo()`, `CreateNeighborRankInfo()`, `GetDivPos()`, `GetLocalOrigin()`, `GetLocalRegion()`, と `GetLocalVoxelSize()`.

6.10.5.4 `int cpm_VoxelInfo::m_neighborRankID[6]` [private]

隣接ランク番号 (外部境界は負の値)

`cpm_VoxelInfo.h` の 167 行で定義されています。

参照元 `cpm_VoxelInfo()`, `CreateNeighborRankInfo()`, `GetNeighborRankID()`, `IsInnerBoundary()`, と `IsOuterBoundary()`.

6.10.5.5 `int cpm_VoxelInfo::m_nRank` [private]

コミュニケーター内のランク数 (=プロセス並列数)

`cpm_VoxelInfo.h` の 165 行で定義されています。

参照元 `cpm_VoxelInfo()`, と `Init()`.

6.10.5.6 `int cpm_VoxelInfo::m_periodicRankID[6]` [private]

周期境界の隣接ランク番号

`cpm_VoxelInfo.h` の 168 行で定義されています。

参照元 `cpm_VoxelInfo()`, `CreateNeighborRankInfo()`, と `GetPeriodicRankID()`.

6.10.5.7 `int* cpm_VoxelInfo::m_rankMap` [private]

ランクマップ

`cpm_VoxelInfo.h` の 170 行で定義されています。

参照元 `cpm_VoxelInfo()`, `CreateLocalDomainInfo()`, `CreateNeighborRankInfo()`, `CreateRankMap()`, と `~cpm_VoxelInfo()`.

6.10.5.8 `int cpm_VoxelInfo::m_rankNo` [private]

コミュニケーター内でのランク番号

`cpm_VoxelInfo.h` の 166 行で定義されています。

参照元 `cpm_VoxelInfo()`, `CreateLocalDomainInfo()`, `CreateNeighborRankInfo()`, と `Init()`.

6.10.5.9 `int cpm_VoxelInfo::m_voxelHeadIndex[3]` [private]

自ランクの始点ボクセルインデックス

`cpm_VoxelInfo.h` の 160 行で定義されています。

参照元 `cpm_VoxelInfo()`, `CreateLocalDomainInfo()`, と `GetVoxelHeadIndex()`.

6.10.5.10 `int cpm_VoxelInfo::m_voxelTailIndex[3] [private]`

自ランクの終点ボクセルインデックス

cpm_VoxelInfo.h の 161 行で定義されています。

参照元 `cpm_VoxelInfo()`, `CreateLocalDomainInfo()`, と `GetVoxelTailIndex()`.

このクラスの説明は次のファイルから生成されました:

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

6.11 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

6.11.1 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。

6.11.2 コンストラクタとデストラクタ

6.11.2.1 `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`.

6.11.2.2 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.

6.11.3 変数

6.11.3.1 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().

6.11.3.2 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().

6.11.3.3 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().

6.11.3.4 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().

6.11.3.5 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().

6.11.3.6 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().

6.11.3.7 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().

6.11.3.8 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](#)

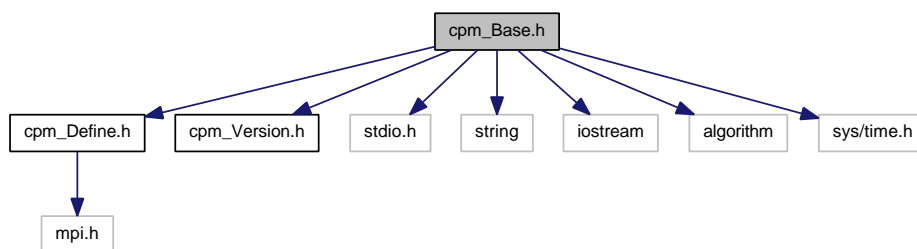
Chapter 7

ファイル

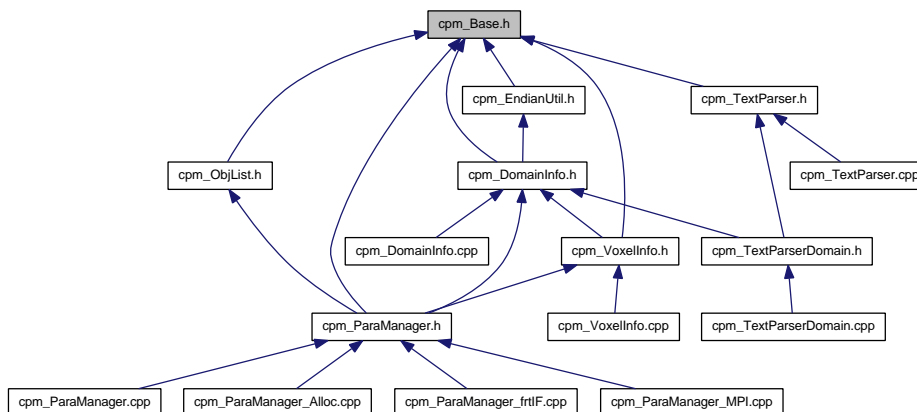
7.1 cpm_Base.h

```
#include "cpm_Define.h"  
#include "cpm_Version.h"  
#include <stdio.h>  
#include <string>  
#include <iostream>  
#include <algorithm>  
#include <sys/time.h>
```

cpm_Base.h のインクルード依存関係図



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



構成

- class `cpm_Base`

マクロ定義

- `#define CPM_INLINE inline`

7.1.1 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。

7.1.2 マクロ定義

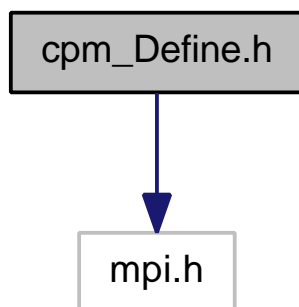
7.1.2.1 `#define CPM_INLINE inline`

`cpm_Base.h` の 38 行で定義されています。

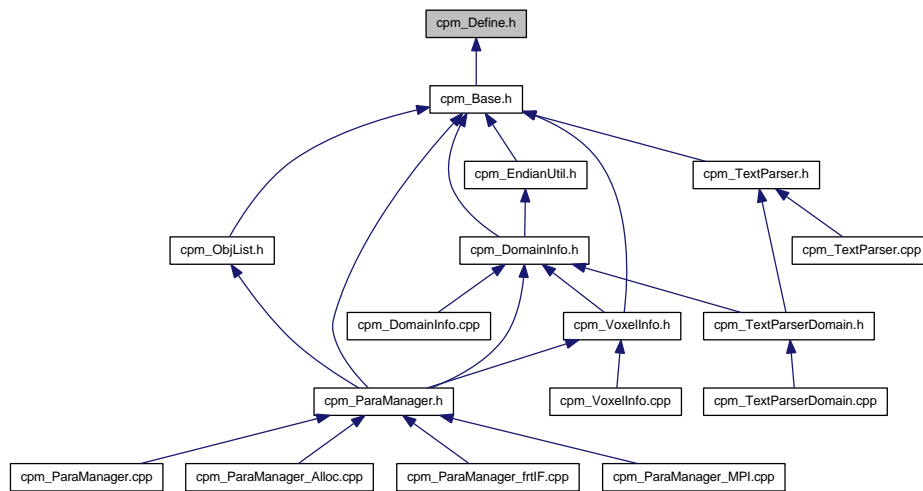
7.2 `cpm_Define.h`

```
#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_BUF_TYPE` double
- #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, 3, _NI, _NJ, _NK, _VC`))

列挙型

- enum `cpm_FaceFlag` {
`X_MINUS = 0, X_PLUS = 1, Y_MINUS = 2, Y_PLUS = 3,`
`Z_MINUS = 4, Z_PLUS = 5` }
- enum `cpm_DirFlag` { `X_DIR = 0, Y_DIR = 1, Z_DIR = 2` }
- enum `cpm_PMFlag` { `PLUS2MINUS = 0, MINUS2PLUS = 1, BOTH = 2` }
- enum `cpm_ErrorCode` {
`CPM_SUCCESS = 0, CPM_ERROR = 1000, CPM_ERROR_PM_INSTANCE = 1001, CPM_ERROR_INVALID_PTR = 1002,`
`CPM_ERROR_INVALID_DOMAIN_NO = 1003, CPM_ERROR_INVALID_OBJKEY = 1004, CPM_ERROR_REGIST_OBJKEY = 1005,`
`CPM_ERROR_TEXTPARSER = 2000, CPM_ERROR_NO_TEXTPARSER = 2001, CPM_ERROR_TP_NOVECTOR = 2002, CPM_ERROR_TP_VECTOR_SIZE = 2003,`
`CPM_ERROR_TP_INVALID_G_ORG = 2004, CPM_ERROR_TP_INVALID_G_VOXEL = 2005, CPM_ERROR_TP_INVALID_G_PITCH = 2006,`
`CPM_ERROR_TP_INVALID_G_RGN = 2007, CPM_ERROR_TP_INVALID_G_DIV = 2008,`
`CPM_ERROR_TP_INVALID_POS = 2009, CPM_ERROR_VOXELINIT = 3000, CPM_ERROR_NOT_IN_PROCGROUP = 3001,`
`CPM_ERROR_ALREADY_VOXELINIT = 3002, CPM_ERROR_MISMATCH_NP_SUBDOMAIN = 3003, CPM_ERROR_CREATE_RANKMAP = 3004,`
`CPM_ERROR_CREATE_NEIGHBOR = 3005, CPM_ERROR_CREATE_LOCALDOMAIN = 3006,`
`CPM_ERROR_INSERT_VOXELMAP = 3007, CPM_ERROR_CREATE_PROCGROUP = 3008, CPM_ERROR_INVALID_VOXELMAP = 3009,`
`CPM_ERROR_INVALID_REGION = 3010, CPM_ERROR_INVALID_DIVNUM = 3011, CPM_ERROR_OPEN_SBDM = 3012, CPM_ERROR_READ_SBDM_HEADER = 3013,`
`CPM_ERROR_READ_SBDM_FORMAT = 3014, CPM_ERROR_READ_SBDM_DIV = 3015, CPM_ERROR_READ_SBDM_CONTENTS = 3016, CPM_ERROR_SBDM_NUMDO = 3017,`
`CPM_ERROR_MISMATCH_DIV_SUBDOMAIN = 3018, CPM_ERROR_GET_INFO = 4000, CPM_ERROR_GET_DIVNUM = 4001, CPM_ERROR_GET_PITCH =`

```

4002, CPM_ERROR_GET_GLOBALVOXELSIZE = 4003,
CPM_ERROR_GET_GLOBALORIGIN = 4004, CPM_ERROR_GET_GLOBALREGION = 4005, CPM_ERROR_GET_LOCALV
= 4006, CPM_ERROR_GET_LOCALORIGIN = 4007,
CPM_ERROR_GET_LOCALREGION = 4008, CPM_ERROR_GET_DIVPOS = 4009, CPM_ERROR_GET_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_MPI_DIMSCREATE = 9016,
CPM_ERROR_BNDCOMM = 9500, CPM_ERROR_BNDCOMM_VOXELSIZE = 9501, CPM_ERROR_BNDCOMM_BUFFER
= 9502, CPM_ERROR_BNDCOMM_BUFFERLENGTH = 9503,
CPM_ERROR_PERIODIC = 9600, CPM_ERROR_PERIODIC_INVALID_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 }

```

7.2.1 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。

7.2.2 マクロ定義

7.2.2.1 #define _IDX_S3D(_I, _J, _K, _NI, _NJ, _NK, _VC)

値:

```

( (long long) (_K+_VC) * (long long) (_NI+2*_VC) * (long long) (_NJ+2*_VC) \
+ (long long) (_J+_VC) * (long long) (_NI+2*_VC) \
+ (long long) (_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 45 行で定義されています。

参照元 cpm_VoxelInfo::CreateLocalDomainInfo(), cpm_VoxelInfo::CreateNeighborRankInfo(), cpm_VoxelInfo::CreateRankMap(), と cpm_ParaManager::GetBndIndexExtGc().

7.2.2.2 #define _IDX_S4D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC)

値:

```
( (long long) (_N) * (long long) (_NI+2*_VC) * (long long) (_NJ+2*_VC) * (long
long) (_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 62 行で定義されています。

参照元 cpm_ParaManager::packX(), cpm_ParaManager::packY(), cpm_ParaManager::packZ(), cpm_ParaManager::unpackX(), cpm_ParaManager::unpackY(), と cpm_ParaManager::unpackZ().

7.2.2.3 #define _IDX_S4DEX(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC)

値:

```
( (long long) (_NN) * _IDX_S3D(_I, _J, _K, _NI, _NJ, _NK, _VC) \
+ (long long) (_N) )
```

4 次元インデクス (n,i,j,k) -> 1 次元インデクス変換マクロ

引数

in	<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 91 行で定義されています。

参照元 `cpm_ParaManager::packXEx()`, `cpm_ParaManager::packYEx()`, `cpm_ParaManager::packZEx()`, `cpm_ParaManager::unpackXEx()`, `cpm_ParaManager::unpackYEx()`, と `cpm_ParaManager::unpackZEx()`.

7.2.2.4 `#define _IDX_V3D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC) (_IDX_S4D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC))`

3 次元インデクス (i,j,k,3) -> 1 次元インデクス変換マクロ

引数

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

cpm_Define.h の 77 行で定義されています。

7.2.2.5 `#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 105 行で定義されています。

7.2.2.6 #define REAL_BUF_TYPE double

袖通信バッファの型指定

- デフォルトでは、REAL_BUF_TYPE=double
- コンパイル時オプション-D_BUFSIZE_FLOAT_を付与することで REAL_BUF_TYPE=float 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 31 行で定義されています。

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

7.2.3 列挙型

7.2.3.1 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 228 行で定義されています。

7.2.3.2 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 119 行で定義されています。

7.2.3.3 enum cpm_ErrorCode

CPM のエラーコード

列挙型の値:

CPM_SUCCESS 正常終了
CPM_ERROR その他のエラー
CPM_ERROR_PM_INSTANCE 並列管理クラス cpm_ParaManager のインスタンス失敗
CPM_ERROR_INVALID_PTR ポインタのエラー
CPM_ERROR_INVALID_DOMAIN_NO 領域番号が不正
CPM_ERROR_INVALID_OBJKEY 指定登録番号のオブジェクトが存在しない
CPM_ERROR_REGIST_OBJKEY オブジェクト登録に失敗:
CPM_ERROR_TEXTPARSER テキストパーサーに関するエラー
CPM_ERROR_NO_TEXTPARSER テキストパーサーを組み込んでいない
CPM_ERROR_TP_NOVECTOR 領域分割情報ファイルのベクトルデータ読み込みエラー
CPM_ERROR_TP_VECTOR_SIZE 領域分割情報ファイルのベクトルデータのサイズが不正
CPM_ERROR_TP_INVALID_G_ORG 領域分割情報ファイルのドメイン原点情報が不正
CPM_ERROR_TP_INVALID_G_VOXEL 領域分割情報ファイルのドメインVOXEL 数情報が不正
CPM_ERROR_TP_INVALID_G_PITCH 領域分割情報ファイルのドメインピッチ情報が不正
CPM_ERROR_TP_INVALID_G_RGN 領域分割情報ファイルのドメイン空間サイズ情報が不正
CPM_ERROR_TP_INVALID_G_DIV 領域分割情報ファイルのドメイン領域分割数情報が不正
CPM_ERROR_TP_INVALID_POS 領域分割情報ファイルのサブドメイン位置情報が不正
CPM_ERROR_VOXELINIT Voxellnit でエラー
CPM_ERROR_NOT_IN_PROCGROUP 自ランクがプロセスグループに含まれていない
CPM_ERROR_ALREADY_VOXELINIT 指定されたプロセスグループが既に領域分割済み:
CPM_ERROR_MISMATCH_NP_SUBDOMAIN 並列数とサブドメイン数が一致していない
CPM_ERROR_CREATE_RANKMAP ランクマップ生成に失敗
CPM_ERROR_CREATE_NEIGHBOR 隣接ランク情報生成に失敗
CPM_ERROR_CREATE_LOCALDOMAIN ローカル領域情報生成に失敗
CPM_ERROR_INSERT_VOXELMAP 領域情報のマップへの登録失敗
CPM_ERROR_CREATE_PROCGROUP プロセスグループ生成に失敗
CPM_ERROR_INVALID_VOXELSIZE VOXEL 数が不正
CPM_ERROR_INVALID_REGION 全体空間サイズが不正
CPM_ERROR_INVALID_DIVNUM 領域分割数が不正
CPM_ERROR_OPEN_SBDM ActiveSubdomain ファイルのオープンに失敗
CPM_ERROR_READ_SBDM_HEADER ActiveSubdomain ファイルのヘッダー読み込みに失敗
CPM_ERROR_READ_SBDM_FORMAT ActiveSubdomain ファイルのフォーマットエラー
CPM_ERROR_READ_SBDM_DIV ActiveSubdomain ファイルの領域分割数読み込みに失敗
CPM_ERROR_READ_SBDM_CONTENTS ActiveSubdomain ファイルのContents 読み込みに失敗
CPM_ERROR_SBDM_NUMDOMAIN_ZERO ActiveSubdomain ファイルの活性ドメイン数が 0.
CPM_ERROR_MISMATCH_DIV_SUBDOMAIN 領域分割数がActiveSubdomain ファイルと一致していない

CPM_ERROR_GET_INFO 情報取得系関数でエラー
CPM_ERROR_GET_DIVNUM 領域分割数の取得エラー
CPM_ERROR_GET_PITCH ピッチの取得エラー
CPM_ERROR_GET_GLOBALVOXELSIZE 全体ボクセル数の取得エラー

CPM_ERROR_GET_GLOBALORIGIN 全体空間の原点の取得エラー
CPM_ERROR_GET_GLOBALREGION 全体空間サイズの取得エラー
CPM_ERROR_GET_LOCALVOXELSIZE 自ランクのボクセル数の取得エラー
CPM_ERROR_GET_LOCALORIGIN 自ランクの空間原点の取得エラー
CPM_ERROR_GET_LOCALREGION 自ランクの空間サイズの取得エラー
CPM_ERROR_GET_DIVPOS 自ランクの領域分割位置の取得エラー
CPM_ERROR_GET_HEADINDEX 始点インデックスの取得エラー
CPM_ERROR_GET_TAILINDEX 終点インデックスの取得エラー
CPM_ERROR_GET_NEIGHBOR_RANK 隣接ランク番号の取得エラー
CPM_ERROR_GET_PERIODIC_RANK 周期境界位置の隣接ランク番号の取得エラー
CPM_ERROR_GET_MYRANK ランク番号の取得エラー
CPM_ERROR_GET_NUMRANK ランク数の取得エラー
CPM_ERROR_MPI MPI のエラー
CPM_ERROR_NO_MPI_INIT MPI_Init がコールされていない
CPM_ERROR_MPI_BARRIER MPI_Barrier でエラー
CPM_ERROR_MPI_BCAST MPI_Bcast でエラー
CPM_ERROR_MPI_SEND MPI_Send でエラー
CPM_ERROR_MPI_RECV MPI_Recv でエラー
CPM_ERROR_MPI_ISEND MPI_Isend でエラー
CPM_ERROR_MPI_Irecv MPI_Irecv でエラー
CPM_ERROR_MPI_WAIT MPI_Wait でエラー
CPM_ERROR_MPI_WAITALL MPI_Waitall でエラー
CPM_ERROR_MPI_ALLREDUCE MPI_Allreduce でエラー
CPM_ERROR_MPI_GATHER MPI_Gather でエラー
CPM_ERROR_MPI_ALLGATHER MPI_Allgather でエラー
CPM_ERROR_MPI_GATHERV MPI_Gatherv でエラー
CPM_ERROR_MPI_ALLGATHERV MPI_Allgatherv でエラー
CPM_ERROR_MPI_DIMSCREATE MPI_Dims_create でエラー
CPM_ERROR_BNDCOMM BndComm でエラー
CPM_ERROR_BNDCOMM_VOXELSIZE VoxelSize 取得でエラー
CPM_ERROR_BNDCOMM_BUFFER 袖通信バッファ取得でエラー
CPM_ERROR_BNDCOMM_BUFFERLENGTH 袖通信バッファサイズが足りない
CPM_ERROR_PERIODIC PeriodicComm でエラー
CPM_ERROR_PERIODIC_INVALID_DIR 不正な軸方向フラグが指定された
CPM_ERROR_PERIODIC_INVALID_PM 不正な正負方向フラグが指定された
CPM_ERROR_MPI_INVALID_COMM MPI コミュニケータが不正
CPM_ERROR_MPI_INVALID_DATATYPE 対応しない型が指定された
CPM_ERROR_MPI_INVALID_OPERATOR 対応しないオペレータが指定された
CPM_ERROR_MPI_INVALID_REQUEST 不正なリクエストが指定された

cpm_Define.h の 135 行で定義されています。

7.2.3.4 enum cpm_FaceFlag

面フラグ

列挙型の値:

X_MINUS -X face

X_PLUS +X face

Y_MINUS -Y face

Y_PLUS +Y face

Z_MINUS -Z face

Z_PLUS +Z face

cpm_Define.h の 108 行で定義されています。

7.2.3.5 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 255 行で定義されています。

7.2.3.6 enum cpm_PMFlag

方向フラグ

列挙型の値:

PLUS2MINUS plus -> minus direction

MINUS2PLUS minus -> plus direction

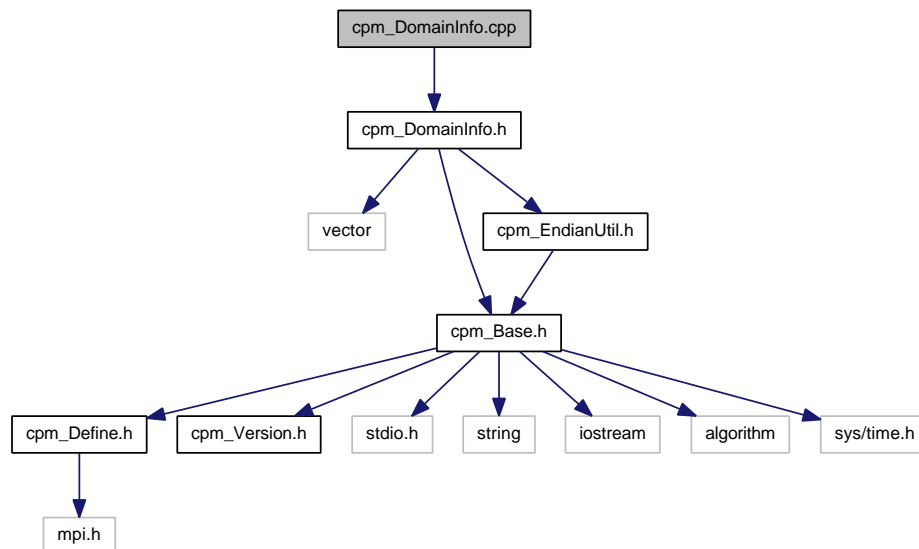
BOTH plus <-> minus direction

cpm_Define.h の 127 行で定義されています。

7.3 cpm_DomainInfo.cpp

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

cpm_DomainInfo.cpp のインクルード依存関係図



7.3.1 説明

DomainInfo クラスのソースファイル

作者

University of Tokyo

日付

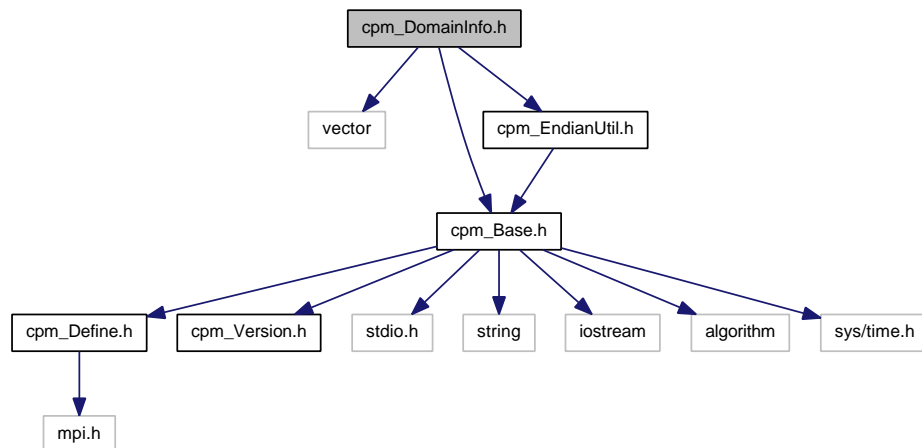
2012/05/31

[cpm_DomainInfo.cpp](#) で定義されています。

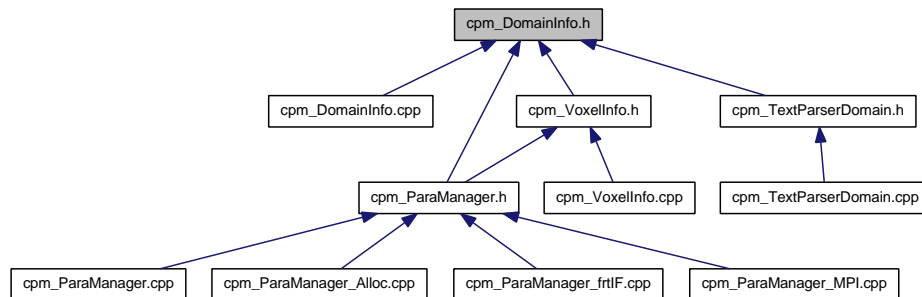
7.4 cpm_DomainInfo.h

```
#include <vector>
#include "cpm_Base.h"
#include "cpm_EndianUtil.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](#)

7.4.1 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

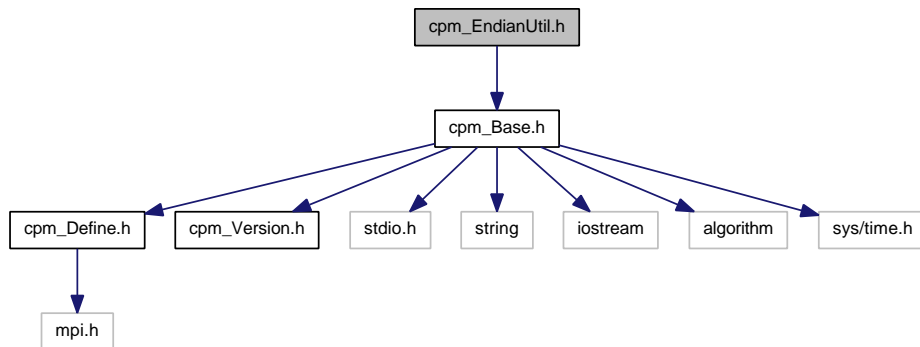
2012/05/31

[cpm_DomainInfo.h](#) で定義されています。

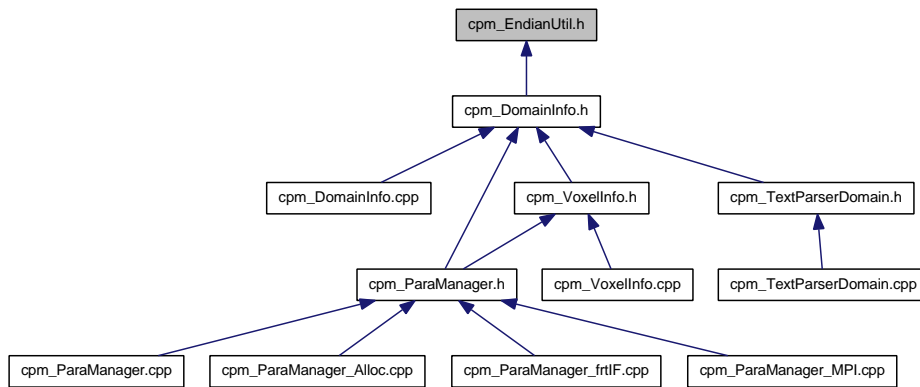
7.5 cpm_EndianUtil.h

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

cpm_EndianUtil.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。



ネームスペース

- namespace `CPM_ENDIAN`

列挙型

- enum `CPM_ENDIAN::EMatchType` { `CPM_ENDIAN::UnKnown` = 0, `CPM_ENDIAN::Match` = 1, `CPM_ENDIAN::UnMatch` = 2 }

関数

- template<class X >
`CPM_INLINE` void `CPM_ENDIAN::BSWAP16` (X &x)
- template<class X >
`CPM_INLINE` void `CPM_ENDIAN::BSWAP32` (X &x)
- template<class X >
`CPM_INLINE` void `CPM_ENDIAN::BSWAP64` (X &x)
- template<class X, class Y >
`CPM_INLINE` void `CPM_ENDIAN::SBSWAPVEC` (X *a, Y n)
- template<class X, class Y >
`CPM_INLINE` void `CPM_ENDIAN::BSWAPVEC` (X *a, Y n)

- `template<class X , class Y >`
`CPM_INLINE void CPM_ENDIAN::DBSWAPVEC (X *a, Y n)`

7.5.1 説明

CPM エンディアンユーティリティヘッダーファイル

作者

University of Tokyo

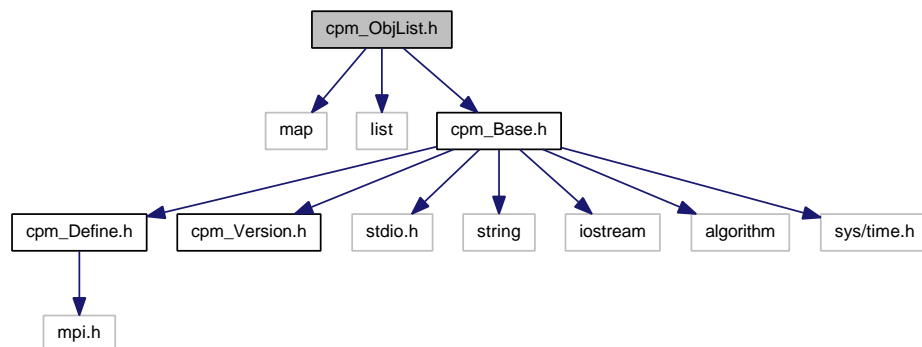
日付

2013/04/02

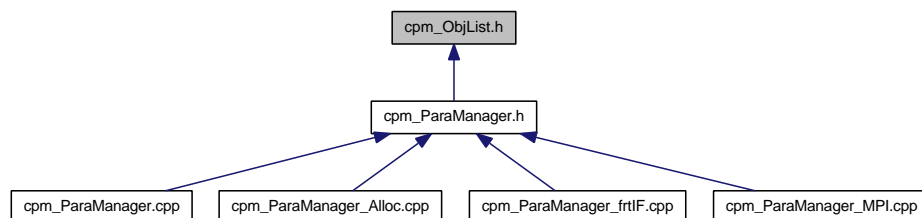
`cpm_EndianUtil.h` で定義されています。

7.6 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`

7.6.1 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。

7.6.2 型定義

7.6.2.1 `typedef std::map<int, int*> RankNoMap`

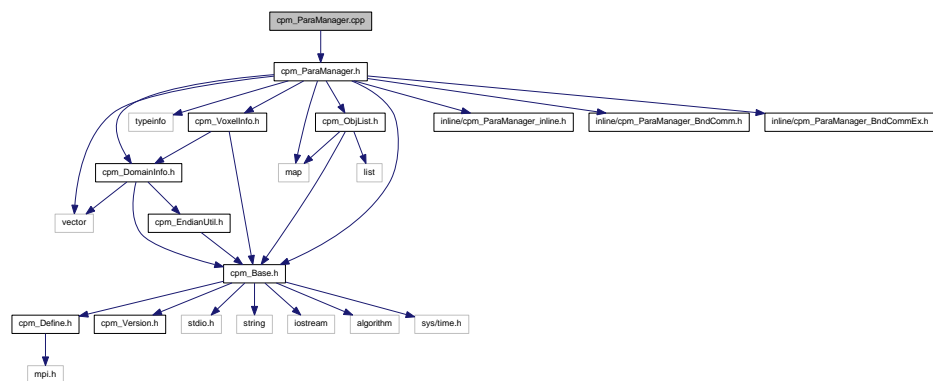
プロセスグループ毎のランク番号マップ

`cpm_ObjList.h` の 24 行で定義されています。

7.7 cpm_ParaManager.cpp

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

`cpm_ParaManager.cpp` のインクルード依存関係図



7.7.1 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

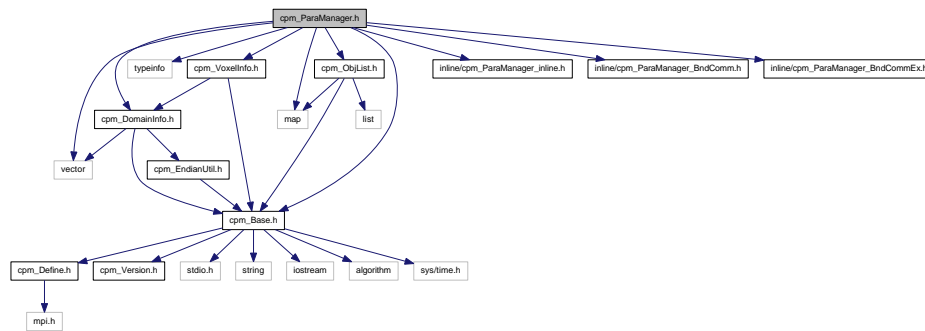
日付

2012/05/31

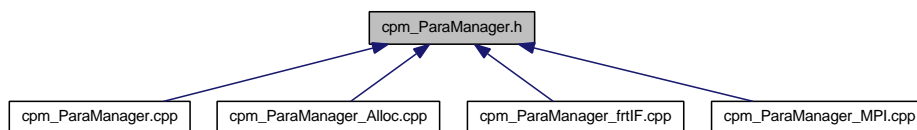
[cpm_ParaManager.cpp](#) で定義されています。

7.8 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](#)

7.8.1 説明

パラレルマネージャクラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_ParaManager.h](#) で定義されています。

7.8.2 型定義

7.8.2.1 typedef std::map<int, S_BNDCOMM_BUFFER*> BndCommInfoMap

プロセスグループ毎の袖通信バッファ情報

cpm_ParaManager.h の 68 行で定義されています。

7.8.2.2 typedef std::map<int, int*> RankNoMap

プロセスグループ毎のランク番号マップ

cpm_ParaManager.h の 30 行で定義されています。

7.8.2.3 typedef std::map<int, cpm_VoxelInfo*> VoxelInfoMap

プロセスグループ毎のVOXEL 空間情報管理マップ

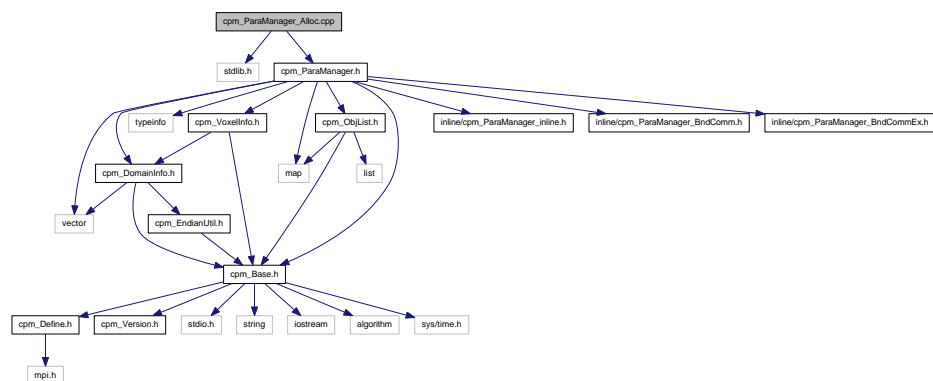
cpm_ParaManager.h の 27 行で定義されています。

7.9 cpm_ParaManager_Alloc.cpp

```
#include <stdlib.h>
```

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

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



7.9.1 説明

パラレルマネージャクラスのソースファイル

作者

University of Tokyo

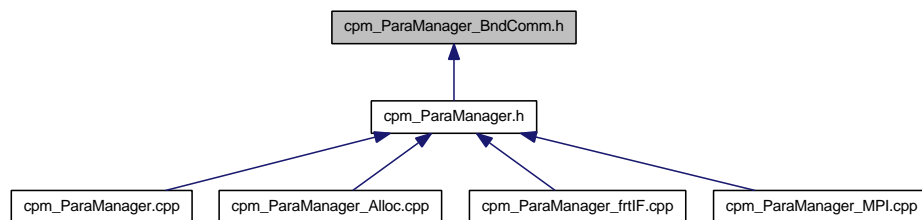
日付

2012/05/31

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

7.10 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)`

7.10.1 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_BndComm.h](#) で定義されています。

7.10.2 マクロ定義

7.10.2.1 `#define _IDXFX(_I, _J, _K, _N, _IS, _NJ, _NK, _VC)`

値:

```
( size_t(_N)      * size_t(_VC) * size_t(_NJ+2*_VC) * size_t(_NK+2*_VC) \
+ size_t(_K+_VC) * size_t(_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_VC) \
+ size_t(_I-(_IS)) \
)
```

cpm_ParaManager_BndComm.h の 18 行で定義されています。

参照元 cpm_ParaManager::packX(), と cpm_ParaManager::unpackX().

