Cartesian Partition Manager Library

2.1.5

作成： Doxygen 1.8.5

Thu Oct 27 2016 08:50:13

Contents

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。

Chapter 2

階層索引

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。

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。

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。

Chapter 5

ネームスペース

5.1 ネームスペース BCMFileIO

構成

- struct OctHeader

  Octree **ファイルヘッダ構造体**
- struct LBHeader

  LeafBlock **ファイルヘッダ構造体**
- struct LBCellIDHeader

  LeafBlock **の***CellID* **ヘッダ構造体**
- struct GridRleCode

  RLE **圧縮符号の走査用構造体**
- struct IdxUnit

  インデックスファイル用単位系情報
- struct IdxProc

  インデックスファイル用プロセス情報
- class BitVoxel

  ビットボクセル圧縮/展開ライブラリ

型定義

- typedef BitVoxel::bitVoxelCell bitVoxelCell

列挙型

- enum LB_KIND {
  LB_CELLID = 0, LB_SCALAR = 1, LB_VECTOR3 = 3, LB_VECTOR4 = 4,
  LB_VECTOR6 = 6, LB_TENSOR = 9 }

  リーフブロックデータタイプ
- enum LB_DATA_TYPE {
  LB_INT8 = 0, LB_UINT8 = 1, LB_INT16 = 2, LB_UINT16 = 3,
  LB_INT32 = 4, LB_UINT32 = 5, LB_INT64 = 6, LB_UINT64 = 7,
  LB_FLOAT32 = 8, LB_FLOAT64 = 9 }

  リーフセルのデータ識別子

関数

- static void BSwap16 (void *a)
  2byte **用エンディアンスワップ**
- static void BSwap32 (void *a)
  4byte **用エンディアンスワップ**
- static void BSwap64 (void *a)
  8byte **用エンディアンスワップ**

変数

- struct BCMFileIO::OctHeader ALIGNMENT

5.1.1 型定義

5.1.1.1 typedef BitVoxel::bitVoxelCell BCMFileIO::bitVoxelCell

BCMFileCommon.h の 84 行で定義されています。

5.1.2 列挙型

5.1.2.1 enum BCMFileIO::LB_DATA_TYPE

リーフセルのデータ識別子

列挙型の値

LB_INT8 符号付き 8bit 整数型
LB_UINT8 符号なし 8bit 整数型
LB_INT16 符号付き 16bit 整数型
LB_UINT16 符号なし 16bit 整数型
LB_INT32 符号付き 32bit 整数型
LB_UINT32 符号なし 32bit 整数型
LB_INT64 符号付き 64bit 整数型
LB_UINT64 符号なし 64bit 整数型
LB_FLOAT32 32bit 浮動小数点 (単精度浮動小数点)
LB_FLOAT64 64bit 浮動小数点 (倍精度浮動小数点)

BCMFileCommon.h の 113 行で定義されています。

5.1.2.2 enum BCMFileIO::LB_KIND

リーフブロックデータタイプ

列挙型の値

LB_CELLID グリッド
LB_SCALAR スカラ
LB_VECTOR3 ベクトル (3 要素)
LB_VECTOR4 ベクトル (4 要素)
LB_VECTOR6 ベクトル (6 要素)
LB_TENSOR テンソル (9 要素)

BCMFileCommon.h の 102 行で定義されています。

5.1.3 関数

5.1.3.1 static void BCMFileIO::BSwap16 (void * *a*) [inline], [static]

2byte 用エンディアンスワップ

BCMFileCommon.h の 146 行で定義されています。

5.1.3.2 static void BCMFileIO::BSwap32 (void * *a*) [inline], [static]

4byte 用エンディアンスワップ

BCMFileCommon.h の 152 行で定義されています。

参照元 cpm_VoxelInfoLMR::LoadOctreeHeader().

5.1.3.3 static void BCMFileIO::BSwap64 (void * *a*) [inline], [static]

8byte 用エンディアンスワップ

BCMFileCommon.h の 159 行で定義されています。

参照元 cpm_VoxelInfoLMR::LoadOctreeFile(), と cpm_VoxelInfoLMR::LoadOctreeHeader().

5.1.4 変数

5.1.4.1 struct BCMFileIO::GridRleCode BCMFileIO::ALIGNMENT

5.2 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= '/')

5.2.1 関数

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

cpm_PathUtil.h の 65 行で定義されています。

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

cpm_PathUtil.h の 25 行で定義されています。

参照元 cpm_TextParserDomainLMR::ReadBCMTree(), と cpm_TextParserDomain::ReadSubdomainInfo().

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

cpm_PathUtil.h の 105 行で定義されています。

参照元 CPM_PATH::cpmPath_normalize().