7.10.2.2 #define _IDXFY(_I, _J, _K, _N, _NI, _JS, _NK, _VC)

値:

```
( size_t(_N)      * size_t(_NI+2*_VC) * size_t(_VC) * size_t(_NK+2*_VC) \
+ size_t(_K+_VC) * size_t(_NI+2*_VC) * size_t(_VC) \
+ size_t(_J-(_JS)) * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
)
```

cpm_ParaManager_BndComm.h の 25 行で定義されています。

参照元 cpm_ParaManager::packY(), と cpm_ParaManager::unpackY().

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

値:

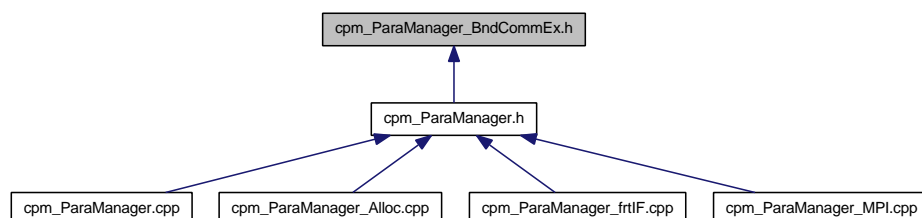
```
( size_t(_N)      * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) * size_t(_VC) \
+ size_t(_K-(_KS)) * size_t(_NI+2*_VC) * size_t(_NJ+2*_VC) \
+ size_t(_J+_VC) * size_t(_NI+2*_VC) \
+ size_t(_I+_VC) \
)
```

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

参照元 cpm_ParaManager::packZ(), と cpm_ParaManager::unpackZ().

7.11 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 _IDAFX(_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)

7.11.1 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。

7.11.2 マクロ定義

7.11.2.1 #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。

参照元 [cpm_ParaManager::packXEx\(\)](#), と [cpm_ParaManager::unpackXEx\(\)](#).

7.11.2.2 #define _IDXFY(*_N*, *_I*, *_J*, *_K*, *_NN*, *_NI*, *_JS*, *_NK*, *_VC*)

値:

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

[cpm_ParaManager_BndCommEx.h](#) の 27 行で定義されています。

参照元 [cpm_ParaManager::packYEx\(\)](#), と [cpm_ParaManager::unpackYEx\(\)](#).

7.11.2.3 #define _IDXFZ(*_N*, *_I*, *_J*, *_K*, *_NN*, *_NI*, *_NJ*, *_KS*, *_VC*)

値:

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

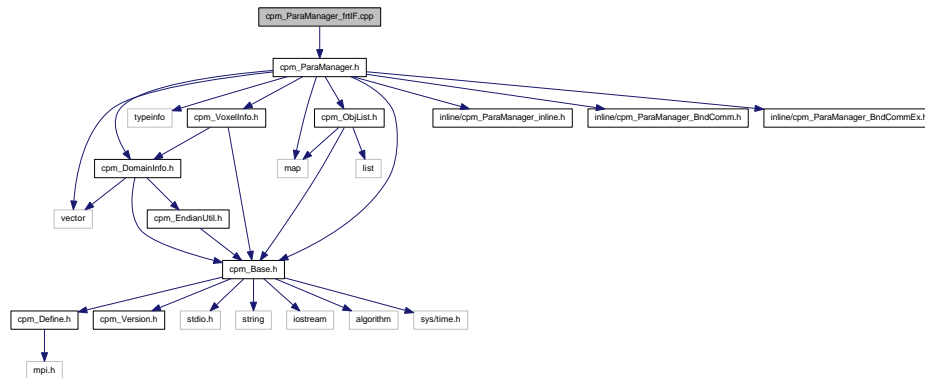
[cpm_ParaManager_BndCommEx.h](#) の 36 行で定義されています。

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

7.12 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_VoxelInit_ cpm_voxelinit_`
- `#define cpm_VoxelInit_nodiv_ cpm_voxelinit_nodiv_`
- `#define cpm_IsParallel_ cpm_isparallel_`
- `#define cpm_GetDivNum_ cpm_getdivnum_`
- `#define cpm_GetPitch_ cpm_getpitch_`
- `#define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_`
- `#define cpm_GetGlobalOrigin_ cpm_getglobalorigin_`
- `#define cpm_GetGlobalRegion_ cpm_getglobalregion_`
- `#define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_`
- `#define cpm_GetLocalOrigin_ cpm_getlocalorigin_`
- `#define cpm_GetLocalRegion_ cpm_getlocalregion_`
- `#define cpm_GetDivPos_ cpm_getdivpos_`
- `#define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_`
- `#define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_`
- `#define cpm_GetNeighborRankID_ cpm_getneighborrandid_`
- `#define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_`
- `#define cpm_GetMyRankID_ cpm_getmyrankid_`
- `#define cpm_GetNumRank_ cpm_getnumrank_`
- `#define cpm_Abort_ cpm_abort_`
- `#define cpm_Barrier_ cpm_barrier_`
- `#define cpm_Wait_ cpm_wait_`
- `#define cpm_Waitall_ cpm_waitall_`
- `#define cpm_Bcast_ cpm_bcast_`
- `#define cpm_Send_ cpm_send_`
- `#define cpm_Recv_ cpm_recv_`
- `#define cpm_Isend_ cpm_isend_`
- `#define cpm_Irecv_ cpm_irecv_`
- `#define cpm_Allreduce_ cpm_allreduce_`
- `#define cpm_Gather_ cpm_gather_`
- `#define cpm_Allgather_ cpm_allgather_`
- `#define cpm_Gatherv_ cpm_gatherv_`
- `#define cpm_Allgatherv_ cpm_allgatherv_`

- `#define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_`
- `#define cpm_BndCommsS3D_ cpm_bndcomms3d_`
- `#define cpm_BndCommV3D_ cpm_bndcommv3d_`
- `#define cpm_BndCommsS4D_ cpm_bndcomms4d_`
- `#define cpm_BndCommsS3D_nowait_ cpm_bndcomms3d_nowait_`
- `#define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_`
- `#define cpm_BndCommsS4D_nowait_ cpm_bndcomms4d_nowait_`
- `#define cpm_wait_BndCommsS3D_ cpm_wait_bndcomms3d_`
- `#define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_`
- `#define cpm_wait_BndCommsS4D_ cpm_wait_bndcomms4d_`
- `#define cpm_BndCommV3DEx_ cpm_bndcommv3dex_`
- `#define cpm_BndCommsS4DEx_ cpm_bndcomms4dex_`
- `#define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_`
- `#define cpm_BndCommsS4DEx_nowait_ cpm_bndcomms4dex_nowait_`
- `#define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_`
- `#define cpm_wait_BndCommsS4DEx_ cpm_wait_bndcomms4dex_`
- `#define cpm_PeriodicCommsS3D_ cpm_periodiccomms3d_`
- `#define cpm_PeriodicCommV3D_ cpm_periodiccommv3d_`
- `#define cpm_PeriodicCommsS4D_ cpm_periodiccomms4d_`
- `#define cpm_PeriodicCommV3DEx_ cpm_periodiccommv3dex_`
- `#define cpm_PeriodicCommsS4DEx_ cpm_periodiccomms4dex_`

関数

- `CPM_EXTERN void cpm_Initialize_ (int *ierr)`
- `CPM_EXTERN void cpm_Voxellnit_ (int *div, int *vox, double *origin, double *pitch, int *maxVC, int *maxN, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Voxellnit_nodiv_ (int *vox, double *origin, double *pitch, int *maxVC, int *maxN, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_IsParallel_ (int *ipara, int *ierr)`
- `CPM_EXTERN void cpm_GetDivNum_ (int *div, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetPitch_ (double *pch, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetGlobalVoxelSize_ (int *wsz, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetGlobalOrigin_ (double *worg, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetGlobalRegion_ (double *wrgn, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetLocalVoxelSize_ (int *lsz, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetLocalOrigin_ (double *lorg, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetLocalRegion_ (double *lrn, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetDivPos_ (int *pos, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetVoxelHeadIndex_ (int *idx, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetVoxelTailIndex_ (int *idx, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetNeighborRankID_ (int *nID, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetPeriodicRankID_ (int *nID, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetMyRankID_ (int *id, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_GetNumRank_ (int *nrank, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Abort_ (int *errorcode)`
- `CPM_EXTERN void cpm_Barrier_ (int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Wait_ (int *reqNo, int *ierr)`
- `CPM_EXTERN void cpm_Waitall_ (int *count, int *reqlist, int *ierr)`
- `CPM_EXTERN void cpm_Bcast_ (void *buf, int *count, int *datatype, int *root, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Send_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Recv_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Isend_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *reqNo, int *ierr)`

- `CPM_EXTERN void cpm_lrecv_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *reqNo, int *ierr)`
- `CPM_EXTERN void cpm_Allreduce_ (void *sendbuf, void *recvbuf, int *count, int *datatype, int *op, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Gather_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *root, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Allgather_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Gatherv_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnts, int *displs, int *recvtype, int *root, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_Allgatherv_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnts, int *displs, int *recvtype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_SetBndCommBuffer_ (int *maxVC, int *maxN, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndComms4D_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndComms3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndComms4D_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndComms3D_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3D_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndComms4D_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndComms3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndCommV3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndComms4DEx_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3DEx_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndComms4DEx_nowait_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_BndCommV3DEx_nowait_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndComms4DEx_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_wait_BndCommV3DEx_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *reqlist, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicComms4D_ (void *array, int *imax, int *jmax, int *kmax, int *nmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicComms3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicCommV3D_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicComms4DEx_ (void *array, int *nmax, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`
- `CPM_EXTERN void cpm_PeriodicCommV3DEx_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)`

7.12.1 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。

7.12.2 マクロ定義

7.12.2.1 `#define cpm_Abort_ cpm_abort_`

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

7.12.2.2 `#define cpm_Allgather_ cpm_allgather_`

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

7.12.2.3 `#define cpm_Allgatherv_ cpm_allgatherv_`

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

7.12.2.4 `#define cpm_Allreduce_ cpm_allreduce_`

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

7.12.2.5 `#define cpm_Barrier_ cpm_barrier_`

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

7.12.2.6 `#define cpm_Bcast_ cpm_bcast_`

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

7.12.2.7 `#define cpm_BndComms3D_ cpm_bndcomms3d_`

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

7.12.2.8 `#define cpm_BndComms3D_nowait_ cpm_bndcomms3d_nowait_`

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

7.12.2.9 #define cpm_BndCommsS4D_ cpm_bndcomms4d_

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

参照元 cpm_BndCommS3D_(), と cpm_BndCommV3D_().

7.12.2.10 #define cpm_BndCommsS4D_nowait_ cpm_bndcomms4d_nowait_

cpm_ParaManager_frtIF.cpp の 64 行で定義されています。

参照元 cpm_BndCommS3D_nowait_(), と cpm_BndCommV3D_nowait_().

7.12.2.11 #define cpm_BndCommsS4DEx_ cpm_bndcomms4dex_

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

参照元 cpm_BndCommV3DEx_().

7.12.2.12 #define cpm_BndCommS4DEx_nowait_ cpm_bndcomms4dex_nowait_

cpm_ParaManager_frtIF.cpp の 71 行で定義されています。

参照元 cpm_BndCommV3DEx_nowait_().

7.12.2.13 #define cpm_BndCommV3D_ cpm_bndcommv3d_

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

7.12.2.14 #define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_

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

7.12.2.15 #define cpm_BndCommV3DEx_ cpm_bndcommv3dex_

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

7.12.2.16 #define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_

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

7.12.2.17 #define CPM_EXTERN extern "C"

extern 宣言

cpm_ParaManager_frtIF.cpp の 17 行で定義されています。

7.12.2.18 #define cpm_Gather_ cpm_gather_

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

7.12.2.19 #define cpm_Gatherv_ cpm_gatherv_

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

7.12.2.20 `#define cpm_GetDivNum_ cpm_getdivnum_`

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

7.12.2.21 `#define cpm_GetDivPos_ cpm_getdivpos_`

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

7.12.2.22 `#define cpm_GetGlobalOrigin_ cpm_getglobalorigin_`

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

7.12.2.23 `#define cpm_GetGlobalRegion_ cpm_getglobalregion_`

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

7.12.2.24 `#define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_`

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

7.12.2.25 `#define cpm_GetLocalOrigin_ cpm_getlocalorigin_`

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

7.12.2.26 `#define cpm_GetLocalRegion_ cpm_getlocalregion_`

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

7.12.2.27 `#define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_`

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

7.12.2.28 `#define cpm_GetMyRankID_ cpm_getmyrankid_`

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

7.12.2.29 `#define cpm_GetNeighborRankID_ cpm_getneighborrankid_`

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

7.12.2.30 `#define cpm_GetNumRank_ cpm_getnumrank_`

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

7.12.2.31 `#define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_`

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

7.12.2.32 `#define cpm_GetPitch_ cpm_getpitch_`

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

7.12.2.33 `#define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_`

cpm_ParaManager_frtIF.cpp の 38 行で定義されています。

7.12.2.34 `#define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_`

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

7.12.2.35 `#define cpm_Initialize_ cpm_initialize_`

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

7.12.2.36 `#define cpm_Irecv_ cpm_irecv_`

cpm_ParaManager_frtIF.cpp の 52 行で定義されています。

7.12.2.37 `#define cpm_Isend_ cpm_isend_`

cpm_ParaManager_frtIF.cpp の 51 行で定義されています。

7.12.2.38 `#define cpm_IsParallel_ cpm_isparallel_`

cpm_ParaManager_frtIF.cpp の 28 行で定義されています。

7.12.2.39 `#define cpm_PeriodicCommsS3D cpm_periodiccomms3d_`

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

7.12.2.40 `#define cpm_PeriodicCommsS4D cpm_periodiccomms4d_`

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

7.12.2.41 `#define cpm_PeriodicCommS4DEx cpm_periodiccomms4dex_`

cpm_ParaManager_frtIF.cpp の 78 行で定義されています。

7.12.2.42 `#define cpm_PeriodicCommV3D cpm_periodiccommv3d_`

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

7.12.2.43 `#define cpm_PeriodicCommV3DEx cpm_periodiccommv3dex_`

cpm_ParaManager_frtIF.cpp の 77 行で定義されています。

7.12.2.44 #define cpm_Recv_ cpm_recv_

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

7.12.2.45 #define cpm_Send_ cpm_send_

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

7.12.2.46 #define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_

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

7.12.2.47 #define cpm_Voxellnit_ cpm_voxellnit_

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

7.12.2.48 #define cpm_Voxellnit_nodiv_ cpm_voxellnit_nodiv_

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

7.12.2.49 #define cpm_Wait_ cpm_wait_

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

7.12.2.50 #define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_

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

7.12.2.51 #define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_

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

参照元 cpm_wait_BndCommS3D_(), と cpm_wait_BndCommV3D_().

7.12.2.52 #define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_

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

参照元 cpm_wait_BndCommV3DEx_().

7.12.2.53 #define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_

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

7.12.2.54 #define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_

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

7.12.2.55 #define cpm_Waitall_ cpm_waitall_

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

7.12.3 関数

7.12.3.1 CPM_EXTERN void cpm_Abort_(int * errorcode)

Abort

- Abort のFortran インターフェイス関数

引数

in	errorcode	MPI_Abort に渡すエラーコード
----	-----------	---------------------

cpm_ParaManager_frtIF.cpp の 912 行で定義されています。

参照先 cpm_ParaManager::Abort(), と cpm_ParaManager::get_instance().

7.12.3.2 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	sendbuf	送信データ
in	sendcnt	送信データのサイズ
in	sendtype	送信データのデータタイプ (cpm_fparam.fi を参照)
out	recvbuf	受信データ
in	recvcnt	受信データのサイズ
in	recvtype	受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 (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1329 行で定義されています。

参照先 cpm_ParaManager::Allgather(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATA-
TYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_
Datatype().

7.12.3.3 CPM_EXTERN void cpm_Allgatherv_(void * sendbuf, int * sendcnt, int * sendtype, void * recvbuf, int * recvcnts, int * displs, int * recvtype, int * procGrpNo, int * ierr)

MPI_Allgatherv のFortran インターフェイス

- MPI_Allgatherv のFortran インターフェイス関数

引数

in	sendbuf	送信データ
in	sendcnt	送信データのサイズ
in	sendtype	送信データのデータタイプ (cpm_fparam.fi を参照)
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	recvtype	受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 (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1431 行で定義されています。

参照先 cpm_ParaManager::Allgather(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATA-
TYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_
Datatype().