5.3 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)

5.3.1 説明

CPM のエンディアンユーティリティ名前空間

5.3.2 列挙型

5.3.2.1 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 113 行で定義されています。

5.3.3 関数

5.3.3.1 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

in,out	*x*	変換する変数

cpm_EndianUtil.h の 30 行で定義されています。

参照元 SBSWAPVEC().

5.3.3.2 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

in,out	*x*	変換する変数

cpm_EndianUtil.h の 41 行で定義されています。

参照元 BSWAPVEC().

5.3.3.3 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

in,out	*x*	変換する変数

cpm_EndianUtil.h の 55 行で定義されています。

参照元 DBSWAPVEC().

5.3.3.4 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

in,out	*a*	変換する変数配列
in	*n*	配列要素数

cpm_EndianUtil.h の 90 行で定義されています。

参照先 BSWAP32().

参照元 cpm_GlobalDomainInfo::ReadActiveSubdomainFile().

5.3.3.5 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

in,out	*a*	変換する変数配列
in	*n*	配列要素数

cpm_EndianUtil.h の 103 行で定義されています。

参照先 BSWAP64().

5.3.3.6 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

in,out	*a*	変換する変数配列
in	*n*	配列要素数

cpm_EndianUtil.h の 76 行で定義されています。

参照先 BSWAP16().

5.4 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)

5.4.1 関数

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

cpm_PathUtil.h の 172 行で定義されています。

参照元 cpmPath_normalize().

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

cpm_PathUtil.h の 214 行で定義されています。

参照先 cpmPath_getDelimChar().

参照元 cpm_TextParserDomainLMR::ReadBCMTree(), と cpm_TextParserDomain::ReadSubdomainInfo().

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

cpm_PathUtil.h の 196 行で定義されています。

参照先 cpmPath_hasDrive().

参照元 cpmPath_isAbsolute(), と cpmPath_normalize().

5.4.1.4 char CPM_PATH::cpmPath_getDelimChar ( ) [inline]

cpm_PathUtil.h の 164 行で定義されています。

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

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

cpm_PathUtil.h の 187 行で定義されています。

参照元 cpmPath_emitDrive().

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

cpm_PathUtil.h の 204 行で定義されています。

参照先 cpmPath_emitDrive(), と cpmPath_getDelimChar().

参照元 cpm_TextParserDomainLMR::ReadBCMTree(), と cpm_TextParserDomain::ReadSubdomainInfo().

5.4.1.7 std::string CPM_PATH::cpmPath_normalize (const std::string & *path*) [inline]

cpm_PathUtil.h の 225 行で定義されています。

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

5.5 ネームスペース Vec3class

構成

- class Vec3

型定義

- typedef Vec3< unsigned char > Vec3uc
- typedef Vec3< int > Vec3i
- typedef Vec3< float > Vec3f
- typedef Vec3< double > Vec3d
- typedef Vec3< REAL_TYPE > Vec3r

列挙型

- enum AxisEnum { AXIS_X = 0, AXIS_Y, AXIS_Z, AXIS_ERROR }

関数

- template<typename T >
  Vec3< T > operator* (T s, const Vec3< T > &v)
- template<typename T >
  Vec3< T > multi (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T dot (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  Vec3< T > cross (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T distanceSquared (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T distance (const Vec3< T > &a, const Vec3< T > &b)
- bool lessVec3f (const Vec3f &a, const Vec3f &b)
- template<typename T >
  std::istream & operator>> (std::istream &is, Vec3< T > &v)
- template<typename T >
  std::ostream & operator<< (std::ostream &os, const Vec3< T > &v)
- std::istream & operator>> (std::istream &is, Vec3uc &v)
- std::ostream & operator<< (std::ostream &os, const Vec3uc &v)

5.5.1 型定義

5.5.1.1 typedef Vec3<double> Vec3class::Vec3d

Vec3.h の 210 行で定義されています。

5.5.1.2 typedef Vec3<float> Vec3class::Vec3f

Vec3.h の 209 行で定義されています。

5.5.1.3 typedef Vec3<int> Vec3class::Vec3i

Vec3.h の 208 行で定義されています。

5.5.1.4 typedef Vec3<REAL_TYPE> Vec3class::Vec3r

Vec3.h の 211 行で定義されています。

5.5.1.5 typedef Vec3<unsigned char> Vec3class::Vec3uc

Vec3.h の 207 行で定義されています。

5.5.2 列挙型

5.5.2.1 enum Vec3class::AxisEnum

列挙型の値

AXIS_X
AXIS_Y
AXIS_Z
AXIS_ERROR

Vec3.h の 50 行で定義されています。

5.5.3 関数

5.5.3.1 template<typename T > Vec3<T> Vec3class::cross (const Vec3< T > & *a,* const Vec3< T > & *b*) [inline]

Vec3.h の 233 行で定義されています。
参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

5.5.3.2 template<typename T > T Vec3class::distance (const Vec3< T > & *a,* const Vec3< T > & *b*) [inline]

Vec3.h の 245 行で定義されています。

5.5.3.3 template<typename T > T Vec3class::distanceSquared (const Vec3< T > & *a,* const Vec3< T > & *b*) [inline]

Vec3.h の 240 行で定義されています。

5.5.3.4 template<typename T > T Vec3class::dot (const Vec3< T > & *a,* const Vec3< T > & *b*) [inline]

Vec3.h の 228 行で定義されています。
参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

5.5.3.5 bool Vec3class::lessVec3f (const Vec3f & *a*, const Vec3f & *b*) [inline]

Vec3.h の 250 行で定義されています。

参照先 Vec3class::Vec3< T >::lengthSquared().

5.5.3.6 template<typename T > Vec3<T> Vec3class::multi (const Vec3< T > & *a*, const Vec3< T > & *b*) [inline]

Vec3.h の 223 行で定義されています。

5.5.3.7 template<typename T > Vec3<T> Vec3class::operator* (T *s*, const Vec3< T > & *v*) [inline]

Vec3.h の 218 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

5.5.3.8 template<typename T > std::ostream& Vec3class::operator<< (std::ostream & *os*, const Vec3< T > & *v*) [inline]

Vec3.h の 265 行で定義されています。

5.5.3.9 std::ostream& Vec3class::operator<< (std::ostream & *os*, const Vec3uc & *v*) [inline]

Vec3.h の 280 行で定義されています。

5.5.3.10 template<typename T > std::istream& Vec3class::operator>> (std::istream & *is*, Vec3< T > & *v*) [inline]

Vec3.h の 259 行で定義されています。

5.5.3.11 std::istream& Vec3class::operator>> (std::istream & *is*, Vec3uc & *v*) [inline]

Vec3.h の 272 行で定義されています。

Chapter 6

クラス

6.1 クラス BCMOctree

`#include <BCMOctree.h>`

BCMOctree のコラボレーション図

Public 型

- enum Ordering { Z, HILBERT, RANDOM, PEDIGREELIST }
  オーダリングタイプ.

Public メソッド

- BCMOctree (RootGrid *rootGrid, Divider *divider, Ordering ordering)
- BCMOctree (RootGrid *rootGrid, const std::vector< Pedigree > &pedigrees)
- ~BCMOctree ()
  デストラクタ.
- void broadcast (MPI::Intracomm &comm=MPI::COMM_WORLD)
- const RootGrid * getRootGrid () const
- int getNumLeafNode () const
- std::vector< Node * > & getLeafNodeArray ()
- const std::vector< Node * > & getLeafNodeArray () const
- bool checkOnOuterBoundary (const Node *node, Face face) const
- Vec3d getOrigin (const Node *node) const
- NeighborInfo * makeNeighborInfo (const Node *node, const Partition *partition) const

Static Public メソッド

- static BCMOctree * ReceiveFromMaster (MPI::Intracomm &comm=MPI::COMM_WORLD)

Private メソッド

- BCMOctree (RootGrid *rootGrid, Ordering ordering, int numLeafNode, const unsigned char *buf)
- void buildTreeFromPedigreeList (int numLeafNode, const unsigned char *buf)
- void pickupLeafNodeZOrdering (Node *node)
- void pickupLeafNodeHilbertOrdering (Node *node, int orientation)
- void randomShuffle ()

リーフノードリストをランダムシャッフル.
- void makeNode (Node *node)
- void deleteNode (Node *node)
- Node * findNeighborNode (const Node *node, Face face) const
- void packPedigrees (Node *node, size_t &ip, unsigned char *buf)

Private 変数

- RootGrid * rootGrid
  ルートノード配置情報
- Divider * divider
  ブロック分割判定クラス
- Ordering ordering
  オーダリング方法
- Node ** rootNodes
  ルートノード配列
- std::vector< Node * > leafNodeArray
  リーフノードリスト

Static Private 変数

- static const int HilbertOrdering [24][8]
  ヒルベルトオーダリング 子ノード選択順テーブル
- static const int HilbertOrientation [24][8]
  ヒルベルトオーダリング 回転テーブル

6.1.1 説明

BCM 用マルチルートOCtree クラス.

BCMOctree.h の 34 行で定義されています。

6.1.2 列挙型

6.1.2.1 enum BCMOctree::Ordering

オーダリングタイプ.

列挙型の値

Z Z(Morton) オーダリング
HILBERT ヒルベルトオーダリング
RANDOM ランダムシャッフル
PEDIGREELIST Pedigree リスト順 (ファイルロード用)

BCMOctree.h の 39 行で定義されています。

6.1.3 コンストラクタとデストラクタ

6.1.3.1 BCMOctree::BCMOctree (RootGrid * *rootGrid*, Divider * *divider*, Ordering *ordering*)

コンストラクタ.