7.12.3.4 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	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	datatype	送受信データのデータタイプ (cpm_fparam.fi を参照)
in	op	オペレータ
in	procGrpNo	プロセスグループ番号
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1228 行で定義されています。

参照先 cpm_ParaManager::Allreduce(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATA-
TYPE, CPM_ERROR_MPI_INVALID_OPERATOR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_
instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::GetMPI_Op().

7.12.3.5 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	procGrpNo	プロセスグループ番号
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 940 行で定義されています。

参照先 cpm_ParaManager::Barrier(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_
ParaManager::get_instance().

7.12.3.6 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

	inout]	buf 送受信バッファ
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	root	送信元のランク番号 (procGrpNo 内でのランク番号)
in	procGrpNo	プロセスグループ番号
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1029 行で定義されています。

参照先 cpm_ParaManager::Bcast(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.7 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 1559 行で定義されています。

参照先 cpm_ParaManager::BndComms3D(), cpm_BndComms4D_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.8 CPM_EXTERN void cpm_BndComms3D_nowait_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (imax,jmax,kmax) の形式の配列の袖通信を行う
- BndComms3D_nowait のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1701 行で定義されています。

参照先 cpm_ParaManager::cpm_BndComms3D_nowait(), cpm_BndComms4D_nowait_, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.12.3.9 **CPM_EXTERN** void **cpm_BndCommsS4D_**(void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *nmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

袖通信 (Scalar4D 版) のFortran インターフェイス

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う
- BndCommsS4D のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1514 行で定義されています。

参照先 cpm_ParaManager::BndCommsS4D(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.10 **CPM_EXTERN** void **cpm_BndCommsS4D_nowait_**(void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *nmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

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

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う
- BndCommsS4D_nowait のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1661 行で定義されています。

参照先 cpm_ParaManager::cpm_BndCommsS4D_nowait(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.12.3.11 **CPM_EXTERN** void **cpm_BndCommS4DEx_**(void * *array*, int * *nmax*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

袖通信 (Scalar4DEx 版) のFortran インターフェイス

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

引数

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

cpm_ParaManager_frtIF.cpp の 1924 行で定義されています。

参照先 cpm_ParaManager::BndCommS4DEx(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.12 **CPM_EXTERN** void **cpm_BndCommS4DEx_nowait_**(void * *array*, int * *nmax*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

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

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommS4DEx_nowait のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 2021 行で定義されています。

参照先 cpm_ParaManager::cpm_BndCommS4DEx_nowait(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.12.3.13 CPM_EXTERN void cpm_BndCommV3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * procGrpNo, int * ierr)

袖通信 (Vector3D 版) のFortran インターフェイス

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う
- BndCommV3D のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1609 行で定義されています。

参照先 cpm_ParaManager::BndCommV3D(), cpm_BndCommS4D_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.14 CPM_EXTERN void cpm_BndCommV3D_nowait_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う
- BndCommV3D_nowait のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1747 行で定義されています。

参照先 cpm_BndCommS4D_nowait_, cpm_ParaManager::cpm_BndCommV3D_nowait(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.12.3.15 CPM_EXTERN void cpm_BndCommV3DEx_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * procGrpNo, int * ierr)

袖通信 (Vector3DEx 版) のFortran インターフェイス

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommV3D のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1969 行で定義されています。

参照先 cpm_ParaManager::BndCommV3DEx(), cpm_BndCommS4DEx_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.16 CPM_EXTERN void cpm_BndCommV3DEx_nowait_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う
- BndCommV3D_nowait のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 2061 行で定義されています。

参照先 cpm_BndCommS4DEx_nowait_, cpm_ParaManager::cpm_BndCommV3DEx_nowait(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.12.3.17 CPM_EXTERN void cpm_Gather_ (void * sendbuf, int * sendcnt, int * sendtype, void * recvbuf, int * recvcnt, int * recvtype, int * root, int * procGrpNo, int * ierr)

MPI_Gather のFortran インターフェイス

- MPI_Gather のFortran インターフェイス関数

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ

in	<i>sendtype</i>	送信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>recvbuf</i>	受信データ
in	<i>recvcnt</i>	受信データのサイズ
in	<i>recvtype</i>	受信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>root</i>	受信するランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1279 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::Gather(), cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.18 CPM_EXTERN void cpm_Gatherv_ (void * *sendbuf*, int * *sendcnt*, int * *sendtype*, void * *recvbuf*, int * *recvcnts*, int * *displs*, int * *recvtype*, int * *root*, int * *procGrpNo*, int * *ierr*)

MPI_Gatherv のFortran インターフェイス

- MPI_Gatherv のFortran インターフェイス関数

引数

in	<i>sendbuf</i>	送信データ
in	<i>sendcnt</i>	送信データのサイズ
in	<i>sendtype</i>	送信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>recvbuf</i>	受信データ
in	<i>recvcnts</i>	各ランクからの受信データサイズ
in	<i>displs</i>	各ランクからの受信データ配置位置
in	<i>recvtype</i>	受信データのデータタイプ (cpm_fparam.fi を参照)
in	<i>root</i>	受信するランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1380 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::Gatherv(), cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.12.3.19 CPM_EXTERN void cpm_GetDivNum_ (int * *div*, int * *procGrpNo*, int * *ierr*)

領域分割数を取得

- GetDivNum のFortran インターフェイス関数

引数

out	<i>div</i>	領域分割数 (3word の整数配列)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 284 行で定義されています。

参照先 CPM_ERROR_GET_DIVNUM, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetDivNum().

7.12.3.20 CPM_EXTERN void cpm_GetDivPos_(int * pos, int * procGrpNo, int * ierr)

自ランクの領域分割位置を取得

- GetDivPos のFortran インターフェイス関数

引数

in	procGrpNo	プロセスグループ番号
out	pos	自ランクの領域分割位置 (3word の整数配列)
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 620 行で定義されています。

参照先 CPM_ERROR_GET_DIVPOS, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetDivPos().

7.12.3.21 CPM_EXTERN void cpm_GetGlobalOrigin_(double * worg, int * procGrpNo, int * ierr)

全体空間の原点を取得

- GetGlobalOrigin のFortran インターフェイス関数

引数

in	procGrpNo	プロセスグループ番号
out	worg	全体空間の原点 (3word の実数配列)
out	ierr	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 410 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALORIGIN, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetGlobalOrigin().

7.12.3.22 CPM_EXTERN void cpm_GetGlobalRegion_(double * wrgn, int * procGrpNo, int * ierr)

全体空間サイズを取得

- GetGlobalRegion のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 452 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALREGION, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetGlobalRegion().

7.12.3.23 CPM_EXTERN void cpm_GetGlobalVoxelSize_(int * wsz, int * procGrpNo, int * ierr)

全体ボクセル数を取得

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

引数

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

cpm_ParaManager_frtIF.cpp の 368 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALVOXELSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetGlobalVoxelSize().

7.12.3.24 CPM_EXTERN void cpm_GetLocalOrigin_ (double * lorg, int * procGrpNo, int * ierr)

自ランクの空間原点を取得

- GetLocalOrigin のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 536 行で定義されています。

参照先 CPM_ERROR_GET_LOCALORIGIN, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetLocalOrigin().

7.12.3.25 CPM_EXTERN void cpm_GetLocalRegion_ (double * lrgn, int * procGrpNo, int * ierr)

自ランクの空間サイズを取得

- GetLocalRegion のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 578 行で定義されています。

参照先 CPM_ERROR_GET_LOCALREGION, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetLocalOrigin().

7.12.3.26 CPM_EXTERN void cpm_GetLocalVoxelSize_ (int * lsz, int * procGrpNo, int * ierr)

自ランクのボクセル数を取得

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

引数

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

cpm_ParaManager_frtIF.cpp の 494 行で定義されています。

参照先 CPM_ERROR_GET_LOCALVOXELSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetLocalVoxelSize().

7.12.3.27 CPM_EXTERN void cpm_GetMyRankID_ (int * *id*, int * *procGrpNo*, int * *ierr*)

ランク番号の取得

- GetMyRankID のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 838 行で定義されています。

参照先 CPM_ERROR_GET_MYRANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMyRankID().

7.12.3.28 CPM_EXTERN void cpm_GetNeighborRankID_ (int * *nID*, int * *procGrpNo*, int * *ierr*)

自ランクの隣接ランク番号を取得

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

引数

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

cpm_ParaManager_frtIF.cpp の 748 行で定義されています。

参照先 CPM_ERROR_GET_NEIGHBOR_RANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetNeighborRankID().

7.12.3.29 CPM_EXTERN void cpm_GetNumRank_ (int * *nrank*, int * *procGrpNo*, int * *ierr*)

ランク数の取得

- GetNumRank のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 876 行で定義されています。

参照先 CPM_ERROR_GET_NUMRANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetNumRank().

7.12.3.30 CPM_EXTERN void cpm_GetPeriodicRankID_ (int * *nID*, int * *procGrpNo*, int * *ierr*)

自ランクの周期境界の隣接ランク番号を取得

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

引数

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

cpm_ParaManager_frtIF.cpp の 793 行で定義されています。

参照先 CPM_ERROR_GET_PERIODIC_RANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetPeriodicRankID().

7.12.3.31 CPM_EXTERN void cpm_GetPitch_ (double * *pch*, int * *procGrpNo*, int * *ierr*)

ピッチを取得

- GetPitch のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 326 行で定義されています。

参照先 CPM_ERROR_GET_PITCH, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetPitch().

7.12.3.32 CPM_EXTERN void cpm_GetVoxelHeadIndex_ (int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの始点VOXEL の全体空間でのインデクスを取得

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

引数

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

cpm_ParaManager_frtIF.cpp の 663 行で定義されています。

参照先 CPM_ERROR_GET_HEADINDEX, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetVoxelHeadIndex().

7.12.3.33 CPM_EXTERN void cpm_GetVoxelTailIndex_ (int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの終点VOXEL の全体空間でのインデクスを取得

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

引数

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

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

参照先 CPM_ERROR_GET_TAILINDEX, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetVoxelTailIndex().

7.12.3.34 CPM_EXTERN void cpm_Initialize_ (int * *ierr*)

初期化処理 (MPI_Init は実行済みの場合)

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

引数

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

cpm_ParaManager_frtIF.cpp の 144 行で定義されています。

参照先 CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::Initialize().

7.12.3.35 CPM_EXTERN void cpm_Irecv_ (void * *buf*, int * *count*, int * *datatype*, int * *source*, int * *procGrpNo*, int * *reqNo*, int * *ierr*)

Irecv

- Irecv のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1189 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Irecv(), CPM_SUCCESS, と cpm_ParaManager::get_instance().

7.12.3.36 CPM_EXTERN void cpm_Isend_ (void * *buf*, int * *count*, int * *datatype*, int * *dest*, int * *procGrpNo*, int * *reqNo*, int * *ierr*)

Isend

- Isend のFortran インターフェイス関数

引数

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

in	<i>datatype</i>	データタイプ (fparam.fi を参照)
in	<i>dest</i>	送信先のランク番号 (procGrpNo 内でのランク番号)
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>reqNo</i>	リクエスト番号
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1150 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Isend(), CPM_SUCCESS, と cpm_ParaManager::get_instance().

7.12.3.37 CPM_EXTERN void cpm_IsParallel_(int * ipara, int * ierr)

並列実行であるかチェックする

- IsParallel のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 250 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::IsParallel().

7.12.3.38 CPM_EXTERN void cpm_PeriodicCommS3D_(void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

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

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

引数

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

cpm_ParaManager_frtIF.cpp の 2258 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4D_(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS3D().

7.12.3.39 CPM_EXTERN void cpm_PeriodicComms4D_(void * array, int * imax, int * jmax, int * kmax, int * nmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

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

- (imax,jmax,kmax,nmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicComms4D のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 2195 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicComms4D().

参照元 cpm_PeriodicCommS3D_(), と cpm_PeriodicCommV3D_().

7.12.3.40 CPM_EXTERN void cpm_PeriodicComms4DEx_(void * array, int * nmax, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

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

- (nmax,imax,jmax,kmax) の形式の配列の周期境界方向の袖通信を行う
- PeriodicComms4DEx のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 2397 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicComms4DEx().

instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicComms4DEx().

参照元 cpm_PeriodicCommV3DEx_().

7.12.3.41 CPM_EXTERN void cpm_PeriodicCommV3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

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

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

引数

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

cpm_ParaManager_frtIF.cpp の 2327 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicComms4D_(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3D().

7.12.3.42 CPM_EXTERN void cpm_PeriodicCommV3DEx_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * dir, int * pm, int * datatype, int * procGrpNo, int * ierr)

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

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

引数

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

cpm_ParaManager_frtIF.cpp の 2460 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4DEx(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3DEx().

7.12.3.43 CPM_EXTERN void cpm_Recv_ (void * buf, int * count, int * datatype, int * source, int * procGrpNo, int * ierr)

Recv

- Recv のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1109 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::Recv().

7.12.3.44 CPM_EXTERN void cpm_Send_ (void * buf, int * count, int * datatype, int * dest, int * procGrpNo, int * ierr)

Send

- Send のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1069 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::Send().

7.12.3.45 CPM_EXTERN void cpm_SetBndCommBuffer_ (int * maxVC, int * maxN, int * procGrpNo, int * ierr)

袖通信バッファのセット (Fortran インターフェイス)

- 袖通信バッファ確保処理のFortran インターフェイス関数

引数

in	<i>maxVC</i>	送受信バッファの最大袖数
in	<i>maxN</i>	送受信バッファの最大成分数
in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1477 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::SetBndCommBuffer().

7.12.3.46 CPM_EXTERN void cpm_Voxellnit_ (int * div, int * vox, double * origin, double * pitch, int * maxVC, int * maxN, int * procGrpNo, int * ierr)

領域分割

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

引数

in	div	領域分割数 (サイズ 3)
in	vox	空間全体のボクセル数 (サイズ 3)
in	origin	空間全体の原点 (サイズ 3)
in	pitch	ボクセルピッチ (サイズ 3)
in	maxVC	最大の袖数 (袖通信用)
in	maxN	最大の成分数 (袖通信用)
in	procGrpNo	領域分割を行うプロセスグループ番号
out	ier	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

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

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::Voxellnit().

7.12.3.47 CPM_EXTERN void cpm_Voxellnit_nodiv_ (int * vox, double * origin, double * pitch, int * maxVC, int * maxN, int * procGrpNo, int * ierr)

領域分割

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

引数

in	vox	空間全体のボクセル数 (サイズ 3)
in	origin	空間全体の原点 (サイズ 3)
in	pitch	ボクセルピッチ (サイズ 3)
in	maxVC	最大の袖数 (袖通信用)
in	maxN	最大の成分数 (袖通信用)
in	procGrpNo	領域分割を行うプロセスグループ番号
out	ier	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 219 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::Voxellnit().

7.12.3.48 CPM_EXTERN void cpm_Wait_ (int * reqNo, int * ierr)

Wait

- Wait のFortran インターフェイス関数

引数

in	<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 968 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Wait(), と cpm_ParaManager::get_instance().

7.12.3.49 CPM_EXTERN void cpm_wait_BndCommS3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS3D のFortran インターフェイス関数

引数

	<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 1834 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_wait_BndCommS3D(), cpm_wait_BndCommS4D_, と cpm_ParaManager::get_instance().

7.12.3.50 CPM_EXTERN void cpm_wait_BndCommS4D_ (void * array, int * imax, int * jmax, int * kmax, int * nmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (imax,jmax,kmax,nmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS4D のFortran インターフェイス関数

引数

	<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 1794 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_wait_BndCommS4D(), と cpm_ParaManager::get_instance().

7.12.3.51 CPM_EXTERN void cpm_wait_BndCommS4DEx_ (void * array, int * nmax, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (nmax,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS4DEx のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 2108 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_wait_BndCommS4DEx(), と cpm_ParaManager::get_instance().

7.12.3.52 CPM_EXTERN void cpm_wait_BndCommV3D_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (imax,jmax,kmax,3) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommV3D のFortran インターフェイス関数

引数

	<i>inout]</i>	array 袖通信をする配列の先頭ポインタ
in	<i>imax</i>	配列サイズ (I 方向)
in	<i>jmax</i>	配列サイズ (J 方向)
in	<i>kmax</i>	配列サイズ (K 方向)
in	<i>vc</i>	仮想セル数
in	<i>vc_comm</i>	通信する仮想セル数
in	<i>datatype</i>	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	<i>reqlist</i>	リクエスト番号のリスト (サイズ 12)

in	<i>procGrpNo</i>	プロセスグループ番号
out	<i>ierr</i>	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1879 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_wait_BndCommS4D_, cpm_ParaManager::cpm_wait_BndCommV3D(), と cpm_ParaManager::get_instance().

7.12.3.53 CPM_EXTERN void cpm_wait_BndCommV3DEx_ (void * array, int * imax, int * jmax, int * kmax, int * vc, int * vc_comm, int * datatype, int * reqlist, int * procGrpNo, int * ierr)

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

- (3,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommV3DEx のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 2148 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_wait_BndCommS4DEx_, cpm_ParaManager::cpm_wait_BndCommV3DEx(), と cpm_ParaManager::get_instance().

7.12.3.54 CPM_EXTERN void cpm_Waitall_ (int * count, int * reqlist, int * ierr)

Waitall

- Waitall のFortran インターフェイス関数

引数

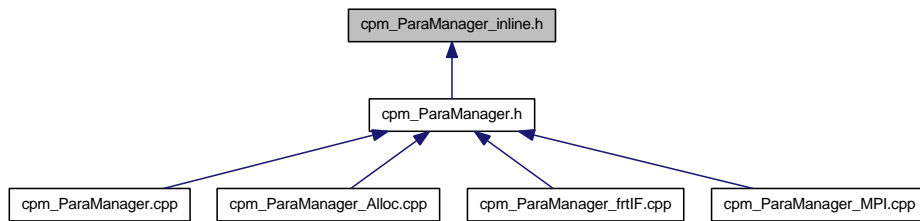
in	<i>count</i>	リクエストの数
in	<i>reqlist</i>	リクエスト番号のリスト (0 以上の整数)
out	<i>ierr</i>	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 997 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Waitall(), と cpm_ParaManager::get_instance().

7.13 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。



7.13.1 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

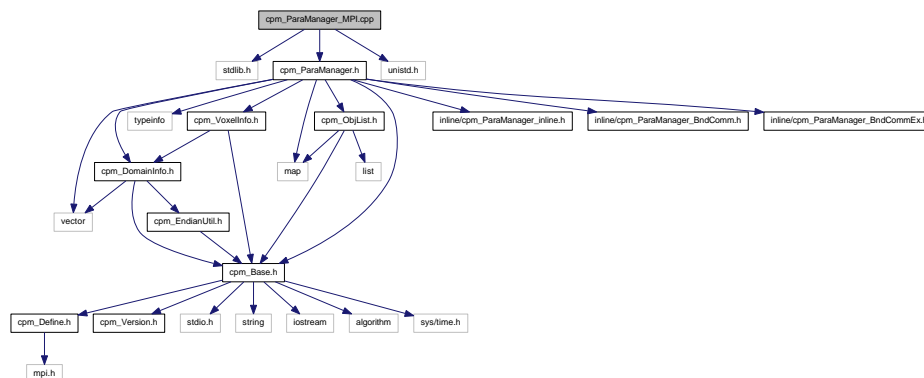
2012/05/31

[cpm_ParaManager_inline.h](#) で定義されています。

7.14 cpm_ParaManager_MPI.cpp

```
#include "stdlib.h"
#include "cpm_ParaManager.h"
#include <unistd.h>
```

cpm_ParaManager_MPI.cpp のインクルード依存関係図



7.14.1 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

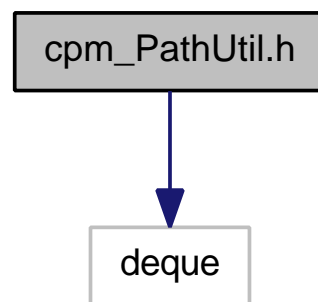
2012/05/31

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

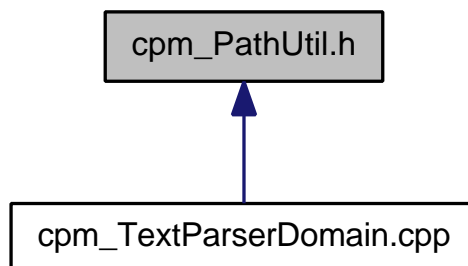
7.15 cpm_PathUtil.h

```
#include <deque>
```

cpm_PathUtil.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。



ネームスペース

- namespace [CES](#)
- namespace [CPM_PATH](#)

関数

- std::string [CES::DirName](#) (const std::string &path, const char dc= '/')
- std::string [CES::BaseName](#) (const std::string &path, const std::string &suffix=std::string(""), const char dc= '/')
- std::string [CES::OmitDots](#) (const std::string &path, const char dc= '/')
- char [CPM_PATH::cpmPath_getDelimChar](#) ()
- void [CPM_PATH::cpmPath_adjustDelim](#) (std::string &path)
- bool [CPM_PATH::cpmPath_hasDrive](#) (const std::string &path)

- std::string [CPM_PATH::cpmPath_emitDrive](#) (std::string &path)
- bool [CPM_PATH::cpmPath_isAbsolute](#) (const std::string &path)
- std::string [CPM_PATH::cpmPath_concat](#) (const std::string &path1, const std::string &path2)
- std::string [CPM_PATH::cpmPath_normalize](#) (const std::string &path)

7.15.1 説明

ファイルパス文字列関連ユーティリティヘッダーファイル

作者

University of Tokyo

日付

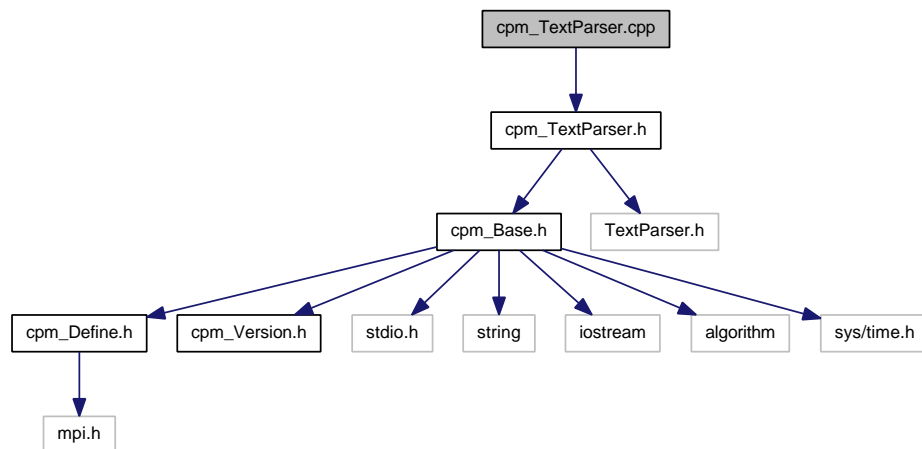
2013/04/02

[cpm_PathUtil.h](#) で定義されています。

7.16 cpm_TextParser.cpp

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

cpm_TextParser.cpp のインクルード依存関係図



7.16.1 説明

TextParser クラスのソースファイル

作者

University of Tokyo

日付

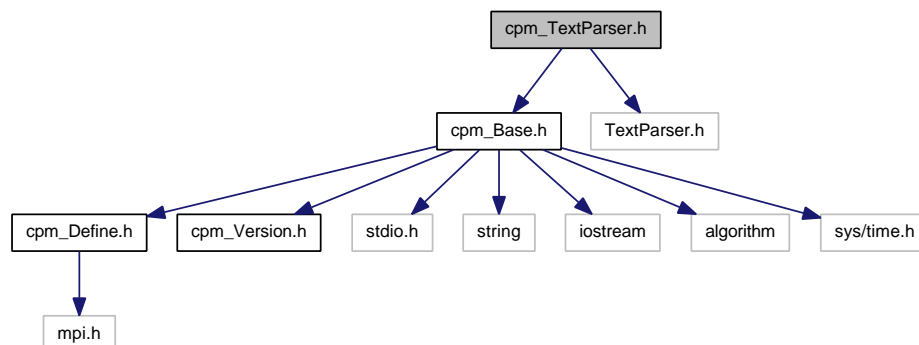
2012/05/31

[cpm_TextParser.cpp](#) で定義されています。

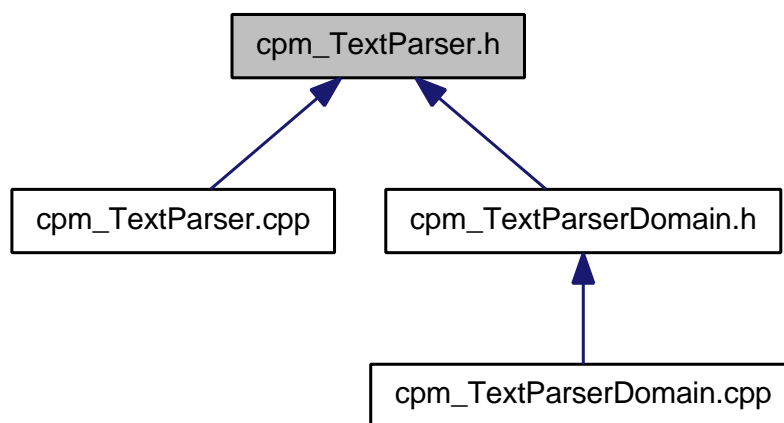
7.17 cpm_TextParser.h

```
#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](#)

7.17.1 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

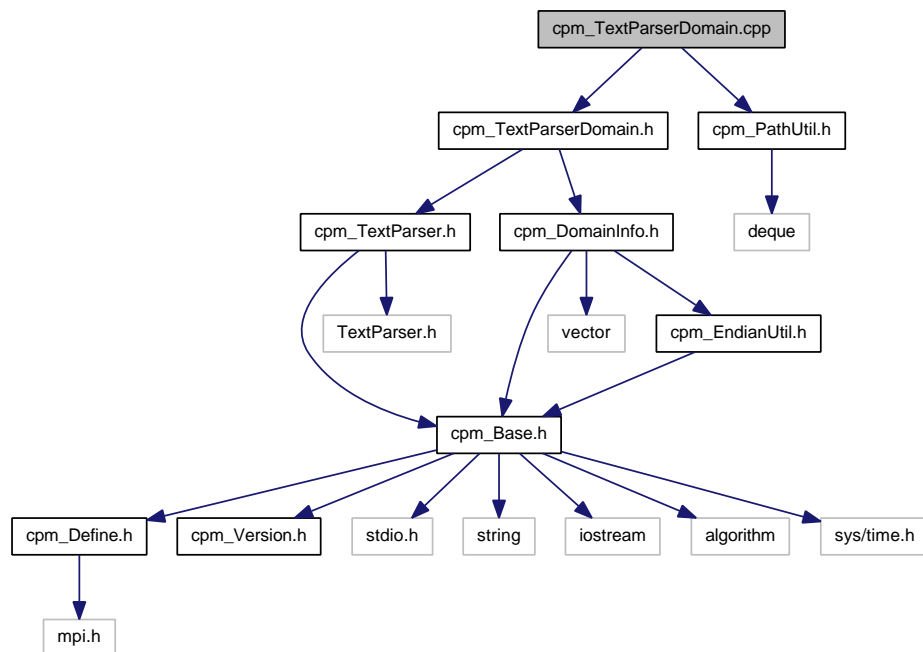
[cpm_TextParser.h](#) で定義されています。

7.18 cpm_TextParserDomain.cpp

```
#include "cpm_TextParserDomain.h"
```

```
#include "cpm_PathUtil.h"
```

cpm_TextParserDomain.cpp のインクルード依存関係図



7.18.1 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

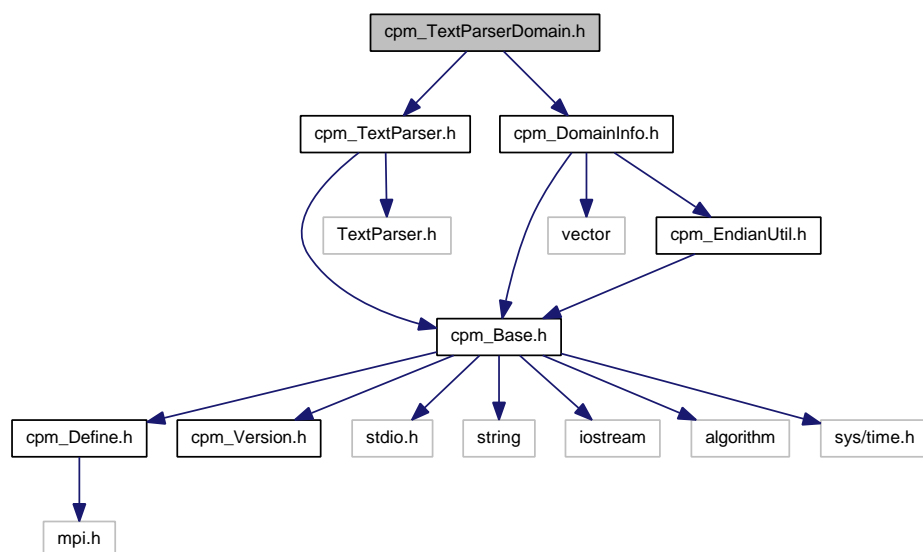
[cpm_TextParserDomain.cpp](#) で定義されています。

7.19 cpm_TextParserDomain.h

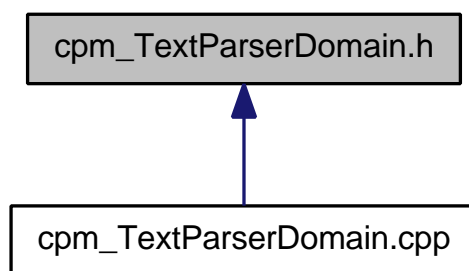
```
#include "cpm_TextParser.h"
```

```
#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](#)

7.19.1 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

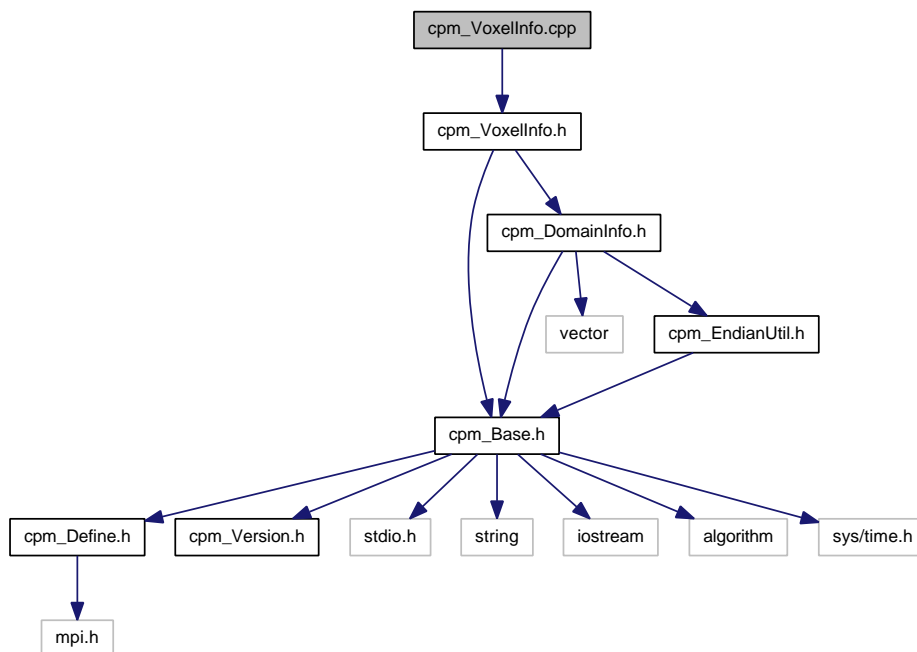
2012/05/31

[cpm_TextParserDomain.h](#) で定義されています。

7.21 cpm_VoxelInfo.cpp

```
#include "cpm_VoxelInfo.h"
```

cpm_VoxelInfo.cpp のインクルード依存関係図



7.21.1 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

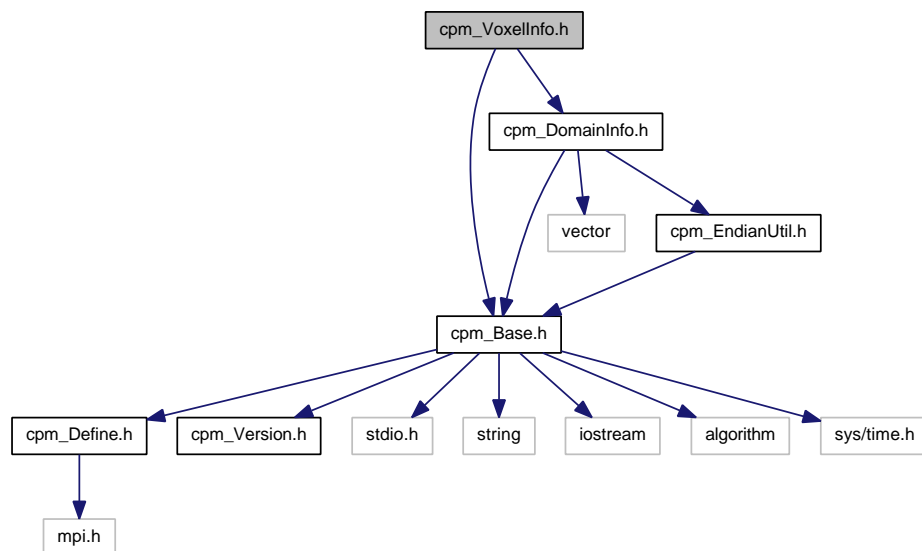
2012/05/31

[cpm_VoxelInfo.cpp](#) で定義されています。

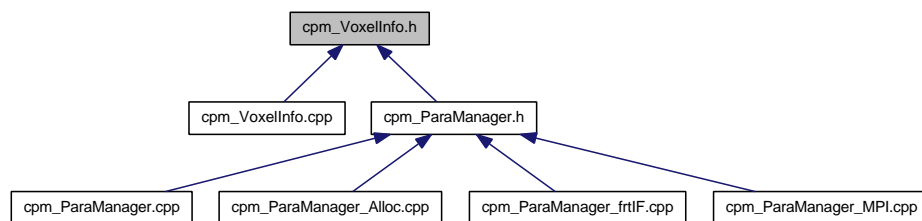
7.22 cpm_VoxelInfo.h

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

cpm_VoxelInfo.h のインクルード依存関係図



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



構成

- class [cpm_VoxelInfo](#)

7.22.1 説明

VOXEL 空間情報クラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

[cpm_VoxelInfo.h](#) で定義されています。

Index

- ~S_BNDCOMM_BUFFER
 - S_BNDCOMM_BUFFER, [127](#)
- ~cpm_ActiveSubdomainInfo
 - cpm_ActiveSubdomainInfo, [16](#)
- ~cpm_Base
 - cpm_Base, [19](#)
- ~cpm_DomainInfo
 - cpm_DomainInfo, [24](#)
- ~cpm_GlobalDomainInfo
 - cpm_GlobalDomainInfo, [29](#)
- ~cpm_LocalDomainInfo
 - cpm_LocalDomainInfo, [34](#)
- ~cpm_ObjList
 - cpm_ObjList, [36](#)
- ~cpm_ParaManager
 - cpm_ParaManager, [45](#)
- ~cpm_TextParser
 - cpm_TextParser, [113](#)
- ~cpm_TextParserDomain
 - cpm_TextParserDomain, [116](#)
- ~cpm_VoxelInfo
 - cpm_VoxelInfo, [120](#)
- _IDXFX
 - cpm_ParaManager_BndComm.h, [148](#)
 - cpm_ParaManager_BndCommEx.h, [150](#)
- _IDXFY
 - cpm_ParaManager_BndComm.h, [149](#)
 - cpm_ParaManager_BndCommEx.h, [150](#)
- _IDXFZ
 - cpm_ParaManager_BndComm.h, [149](#)
 - cpm_ParaManager_BndCommEx.h, [150](#)
- _IDX_S3D
 - cpm_Define.h, [134](#)
- _IDX_S4D
 - cpm_Define.h, [135](#)
- _IDX_S4DEX
 - cpm_Define.h, [135](#)
- _IDX_V3D
 - cpm_Define.h, [136](#)
- _IDX_V3DEX
 - cpm_Define.h, [136](#)
- Abort
 - cpm_ParaManager, [45](#)
- Add
 - cpm_ObjList, [36](#)
- AddSubdomain
 - cpm_GlobalDomainInfo, [29](#)
- Allgather
 - cpm_ParaManager, [45](#)
- Allgatherv
 - cpm_ParaManager, [46](#)
- AllocDoubleS3D
 - cpm_ParaManager, [47](#)
- AllocDoubleS4D
 - cpm_ParaManager, [47](#)
- AllocDoubleS4DEX
 - cpm_ParaManager, [47](#)
- AllocDoubleV3D
 - cpm_ParaManager, [48](#)
- AllocDoubleV3DEX
 - cpm_ParaManager, [48](#)
- AllocFloatS3D
 - cpm_ParaManager, [48](#)
- AllocFloatS4D
 - cpm_ParaManager, [49](#)
- AllocFloatS4DEX
 - cpm_ParaManager, [49](#)
- AllocFloatV3D
 - cpm_ParaManager, [49](#)
- AllocFloatV3DEX
 - cpm_ParaManager, [50](#)
- AllocIntS3D
 - cpm_ParaManager, [50](#)
- AllocIntS4D
 - cpm_ParaManager, [50](#)
- AllocIntS4DEX
 - cpm_ParaManager, [51](#)
- AllocIntV3D
 - cpm_ParaManager, [51](#)
- AllocIntV3DEX
 - cpm_ParaManager, [51](#)
- Allreduce
 - cpm_ParaManager, [51](#), [52](#)
- BOTH
 - cpm_Define.h, [140](#)
- BSWAP16
 - CPM_ENDIAN, [10](#)
- BSWAP32
 - CPM_ENDIAN, [10](#)
- BSWAP64
 - CPM_ENDIAN, [10](#)
- BSWAPVEC
 - CPM_ENDIAN, [11](#)
- Barrier
 - cpm_ParaManager, [52](#)
- BaseName
 - CES, [9](#)
- Bcast

- cpm_ParaManager, [53](#)
- BndCommInfoMap
 - cpm_ParaManager.h, [147](#)
- BndCommsS3D
 - cpm_ParaManager, [54](#)
- BndCommsS3D_nowait
 - cpm_ParaManager, [54](#), [55](#)
- BndCommsS4D
 - cpm_ParaManager, [56](#)
- BndCommsS4D_nowait
 - cpm_ParaManager, [57](#)
- BndCommsS4DEx
 - cpm_ParaManager, [58](#)
- BndCommsS4DEx_nowait
 - cpm_ParaManager, [59](#)
- BndCommV3D
 - cpm_ParaManager, [60](#)
- BndCommV3D_nowait
 - cpm_ParaManager, [61](#)
- BndCommV3DEx
 - cpm_ParaManager, [62](#)
- BndCommV3DEx_nowait
 - cpm_ParaManager, [63](#)
- CES, [9](#)
 - BaseName, [9](#)
 - DirName, [9](#)
 - OmitDots, [9](#)
- CPM_BAND
 - cpm_Define.h, [140](#)
- CPM_BOR
 - cpm_Define.h, [140](#)
- CPM_BXOR
 - cpm_Define.h, [140](#)
- CPM_BYTE
 - cpm_Define.h, [137](#)
- CPM_CHAR
 - cpm_Define.h, [137](#)
- CPM_DOUBLE
 - cpm_Define.h, [137](#)
- CPM_Datatype
 - cpm_Define.h, [137](#)
- CPM_ENDIAN, [9](#)
 - BSWAP16, [10](#)
 - BSWAP32, [10](#)
 - BSWAP64, [10](#)
 - BSWAPVEC, [11](#)
 - DBSWAPVEC, [11](#)
 - EMatchType, [10](#)
 - Match, [10](#)
 - SBSWAPVEC, [11](#)
 - UnKnown, [10](#)
 - UnMatch, [10](#)
- CPM_ERROR
 - cpm_Define.h, [138](#)
- CPM_ERROR_ALREADY_VOXELINIIT
 - cpm_Define.h, [138](#)
- CPM_ERROR_BNDCOMM
 - cpm_Define.h, [139](#)
- CPM_ERROR_BNDCOMM_BUFFER
 - cpm_Define.h, [139](#)
- CPM_ERROR_BNDCOMM_BUFFERLENGTH
 - cpm_Define.h, [139](#)
- CPM_ERROR_BNDCOMM_VOXELSIZE
 - cpm_Define.h, [139](#)
- CPM_ERROR_CREATE_LOCALDOMAIN
 - cpm_Define.h, [138](#)
- CPM_ERROR_CREATE_NEIGHBOR
 - cpm_Define.h, [138](#)
- CPM_ERROR_CREATE_PROCGROUP
 - cpm_Define.h, [138](#)
- CPM_ERROR_CREATE_RANKMAP
 - cpm_Define.h, [138](#)
- CPM_ERROR_GET_DIVNUM
 - cpm_Define.h, [138](#)
- CPM_ERROR_GET_DIVPOS
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_GLOBALORIGIN
 - cpm_Define.h, [138](#)
- CPM_ERROR_GET_GLOBALREGION
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_GLOBALVOXELSIZE
 - cpm_Define.h, [138](#)
- CPM_ERROR_GET_HEADINDEX
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_INFO
 - cpm_Define.h, [138](#)
- CPM_ERROR_GET_LOCALORIGIN
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_LOCALREGION
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_LOCALVOXELSIZE
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_MYRANK
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_NEIGHBOR_RANK
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_NUMRANK
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_PERIODIC_RANK
 - cpm_Define.h, [139](#)
- CPM_ERROR_GET_PITCH
 - cpm_Define.h, [138](#)
- CPM_ERROR_GET_TAILINDEX
 - cpm_Define.h, [139](#)
- CPM_ERROR_INSERT_VOXELMAP
 - cpm_Define.h, [138](#)
- CPM_ERROR_INVALID_DIVNUM
 - cpm_Define.h, [138](#)
- CPM_ERROR_INVALID_DOMAIN_NO
 - cpm_Define.h, [138](#)
- CPM_ERROR_INVALID_OBJKEY
 - cpm_Define.h, [138](#)
- CPM_ERROR_INVALID_PTR
 - cpm_Define.h, [138](#)
- CPM_ERROR_INVALID_REGION
 - cpm_Define.h, [138](#)

- CPM_ERROR_INVALID_VOXELSIZE
cpm_Define.h, [138](#)
- CPM_ERROR_MISMATCH_DIV_SUBDOMAIN
cpm_Define.h, [138](#)
- CPM_ERROR_MISMATCH_NP_SUBDOMAIN
cpm_Define.h, [138](#)
- CPM_ERROR_MPI
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_ALLGATHER
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_ALLGATHERV
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_ALLREDUCE
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_BARRIER
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_BCAST
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_DIMSCREATE
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_GATHER
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_GATHERV
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_INVALID_COMM
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_INVALID_DATATYPE
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_INVALID_OPERATOR
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_INVALID_REQUEST
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_IRECV
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_ISEND
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_RECV
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_SEND
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_WAIT
cpm_Define.h, [139](#)
- CPM_ERROR_MPI_WAITALL
cpm_Define.h, [139](#)
- CPM_ERROR_NO_MPI_INIT
cpm_Define.h, [139](#)
- CPM_ERROR_NO_TEXTPARSER
cpm_Define.h, [138](#)
- CPM_ERROR_NOT_IN_PROCGROUP
cpm_Define.h, [138](#)
- CPM_ERROR_OPEN_SBDM
cpm_Define.h, [138](#)
- CPM_ERROR_PERIODIC
cpm_Define.h, [139](#)
- CPM_ERROR_PERIODIC_INVALID_DIR
cpm_Define.h, [139](#)
- CPM_ERROR_PERIODIC_INVALID_PM
cpm_Define.h, [139](#)
- CPM_ERROR_PM_INSTANCE
cpm_Define.h, [138](#)
- CPM_ERROR_READ_SBDM_CONTENTS
cpm_Define.h, [138](#)
- CPM_ERROR_READ_SBDM_DIV
cpm_Define.h, [138](#)
- CPM_ERROR_READ_SBDM_FORMAT
cpm_Define.h, [138](#)
- CPM_ERROR_READ_SBDM_HEADER
cpm_Define.h, [138](#)
- CPM_ERROR_REGIST_OBJKEY
cpm_Define.h, [138](#)
- CPM_ERROR_SBDM_NUMDOMAIN_ZERO
cpm_Define.h, [138](#)
- CPM_ERROR_TEXTPARSER
cpm_Define.h, [138](#)
- CPM_ERROR_TP_INVALID_G_DIV
cpm_Define.h, [138](#)
- CPM_ERROR_TP_INVALID_G_ORG
cpm_Define.h, [138](#)
- CPM_ERROR_TP_INVALID_G_PITCH
cpm_Define.h, [138](#)
- CPM_ERROR_TP_INVALID_G_RGN
cpm_Define.h, [138](#)
- CPM_ERROR_TP_INVALID_G_VOXEL
cpm_Define.h, [138](#)
- CPM_ERROR_TP_INVALID_POS
cpm_Define.h, [138](#)
- CPM_ERROR_TP_NOVECTOR
cpm_Define.h, [138](#)
- CPM_ERROR_TP_VECTOR_SIZE
cpm_Define.h, [138](#)
- CPM_ERROR_VOXELINIT
cpm_Define.h, [138](#)
- CPM_EXTERN
cpm_ParaManager_frtIF.cpp, [155](#)
- CPM_FLOAT
cpm_Define.h, [137](#)
- CPM_INLINE
cpm_Base.h, [132](#)
- CPM_INT
cpm_Define.h, [137](#)
- CPM_LAND
cpm_Define.h, [140](#)
- CPM_LONG
cpm_Define.h, [137](#)
- CPM_LONG_DOUBLE
cpm_Define.h, [137](#)
- CPM_LOR
cpm_Define.h, [140](#)
- CPM_LXOR
cpm_Define.h, [140](#)
- CPM_MAX
cpm_Define.h, [140](#)
- CPM_MAXLOC
cpm_Define.h, [140](#)
- CPM_MIN
cpm_Define.h, [140](#)

- CPM_MINLOC
 - cpm_Define.h, 140
- CPM_Op
 - cpm_Define.h, 140
- CPM_PATH, 11
 - cpmPath_adjustDelim, 12
 - cpmPath_concat, 12
 - cpmPath_emitDrive, 12
 - cpmPath_getDelimChar, 12
 - cpmPath_hasDrive, 12
 - cpmPath_isAbsolute, 12
 - cpmPath_normalize, 12
- CPM_PROD
 - cpm_Define.h, 140
- CPM_REAL
 - cpm_Define.h, 137
- CPM_REVISION
 - cpm_Version.h, 186
- CPM_SHORT
 - cpm_Define.h, 137
- CPM_SUCCESS
 - cpm_Define.h, 138
- CPM_SUM
 - cpm_Define.h, 140
- CPM_UNSIGNED
 - cpm_Define.h, 137
- CPM_UNSIGNED_CHAR
 - cpm_Define.h, 137
- CPM_UNSIGNED_LONG
 - cpm_Define.h, 137
- CPM_UNSIGNED_SHORT
 - cpm_Define.h, 137
- CPM_VERSION_NO
 - cpm_Version.h, 186
- CalcCommSize
 - cpm_ParaManager, 64
- CheckData
 - cpm_DomainInfo, 24
 - cpm_GlobalDomainInfo, 29
- clear
 - cpm_ActiveSubdomainInfo, 16
 - cpm_DomainInfo, 25
 - cpm_GlobalDomainInfo, 30
 - cpm_LocalDomainInfo, 34
- CopyArray
 - cpm_ParaManager, 64
- cpm_Abort_
 - cpm_ParaManager_frtIF.cpp, 154, 159
- cpm_ActiveSubdomainInfo, 15
 - ~cpm_ActiveSubdomainInfo, 16
 - clear, 16
 - cpm_ActiveSubdomainInfo, 16
 - cpm_ActiveSubdomainInfo, 16
 - GetPos, 17
 - m_pos, 18
 - operator==, 17
 - SetPos, 17
- cpm_Allgather_
 - cpm_ParaManager_frtIF.cpp, 154, 159
- cpm_Allgatherv_
 - cpm_ParaManager_frtIF.cpp, 154, 159
- cpm_Allreduce_
 - cpm_ParaManager_frtIF.cpp, 154, 160
- cpm_Barrier_
 - cpm_ParaManager_frtIF.cpp, 154, 160
- cpm_Base, 18
 - ~cpm_Base, 19
 - cpm_Base, 19
 - cpm_strCompare, 19
 - cpm_strCompareN, 20
 - cpm_Base, 19
 - getCommNull, 20
 - GetMemString, 20
 - getRankNull, 20
 - GetSpanTime, 21
 - GetTime, 21
 - GetWSpanTime, 21
 - GetWTime, 21
 - IsCommNull, 22
 - IsRankNull, 22
 - ReallsDouble, 22
 - VersionInfo, 22, 23
- cpm_Base.h, 131
 - CPM_INLINE, 132
- cpm_Bcast_
 - cpm_ParaManager_frtIF.cpp, 154, 160
- cpm_BndCommS3D_
 - cpm_ParaManager_frtIF.cpp, 154, 161
- cpm_BndCommS3D_nowait
 - cpm_ParaManager, 65
- cpm_BndCommS3D_nowait_
 - cpm_ParaManager_frtIF.cpp, 154, 161
- cpm_BndCommS4D_
 - cpm_ParaManager_frtIF.cpp, 154, 161
- cpm_BndCommS4D_nowait
 - cpm_ParaManager, 65
- cpm_BndCommS4D_nowait_
 - cpm_ParaManager_frtIF.cpp, 155, 162
- cpm_BndCommS4DEx_
 - cpm_ParaManager_frtIF.cpp, 155, 162
- cpm_BndCommS4DEx_nowait
 - cpm_ParaManager, 66
- cpm_BndCommS4DEx_nowait_
 - cpm_ParaManager_frtIF.cpp, 155, 163
- cpm_BndCommV3D_
 - cpm_ParaManager_frtIF.cpp, 155, 163
- cpm_BndCommV3D_nowait
 - cpm_ParaManager, 66
- cpm_BndCommV3D_nowait_
 - cpm_ParaManager_frtIF.cpp, 155, 164
- cpm_BndCommV3DEx_
 - cpm_ParaManager_frtIF.cpp, 155, 164
- cpm_BndCommV3DEx_nowait
 - cpm_ParaManager, 67
- cpm_BndCommV3DEx_nowait_
 - cpm_ParaManager_frtIF.cpp, 155, 165

- cpm_Define.h, 132
- _IDX_S3D, 134
- _IDX_S4D, 135
- _IDX_S4DEX, 135
- _IDX_V3D, 136
- _IDX_V3DEX, 136
- BOTH, 140
- CPM_BAND, 140
- CPM_BOR, 140
- CPM_BXOR, 140
- CPM_BYTE, 137
- CPM_CHAR, 137
- CPM_DOUBLE, 137
- CPM_Datatype, 137
- CPM_ERROR, 138
- CPM_ERROR_ALREADY_VOXELINIT, 138
- CPM_ERROR_BNDCOMM, 139
- CPM_ERROR_BNDCOMM_BUFFER, 139
- CPM_ERROR_BNDCOMM_BUFFERLENGTH, 139
- CPM_ERROR_BNDCOMM_VOXELSIZE, 139
- CPM_ERROR_CREATE_LOCALDOMAIN, 138
- CPM_ERROR_CREATE_NEIGHBOR, 138
- CPM_ERROR_CREATE_PROCGROUP, 138
- CPM_ERROR_CREATE_RANKMAP, 138
- CPM_ERROR_GET_DIVNUM, 138
- CPM_ERROR_GET_DIVPOS, 139
- CPM_ERROR_GET_GLOBALORIGIN, 138
- CPM_ERROR_GET_GLOBALREGION, 139
- CPM_ERROR_GET_GLOBALVOXELSIZE, 138
- CPM_ERROR_GET_HEADINDEX, 139
- CPM_ERROR_GET_INFO, 138
- CPM_ERROR_GET_LOCALORIGIN, 139
- CPM_ERROR_GET_LOCALREGION, 139
- CPM_ERROR_GET_LOCALVOXELSIZE, 139
- CPM_ERROR_GET_MYRANK, 139
- CPM_ERROR_GET_NEIGHBOR_RANK, 139
- CPM_ERROR_GET_NUMRANK, 139
- CPM_ERROR_GET_PERIODIC_RANK, 139
- CPM_ERROR_GET_PITCH, 138
- CPM_ERROR_GET_TAILINDEX, 139
- CPM_ERROR_INSERT_VOXELMAP, 138
- CPM_ERROR_INVALID_DIVNUM, 138
- CPM_ERROR_INVALID_DOMAIN_NO, 138
- CPM_ERROR_INVALID_OBJKEY, 138
- CPM_ERROR_INVALID_PTR, 138
- CPM_ERROR_INVALID_REGION, 138
- CPM_ERROR_INVALID_VOXELSIZE, 138
- CPM_ERROR_MISMATCH_DIV_SUBDOMAIN, 138
- CPM_ERROR_MISMATCH_NP_SUBDOMAIN, 138
- CPM_ERROR_MPI, 139
- CPM_ERROR_MPI_ALLGATHER, 139
- CPM_ERROR_MPI_ALLGATHERV, 139
- CPM_ERROR_MPI_ALLREDUCE, 139
- CPM_ERROR_MPI_BARRIER, 139
- CPM_ERROR_MPI_BCAST, 139
- CPM_ERROR_MPI_DIMSCREATE, 139
- CPM_ERROR_MPI_GATHER, 139
- CPM_ERROR_MPI_GATHERV, 139
- CPM_ERROR_MPI_INVALID_COMM, 139
- CPM_ERROR_MPI_INVALID_DATATYPE, 139
- CPM_ERROR_MPI_INVALID_OPERATOR, 139
- CPM_ERROR_MPI_INVALID_REQUEST, 139
- CPM_ERROR_MPI_Irecv, 139
- CPM_ERROR_MPI_Isend, 139
- CPM_ERROR_MPI_RECV, 139
- CPM_ERROR_MPI_SEND, 139
- CPM_ERROR_MPI_WAIT, 139
- CPM_ERROR_MPI_WAITALL, 139
- CPM_ERROR_NO_MPI_INIT, 139
- CPM_ERROR_NO_TEXTPARSER, 138
- CPM_ERROR_NOT_IN_PROCGROUP, 138
- CPM_ERROR_OPEN_SBDM, 138
- CPM_ERROR_PERIODIC, 139
- CPM_ERROR_PERIODIC_INVALID_DIR, 139
- CPM_ERROR_PERIODIC_INVALID_PM, 139
- CPM_ERROR_PM_INSTANCE, 138
- CPM_ERROR_READ_SBDM_CONTENTS, 138
- CPM_ERROR_READ_SBDM_DIV, 138
- CPM_ERROR_READ_SBDM_FORMAT, 138
- CPM_ERROR_READ_SBDM_HEADER, 138
- CPM_ERROR_REGIST_OBJKEY, 138
- CPM_ERROR_SBDM_NUMDOMAIN_ZERO, 138
- CPM_ERROR_TEXTPARSER, 138
- CPM_ERROR_TP_INVALID_G_DIV, 138
- CPM_ERROR_TP_INVALID_G_ORG, 138
- CPM_ERROR_TP_INVALID_G_PITCH, 138
- CPM_ERROR_TP_INVALID_G_RGN, 138
- CPM_ERROR_TP_INVALID_G_VOXEL, 138
- CPM_ERROR_TP_INVALID_POS, 138
- CPM_ERROR_TP_NOVECTOR, 138
- CPM_ERROR_TP_VECTOR_SIZE, 138
- CPM_ERROR_VOXELINIT, 138
- CPM_FLOAT, 137
- CPM_INT, 137
- CPM_LAND, 140
- CPM_LONG, 137
- CPM_LONG_DOUBLE, 137
- CPM_LOR, 140
- CPM_LXOR, 140
- CPM_MAX, 140
- CPM_MAXLOC, 140
- CPM_MIN, 140
- CPM_MINLOC, 140
- CPM_Op, 140
- CPM_PROD, 140
- CPM_REAL, 137
- CPM_SHORT, 137
- CPM_SUCCESS, 138
- CPM_SUM, 140
- CPM_UNSIGNED, 137
- CPM_UNSIGNED_CHAR, 137
- CPM_UNSIGNED_LONG, 137
- CPM_UNSIGNED_SHORT, 137

- cpm_DirFlag, 137
- cpm_ErrorCode, 137
- cpm_FaceFlag, 139
- cpm_PMFlag, 140
- MINUS2PLUS, 140
- PLUS2MINUS, 140
- REAL_BUF_TYPE, 136
- X_DIR, 137
- X_MINUS, 140
- X_PLUS, 140
- Y_DIR, 137
- Y_MINUS, 140
- Y_PLUS, 140
- Z_DIR, 137
- Z_MINUS, 140
- Z_PLUS, 140
- cpm_DirFlag
 - cpm_Define.h, 137
- cpm_DomainInfo, 23
 - ~cpm_DomainInfo, 24
 - CheckData, 24
 - clear, 25
 - cpm_DomainInfo, 24
 - cpm_DomainInfo, 24
 - GetOrigin, 25
 - GetPitch, 25
 - GetRegion, 25
 - GetVoxNum, 25
 - m_origin, 27
 - m_pitch, 27
 - m_region, 27
 - m_voxNum, 27
 - SetOrigin, 26
 - SetPitch, 26
 - SetRegion, 26
 - SetVoxNum, 26
- cpm_DomainInfo.cpp, 141
- cpm_DomainInfo.h, 141
- cpm_EndianUtil.h, 143
- cpm_ErrorCode
 - cpm_Define.h, 137
- cpm_FaceFlag
 - cpm_Define.h, 139
- cpm_Gather_
 - cpm_ParaManager_frtlF.cpp, 155, 165
- cpm_Gatherv_
 - cpm_ParaManager_frtlF.cpp, 155, 166
- cpm_GetDivNum_
 - cpm_ParaManager_frtlF.cpp, 155, 166
- cpm_GetDivPos_
 - cpm_ParaManager_frtlF.cpp, 156, 166
- cpm_GetGlobalOrigin_
 - cpm_ParaManager_frtlF.cpp, 156, 167
- cpm_GetGlobalRegion_
 - cpm_ParaManager_frtlF.cpp, 156, 167
- cpm_GetGlobalVoxelSize_
 - cpm_ParaManager_frtlF.cpp, 156, 167
- cpm_GetLocalOrigin_
 - cpm_ParaManager_frtlF.cpp, 156, 168
- cpm_GetLocalRegion_
 - cpm_ParaManager_frtlF.cpp, 156, 168
- cpm_GetLocalVoxelSize_
 - cpm_ParaManager_frtlF.cpp, 156, 168
- cpm_GetMyRankID_
 - cpm_ParaManager_frtlF.cpp, 156, 169
- cpm_GetNeighborRankID_
 - cpm_ParaManager_frtlF.cpp, 156, 169
- cpm_GetNumRank_
 - cpm_ParaManager_frtlF.cpp, 156, 169
- cpm_GetPeriodicRankID_
 - cpm_ParaManager_frtlF.cpp, 156, 169
- cpm_GetPitch_
 - cpm_ParaManager_frtlF.cpp, 156, 170
- cpm_GetVoxelHeadIndex_
 - cpm_ParaManager_frtlF.cpp, 157, 170
- cpm_GetVoxelTailIndex_
 - cpm_ParaManager_frtlF.cpp, 157, 170
- cpm_GlobalDomainInfo, 27
 - ~cpm_GlobalDomainInfo, 29
 - AddSubdomain, 29
 - CheckData, 29
 - clear, 30
 - cpm_GlobalDomainInfo, 29
 - cpm_GlobalDomainInfo, 29
 - GetDivNum, 30
 - GetSubdomainArraySize, 30
 - GetSubdomainInfo, 30
 - GetSubdomainNum, 31
 - IsExistSubdomain, 31
 - isMatchEndianSbdmMagick, 31
 - m_divNum, 33
 - m_subDomainInfo, 33
 - ReadActiveSubdomainFile, 32
 - SetDivNum, 32
- cpm_Initialize_
 - cpm_ParaManager_frtlF.cpp, 157, 171
- cpm_Irecv
 - cpm_ParaManager, 67
- cpm_Irecv_
 - cpm_ParaManager_frtlF.cpp, 157, 171
- cpm_IsParallel_
 - cpm_ParaManager_frtlF.cpp, 157, 172
- cpm_Isend
 - cpm_ParaManager, 68
- cpm_Isend_
 - cpm_ParaManager_frtlF.cpp, 157, 171
- cpm_LocalDomainInfo, 33
 - ~cpm_LocalDomainInfo, 34
 - clear, 34
 - cpm_LocalDomainInfo, 34
 - cpm_LocalDomainInfo, 34
- cpm_ObjList
 - ~cpm_ObjList, 36
 - Add, 36
 - cpm_ObjList, 36
 - cpm_ObjList, 36

- Create, [36](#)
- DelKeyList, [36](#)
- Delete, [37](#)
- Get, [37](#)
- m_DelKeyList, [37](#)
- m_ObjectMap, [38](#)
- m_newKey, [38](#)
- ObjectMap, [36](#)
- cpm_ObjList< T >, [35](#)
- cpm_ObjList.h, [144](#)
- RankNoMap, [145](#)
- cpm_PMFlag
 - cpm_Define.h, [140](#)
- cpm_ParaManager, [38](#)
 - ~cpm_ParaManager, [45](#)
 - Abort, [45](#)
 - Allgather, [45](#)
 - Allgatherv, [46](#)
 - AllocDoubleS3D, [47](#)
 - AllocDoubleS4D, [47](#)
 - AllocDoubleS4DEx, [47](#)
 - AllocDoubleV3D, [48](#)
 - AllocDoubleV3DEx, [48](#)
 - AllocFloatS3D, [48](#)
 - AllocFloatS4D, [49](#)
 - AllocFloatS4DEx, [49](#)
 - AllocFloatV3D, [49](#)
 - AllocFloatV3DEx, [50](#)
 - AllocIntS3D, [50](#)
 - AllocIntS4D, [50](#)
 - AllocIntS4DEx, [51](#)
 - AllocIntV3D, [51](#)
 - AllocIntV3DEx, [51](#)
 - Allreduce, [51](#), [52](#)
 - Barrier, [52](#)
 - Bcast, [53](#)
 - BndCommS3D, [54](#)
 - BndCommS3D_nowait, [54](#), [55](#)
 - BndCommS4D, [56](#)
 - BndCommS4D_nowait, [57](#)
 - BndCommS4DEx, [58](#)
 - BndCommS4DEx_nowait, [59](#)
 - BndCommV3D, [60](#)
 - BndCommV3D_nowait, [61](#)
 - BndCommV3DEx, [62](#)
 - BndCommV3DEx_nowait, [63](#)
 - CalcCommSize, [64](#)
 - CopyArray, [64](#)
 - cpm_BndCommS3D_nowait, [65](#)
 - cpm_BndCommS4D_nowait, [65](#)
 - cpm_BndCommS4DEx_nowait, [66](#)
 - cpm_BndCommV3D_nowait, [66](#)
 - cpm_BndCommV3DEx_nowait, [67](#)
 - cpm_Irecv, [67](#)
 - cpm_Isend, [68](#)
 - cpm_ParaManager, [45](#)
 - cpm_Wait, [68](#)
 - cpm_Waitall, [71](#)
 - cpm_wait_BndCommS3D, [69](#)
 - cpm_wait_BndCommS4D, [69](#)
 - cpm_wait_BndCommS4DEx, [70](#)
 - cpm_wait_BndCommV3D, [70](#)
 - cpm_wait_BndCommV3DEx, [71](#)
 - cpm_ParaManager, [45](#)
 - cpm_VoxelInfo, [125](#)
 - CreateProcessGroup, [72](#)
 - DecideDivPattern, [72](#)
 - FindVoxelInfo, [73](#)
 - flush, [73](#)
 - Gather, [73](#), [74](#)
 - Gatherv, [74](#)
 - get_instance, [75](#)
 - GetBndCommBuffer, [76](#)
 - GetBndCommBufferSize, [76](#)
 - GetBndIndexExtGc, [76](#), [77](#)
 - GetDivNum, [77](#)
 - GetDivPos, [78](#)
 - GetGlobalOrigin, [78](#)
 - GetGlobalRegion, [78](#)
 - GetGlobalVoxelSize, [79](#)
 - GetHostName, [79](#)
 - GetLocalOrigin, [79](#)
 - GetLocalRegion, [80](#)
 - GetLocalVoxelSize, [80](#)
 - GetMPI_Comm, [80](#)
 - GetMPI_Datatype, [81](#)
 - GetMPI_Op, [81](#)
 - GetMyRankID, [82](#)
 - GetNeighborRankID, [82](#)
 - GetNumRank, [82](#)
 - GetPeriodicRankID, [83](#)
 - GetPitch, [83](#)
 - GetVoxelHeadIndex, [83](#)
 - GetVoxelTailIndex, [84](#)
 - InitArray, [84](#)
 - Initialize, [84](#), [85](#)
 - Irecv, [85](#)
 - IsInnerBoundary, [87](#)
 - IsOuterBoundary, [87](#)
 - IsParallel, [87](#), [88](#)
 - Isend, [86](#)
 - m_bndCommInfoMap, [110](#)
 - m_nRank, [110](#)
 - m_procGrpList, [110](#)
 - m_rankNo, [111](#)
 - m_rankNoMap, [111](#)
 - m_reqList, [111](#)
 - m_voxelInfoMap, [111](#)
 - packX, [88](#)
 - packXEx, [88](#)
 - packY, [89](#)
 - packYEx, [89](#)
 - packZ, [90](#)
 - packZEx, [90](#)
 - PeriodicCommS3D, [91](#)
 - PeriodicCommS4D, [92](#)

- PeriodicCommS4DEx, [93](#), [94](#)
- PeriodicCommV3D, [94](#), [95](#)
- PeriodicCommV3DEx, [95](#), [96](#)
- Recv, [96](#), [97](#)
- Send, [97](#)
- sendrecv, [98](#)
- SetBndCommBuffer, [98](#)
- unpackX, [99](#)
- unpackXEx, [99](#)
- unpackY, [100](#)
- unpackYEx, [100](#)
- unpackZ, [101](#)
- unpackZEx, [101](#)
- Voxellnit, [102](#), [103](#)
- Voxellnit_Subdomain, [103](#), [104](#)
- Wait, [104](#)
- wait_BndComms3D, [105](#)
- wait_BndComms4D, [106](#)
- wait_BndComms4DEx, [107](#)
- wait_BndCommV3D, [108](#)
- wait_BndCommV3DEx, [109](#)
- Waitall, [110](#)
- cpm_ParaManager.cpp, [145](#)
- cpm_ParaManager.h, [146](#)
 - BndCommInfoMap, [147](#)
 - RankNoMap, [147](#)
 - VoxellInfoMap, [147](#)
- cpm_ParaManager_Alloc.cpp, [147](#)
- cpm_ParaManager_BndComm.h, [148](#)
 - _IDXFX, [148](#)
 - _IDXFY, [149](#)
 - _IDXFZ, [149](#)
- cpm_ParaManager_BndCommEx.h, [149](#)
 - _IDXFX, [150](#)
 - _IDXFY, [150](#)
 - _IDXFZ, [150](#)
- cpm_ParaManager_MPI.cpp, [180](#)
- cpm_ParaManager_frtIF.cpp, [151](#)
 - CPM_EXTERN, [155](#)
 - cpm_Abort_, [154](#), [159](#)
 - cpm_Allgather_, [154](#), [159](#)
 - cpm_Allgatherv_, [154](#), [159](#)
 - cpm_Allreduce_, [154](#), [160](#)
 - cpm_Barrier_, [154](#), [160](#)
 - cpm_Bcast_, [154](#), [160](#)
 - cpm_BndComms3D_, [154](#), [161](#)
 - cpm_BndComms3D_nowait_, [154](#), [161](#)
 - cpm_BndComms4D_, [154](#), [161](#)
 - cpm_BndComms4D_nowait_, [155](#), [162](#)
 - cpm_BndComms4DEx_, [155](#), [162](#)
 - cpm_BndComms4DEx_nowait_, [155](#), [163](#)
 - cpm_BndCommV3D_, [155](#), [163](#)
 - cpm_BndCommV3D_nowait_, [155](#), [164](#)
 - cpm_BndCommV3DEx_, [155](#), [164](#)
 - cpm_BndCommV3DEx_nowait_, [155](#), [165](#)
 - cpm_Gather_, [155](#), [165](#)
 - cpm_Gatherv_, [155](#), [166](#)
 - cpm_GetDivNum_, [155](#), [166](#)
 - cpm_GetDivPos_, [156](#), [166](#)
 - cpm_GetGlobalOrigin_, [156](#), [167](#)
 - cpm_GetGlobalRegion_, [156](#), [167](#)
 - cpm_GetGlobalVoxelSize_, [156](#), [167](#)
 - cpm_GetLocalOrigin_, [156](#), [168](#)
 - cpm_GetLocalRegion_, [156](#), [168](#)
 - cpm_GetLocalVoxelSize_, [156](#), [168](#)
 - cpm_GetMyRankID_, [156](#), [169](#)
 - cpm_GetNeighborRankID_, [156](#), [169](#)
 - cpm_GetNumRank_, [156](#), [169](#)
 - cpm_GetPeriodicRankID_, [156](#), [169](#)
 - cpm_GetPitch_, [156](#), [170](#)
 - cpm_GetVoxelHeadIndex_, [157](#), [170](#)
 - cpm_GetVoxelTailIndex_, [157](#), [170](#)
 - cpm_Initialize_, [157](#), [171](#)
 - cpm_Irecv_, [157](#), [171](#)
 - cpm_IsParallel_, [157](#), [172](#)
 - cpm_Isend_, [157](#), [171](#)
 - cpm_PeriodicComms3D, [157](#)
 - cpm_PeriodicComms3D_, [172](#)
 - cpm_PeriodicComms4D, [157](#)
 - cpm_PeriodicComms4D_, [172](#)
 - cpm_PeriodicComms4DEx, [157](#)
 - cpm_PeriodicComms4DEx_, [173](#)
 - cpm_PeriodicCommV3D, [157](#)
 - cpm_PeriodicCommV3D_, [174](#)
 - cpm_PeriodicCommV3DEx, [157](#)
 - cpm_PeriodicCommV3DEx_, [174](#)
 - cpm_Recv_, [157](#), [175](#)
 - cpm_Send_, [158](#), [175](#)
 - cpm_SetBndCommBuffer_, [158](#), [175](#)
 - cpm_Voxellnit_, [158](#), [176](#)
 - cpm_Voxellnit_nodiv_, [158](#), [176](#)
 - cpm_Wait_, [158](#), [176](#)
 - cpm_Waitall_, [158](#), [179](#)
 - cpm_wait_BndComms3D_, [158](#), [177](#)
 - cpm_wait_BndComms4D_, [158](#), [177](#)
 - cpm_wait_BndComms4DEx_, [158](#), [178](#)
 - cpm_wait_BndCommV3D_, [158](#), [178](#)
 - cpm_wait_BndCommV3DEx_, [158](#), [179](#)
 - cpm_ParaManager_inline.h, [180](#)
 - cpm_PathUtil.h, [181](#)
 - cpm_PeriodicComms3D
 - cpm_ParaManager_frtIF.cpp, [157](#)
 - cpm_PeriodicComms3D_
 - cpm_ParaManager_frtIF.cpp, [172](#)
 - cpm_PeriodicComms4D
 - cpm_ParaManager_frtIF.cpp, [157](#)
 - cpm_PeriodicComms4D_
 - cpm_ParaManager_frtIF.cpp, [172](#)
 - cpm_PeriodicComms4DEx
 - cpm_ParaManager_frtIF.cpp, [157](#)
 - cpm_PeriodicComms4DEx_
 - cpm_ParaManager_frtIF.cpp, [173](#)
 - cpm_PeriodicCommV3D
 - cpm_ParaManager_frtIF.cpp, [157](#)
 - cpm_PeriodicCommV3D_
 - cpm_ParaManager_frtIF.cpp, [174](#)

- cpm_PeriodicCommV3DEx
 - cpm_ParaManager_frtlF.cpp, 157
- cpm_PeriodicCommV3DEx_
 - cpm_ParaManager_frtlF.cpp, 174
- cpm_Recv_
 - cpm_ParaManager_frtlF.cpp, 157, 175
- cpm_Send_
 - cpm_ParaManager_frtlF.cpp, 158, 175
- cpm_SetBndCommBuffer_
 - cpm_ParaManager_frtlF.cpp, 158, 175
- cpm_TextParser, 112
 - ~cpm_TextParser, 113
 - cpm_TextParser, 113
 - cpm_TextParser, 113
 - m_tp, 115
 - Read, 113
 - readVector, 113, 114
- cpm_TextParser.cpp, 182
- cpm_TextParser.h, 183
- cpm_TextParserDomain, 115
 - ~cpm_TextParserDomain, 116
 - cpm_TextParserDomain, 116
 - cpm_TextParserDomain, 116
 - Read, 116
 - ReadDomainInfo, 117
 - ReadMain, 117
 - ReadSubdomainInfo, 117
- cpm_TextParserDomain.cpp, 184
- cpm_TextParserDomain.h, 184
- cpm_Version.h, 186
 - CPM_REVISION, 186
 - CPM_VERSION_NO, 186
- cpm_VoxelInfo, 118
 - ~cpm_VoxelInfo, 120
 - cpm_ParaManager, 125
 - cpm_VoxelInfo, 120
 - cpm_VoxelInfo, 120
 - CreateLocalDomainInfo, 120
 - CreateNeighborRankInfo, 120
 - CreateRankMap, 121
 - GetDivNum, 121
 - GetDivPos, 121
 - GetGlobalOrigin, 121
 - GetGlobalRegion, 121
 - GetGlobalVoxelSize, 122
 - GetLocalOrigin, 122
 - GetLocalRegion, 122
 - GetLocalVoxelSize, 122
 - GetNeighborRankID, 123
 - GetPeriodicRankID, 123
 - GetPitch, 123
 - GetVoxelHeadIndex, 123
 - GetVoxelTailIndex, 124
 - Init, 124
 - IsInnerBoundary, 124
 - IsOuterBoundary, 125
 - m_comm, 125
 - m_globalDomainInfo, 125
 - m_localDomainInfo, 125
 - m_nRank, 126
 - m_neighborRankID, 126
 - m_periodicRankID, 126
 - m_rankMap, 126
 - m_rankNo, 126
 - m_voxelHeadIndex, 126
 - m_voxelTailIndex, 126
- cpm_VoxelInfo.cpp, 187
- cpm_VoxelInfo.h, 188
- cpm_VoxelInit_
 - cpm_ParaManager_frtlF.cpp, 158, 176
- cpm_VoxelInit_nodiv_
 - cpm_ParaManager_frtlF.cpp, 158, 176
- cpm_Wait
 - cpm_ParaManager, 68
- cpm_Wait_
 - cpm_ParaManager_frtlF.cpp, 158, 176
- cpm_Waitall
 - cpm_ParaManager, 71
- cpm_Waitall_
 - cpm_ParaManager_frtlF.cpp, 158, 179
- cpm_strCompare
 - cpm_Base, 19
- cpm_strCompareN
 - cpm_Base, 20
- cpm_wait_BndCommS3D
 - cpm_ParaManager, 69
- cpm_wait_BndCommS3D_
 - cpm_ParaManager_frtlF.cpp, 158, 177
- cpm_wait_BndCommS4D
 - cpm_ParaManager, 69
- cpm_wait_BndCommS4D_
 - cpm_ParaManager_frtlF.cpp, 158, 177
- cpm_wait_BndCommS4DEx
 - cpm_ParaManager, 70
- cpm_wait_BndCommS4DEx_
 - cpm_ParaManager_frtlF.cpp, 158, 178
- cpm_wait_BndCommV3D
 - cpm_ParaManager, 70
- cpm_wait_BndCommV3D_
 - cpm_ParaManager_frtlF.cpp, 158, 178
- cpm_wait_BndCommV3DEx
 - cpm_ParaManager, 71
- cpm_wait_BndCommV3DEx_
 - cpm_ParaManager_frtlF.cpp, 158, 179
- cpmPath_adjustDelim
 - CPM_PATH, 12
- cpmPath_concat
 - CPM_PATH, 12
- cpmPath_emitDrive
 - CPM_PATH, 12
- cpmPath_getDelimChar
 - CPM_PATH, 12
- cpmPath_hasDrive
 - CPM_PATH, 12
- cpmPath_isAbsolute
 - CPM_PATH, 12

- cpmPath_normalize
 - CPM_PATH, [12](#)
- Create
 - cpm_ObjList, [36](#)
- CreateLocalDomainInfo
 - cpm_VoxelInfo, [120](#)
- CreateNeighborRankInfo
 - cpm_VoxelInfo, [120](#)
- CreateProcessGroup
 - cpm_ParaManager, [72](#)
- CreateRankMap
 - cpm_VoxelInfo, [121](#)
- DBSWAPVEC
 - CPM_ENDIAN, [11](#)
- DecideDivPattern
 - cpm_ParaManager, [72](#)
- DelKeyList
 - cpm_ObjList, [36](#)
- Delete
 - cpm_ObjList, [37](#)
- DirName
 - CES, [9](#)
- EMatchType
 - CPM_ENDIAN, [10](#)
- FindVoxelInfo
 - cpm_ParaManager, [73](#)
- flush
 - cpm_ParaManager, [73](#)
- Gather
 - cpm_ParaManager, [73](#), [74](#)
- Gatherv
 - cpm_ParaManager, [74](#)
- Get
 - cpm_ObjList, [37](#)
- get_instance
 - cpm_ParaManager, [75](#)
- GetBndCommBuffer
 - cpm_ParaManager, [76](#)
- GetBndCommBufferSize
 - cpm_ParaManager, [76](#)
- GetBndIndexExtGc
 - cpm_ParaManager, [76](#), [77](#)
- getCommNull
 - cpm_Base, [20](#)
- GetDivNum
 - cpm_GlobalDomainInfo, [30](#)
 - cpm_ParaManager, [77](#)
 - cpm_VoxelInfo, [121](#)
- GetDivPos
 - cpm_ParaManager, [78](#)
 - cpm_VoxelInfo, [121](#)
- GetGlobalOrigin
 - cpm_ParaManager, [78](#)
 - cpm_VoxelInfo, [121](#)
- GetGlobalRegion
 - cpm_ParaManager, [78](#)
 - cpm_VoxelInfo, [121](#)
- GetGlobalVoxelSize
 - cpm_ParaManager, [79](#)
 - cpm_VoxelInfo, [122](#)
- GetHostName
 - cpm_ParaManager, [79](#)
- GetLocalOrigin
 - cpm_ParaManager, [79](#)
 - cpm_VoxelInfo, [122](#)
- GetLocalRegion
 - cpm_ParaManager, [80](#)
 - cpm_VoxelInfo, [122](#)
- GetLocalVoxelSize
 - cpm_ParaManager, [80](#)
 - cpm_VoxelInfo, [122](#)
- GetMPI_Comm
 - cpm_ParaManager, [80](#)
- GetMPI_Datatype
 - cpm_ParaManager, [81](#)
- GetMPI_Op
 - cpm_ParaManager, [81](#)
- GetMemString
 - cpm_Base, [20](#)
- GetMyRankID
 - cpm_ParaManager, [82](#)
- GetNeighborRankID
 - cpm_ParaManager, [82](#)
 - cpm_VoxelInfo, [123](#)
- GetNumRank
 - cpm_ParaManager, [82](#)
- GetOrigin
 - cpm_DomainInfo, [25](#)
- GetPeriodicRankID
 - cpm_ParaManager, [83](#)
 - cpm_VoxelInfo, [123](#)
- GetPitch
 - cpm_DomainInfo, [25](#)
 - cpm_ParaManager, [83](#)
 - cpm_VoxelInfo, [123](#)
- GetPos
 - cpm_ActiveSubdomainInfo, [17](#)
- getRankNull
 - cpm_Base, [20](#)
- GetRegion
 - cpm_DomainInfo, [25](#)
- GetSpanTime
 - cpm_Base, [21](#)
- GetSubdomainArraySize
 - cpm_GlobalDomainInfo, [30](#)
- GetSubdomainInfo
 - cpm_GlobalDomainInfo, [30](#)
- GetSubdomainNum
 - cpm_GlobalDomainInfo, [31](#)
- GetTime
 - cpm_Base, [21](#)
- GetVoxNum
 - cpm_DomainInfo, [25](#)

- GetVoxelHeadIndex
 - cpm_ParaManager, [83](#)
 - cpm_VoxelInfo, [123](#)
- GetVoxelTailIndex
 - cpm_ParaManager, [84](#)
 - cpm_VoxelInfo, [124](#)
- GetWSpanTime
 - cpm_Base, [21](#)
- GetWTime
 - cpm_Base, [21](#)
- Init
 - cpm_VoxelInfo, [124](#)
- InitArray
 - cpm_ParaManager, [84](#)
- Initialize
 - cpm_ParaManager, [84](#), [85](#)
- Irecv
 - cpm_ParaManager, [85](#)
- IsCommNull
 - cpm_Base, [22](#)
- IsExistSubdomain
 - cpm_GlobalDomainInfo, [31](#)
- IsInnerBoundary
 - cpm_ParaManager, [87](#)
 - cpm_VoxelInfo, [124](#)
- isMatchEndianSbdmMagick
 - cpm_GlobalDomainInfo, [31](#)
- IsOuterBoundary
 - cpm_ParaManager, [87](#)
 - cpm_VoxelInfo, [125](#)
- IsParallel
 - cpm_ParaManager, [87](#), [88](#)
- IsRankNull
 - cpm_Base, [22](#)
- Isend
 - cpm_ParaManager, [86](#)
- m_DelKeyList
 - cpm_ObjList, [37](#)
- m_ObjectMap
 - cpm_ObjList, [38](#)
- m_bndCommInfoMap
 - cpm_ParaManager, [110](#)
- m_bufX
 - S_BNDCOMM_BUFFER, [128](#)
- m_bufY
 - S_BNDCOMM_BUFFER, [128](#)
- m_bufZ
 - S_BNDCOMM_BUFFER, [128](#)
- m_comm
 - cpm_VoxelInfo, [125](#)
- m_divNum
 - cpm_GlobalDomainInfo, [33](#)
- m_globalDomainInfo
 - cpm_VoxelInfo, [125](#)
- m_localDomainInfo
 - cpm_VoxelInfo, [125](#)
- m_maxN
 - S_BNDCOMM_BUFFER, [128](#)
- m_maxVC
 - S_BNDCOMM_BUFFER, [128](#)
- m_nRank
 - cpm_ParaManager, [110](#)
 - cpm_VoxelInfo, [126](#)
- m_neighborRankID
 - cpm_VoxelInfo, [126](#)
- m_newKey
 - cpm_ObjList, [38](#)
- m_nwX
 - S_BNDCOMM_BUFFER, [128](#)
- m_nwY
 - S_BNDCOMM_BUFFER, [129](#)
- m_nwZ
 - S_BNDCOMM_BUFFER, [129](#)
- m_origin
 - cpm_DomainInfo, [27](#)
- m_periodicRankID
 - cpm_VoxelInfo, [126](#)
- m_pitch
 - cpm_DomainInfo, [27](#)
- m_pos
 - cpm_ActiveSubdomainInfo, [18](#)
- m_procGrpList
 - cpm_ParaManager, [110](#)
- m_rankMap
 - cpm_VoxelInfo, [126](#)
- m_rankNo
 - cpm_ParaManager, [111](#)
 - cpm_VoxelInfo, [126](#)
- m_rankNoMap
 - cpm_ParaManager, [111](#)
- m_region
 - cpm_DomainInfo, [27](#)
- m_reqList
 - cpm_ParaManager, [111](#)
- m_subDomainInfo
 - cpm_GlobalDomainInfo, [33](#)
- m_tp
 - cpm_TextParser, [115](#)
- m_voxNum
 - cpm_DomainInfo, [27](#)
- m_voxelHeadIndex
 - cpm_VoxelInfo, [126](#)
- m_voxelInfoMap
 - cpm_ParaManager, [111](#)
- m_voxelTailIndex
 - cpm_VoxelInfo, [126](#)
- MINUS2PLUS
 - cpm_Define.h, [140](#)
- Match
 - CPM_ENDIAN, [10](#)
- ObjectMap
 - cpm_ObjList, [36](#)
- OmitDots
 - CES, [9](#)
- operator==

- cpm_ActiveSubdomainInfo, 17
- PLUS2MINUS
 - cpm_Define.h, 140
- packX
 - cpm_ParaManager, 88
- packXEx
 - cpm_ParaManager, 88
- packY
 - cpm_ParaManager, 89
- packYEx
 - cpm_ParaManager, 89
- packZ
 - cpm_ParaManager, 90
- packZEx
 - cpm_ParaManager, 90
- PeriodicCommS3D
 - cpm_ParaManager, 91
- PeriodicCommS4D
 - cpm_ParaManager, 92
- PeriodicCommS4DEx
 - cpm_ParaManager, 93, 94
- PeriodicCommV3D
 - cpm_ParaManager, 94, 95
- PeriodicCommV3DEx
 - cpm_ParaManager, 95, 96
- REAL_BUF_TYPE
 - cpm_Define.h, 136
- RankNoMap
 - cpm_ObjList.h, 145
 - cpm_ParaManager.h, 147
- Read
 - cpm_TextParser, 113
 - cpm_TextParserDomain, 116
- ReadActiveSubdomainFile
 - cpm_GlobalDomainInfo, 32
- ReadDomainInfo
 - cpm_TextParserDomain, 117
- ReadMain
 - cpm_TextParserDomain, 117
- ReadSubdomainInfo
 - cpm_TextParserDomain, 117
- readVector
 - cpm_TextParser, 113, 114
- ReallsDouble
 - cpm_Base, 22
- Recv
 - cpm_ParaManager, 96, 97
- S_BNDCOMM_BUFFER, 127
 - ~S_BNDCOMM_BUFFER, 127
 - m_bufX, 128
 - m_bufY, 128
 - m_bufZ, 128
 - m_maxN, 128
 - m_maxVC, 128
 - m_nwX, 128
 - m_nwY, 129
 - m_nwZ, 129
 - S_BNDCOMM_BUFFER, 127
 - S_BNDCOMM_BUFFER, 127
- SBSWAPVEC
 - CPM_ENDIAN, 11
- Send
 - cpm_ParaManager, 97
- sendrecv
 - cpm_ParaManager, 98
- SetBndCommBuffer
 - cpm_ParaManager, 98
- SetDivNum
 - cpm_GlobalDomainInfo, 32
- SetOrigin
 - cpm_DomainInfo, 26
- SetPitch
 - cpm_DomainInfo, 26
- SetPos
 - cpm_ActiveSubdomainInfo, 17
- SetRegion
 - cpm_DomainInfo, 26
- SetVoxNum
 - cpm_DomainInfo, 26
- Unknown
 - CPM_ENDIAN, 10
- UnMatch
 - CPM_ENDIAN, 10
- unpackX
 - cpm_ParaManager, 99
- unpackXEx
 - cpm_ParaManager, 99
- unpackY
 - cpm_ParaManager, 100
- unpackYEx
 - cpm_ParaManager, 100
- unpackZ
 - cpm_ParaManager, 101
- unpackZEx
 - cpm_ParaManager, 101
- VersionInfo
 - cpm_Base, 22, 23
- VoxelInfoMap
 - cpm_ParaManager.h, 147
- VoxelInit
 - cpm_ParaManager, 102, 103
- VoxelInit_Subdomain
 - cpm_ParaManager, 103, 104
- Wait
 - cpm_ParaManager, 104
- wait_BndCommS3D
 - cpm_ParaManager, 105
- wait_BndCommS4D
 - cpm_ParaManager, 106
- wait_BndCommS4DEx
 - cpm_ParaManager, 107
- wait_BndCommV3D

- cpm_ParaManager, [108](#)
- wait_BndCommV3DEx
 - cpm_ParaManager, [109](#)
- Waitall
 - cpm_ParaManager, [110](#)
- X_DIR
 - cpm_Define.h, [137](#)
- X_MINUS
 - cpm_Define.h, [140](#)
- X_PLUS
 - cpm_Define.h, [140](#)
- Y_DIR
 - cpm_Define.h, [137](#)
- Y_MINUS
 - cpm_Define.h, [140](#)
- Y_PLUS
 - cpm_Define.h, [140](#)
- Z_DIR
 - cpm_Define.h, [137](#)
- Z_MINUS
 - cpm_Define.h, [140](#)
- Z_PLUS
 - cpm_Define.h, [140](#)