引数

in	*rootGrid*	ルートノード配置情報
in	*divider*	ブロック分割判定クラス
in	*ordering*	オーダリング方法

覚え書き

rootGrid と divider は、デストラクタにより解放される.

BCMOctree.cpp の 17 行で定義されています。

参照先 RootGrid::getSize(), HILBERT, makeNode(), pickupLeafNodeHilbertOrdering(), pickupLeafNodeZ-Ordering(), RANDOM, randomShuffle(), と rootNodes.

参照元 ReceiveFromMaster().

6.1.3.2 BCMOctree::BCMOctree (RootGrid * *rootGrid,* const std::vector< Pedigree > & *pedigrees*)

コンストラクタ.

引数

in	*rootGrid*	ルートノード配置情報
in	*pedigrees*	ペディグリリスト

覚え書き

rootGrid は、デストラクタにより解放される.

BCMOctree.cpp の 60 行で定義されています。

参照先 Node::getChild(), RootGrid::getSize(), Node::isLeafNode(), leafNodeArray, Node::makeChildNodes(), root-Nodes, Node::setActive(), と Node::setBlockID().

6.1.3.3 BCMOctree::~BCMOctree ( )

デストラクタ.

BCMOctree.cpp の 93 行で定義されています。

参照先 divider, RootGrid::getSize(), rootGrid, と rootNodes.

6.1.3.4 BCMOctree::BCMOctree (RootGrid * *rootGrid,* Ordering *ordering,* int *numLeafNode,* const unsigned char * *buf*) [private]

コンストラクタ (リーフノードのPedigree リストから).

引数

in	*rootGrid*	ルートノード配置情報
in	*ordering*	オーダリング方法
in	*numLeafNode*	リーフノード総数
in	*buf*	シリアライズされたPedigree リスト

BCMOctree.cpp の 39 行で定義されています。

参照先 buildTreeFromPedigreeList(), RootGrid::getSize(), HILBERT, pickupLeafNodeHilbertOrdering(), pickupLeaf-NodeZOrdering(), RANDOM, randomShuffle(), と rootNodes.

6.1.4 関数

6.1.4.1 void BCMOctree::broadcast (MPI::Intracomm & *comm* = MPI::COMM_WORLD)

Octree 情報を他 rank にブロードキャスト.

引数

in	*comm*	MPI コミュニケータ

覚え書き

rank0 のみが呼ぶこと

BCMOctree.cpp の 122 行で定義されています。

参照先 RootGrid::broadcast(), Pedigree::GetSerializeSize(), RootGrid::getSize(), leafNodeArray, ordering, packPedigrees(), rootGrid, と rootNodes.

6.1.4.2 void BCMOctree::buildTreeFromPedigreeList (int *numLeafNode,* const unsigned char * *buf*) [private]

リーフノードのPedigree リストからOctree を再構築.

引数

in	*numLeafNode*	リーフノード総数
in	*buf*	シリアライズされたPedigree リスト

BCMOctree.cpp の 200 行で定義されています。

参照先 Pedigree::deserialize(), Node::getChild(), Pedigree::getChildId(), Pedigree::getLevel(), Pedigree::getRootID(), Pedigree::GetSerializeSize(), RootGrid::getSize(), Node::isLeafNode(), Node::makeChildNodes(), rootGrid, rootNodes, と Node::setActive().

参照元 BCMOctree().

6.1.4.3 bool BCMOctree::checkOnOuterBoundary (const Node * *node,* Face *face*) const

指定したノードの面が外部境界 (周期境界も含む) かどうかチェック.

引数

in	*node*	ノード
in	*face*	面

戻り値

外部境界なら true

BCMOctree.cpp の 362 行で定義されています。

参照先 Node::getPedigree(), Pedigree::getRootID(), Pedigree::getUpperBound(), Pedigree::getX(), Pedigree::getY(), Pedigree::getZ(), RootGrid::isOuterBoundary(), rootGrid, X_M, X_P, Y_M, Y_P, Z_M, と Z_P.

6.1.4.4 void BCMOctree::deleteNode (Node * *node*) [private]

再帰呼び出しによるツリー消去.

引数

in	*node*	ノード

BCMOctree.cpp の 260 行で定義されています。

参照先 Node::getChild(), と Node::isLeafNode().

6.1.4.5 Node * BCMOctree::findNeighborNode (const Node * *node*, Face *face*) const [private]

隣接ノード探索.

引数

in	*node*	基準ノード
in	*face*	面

戻り値

隣接ノードへのポインタ

覚え書き

隣接ノードが非アクティブな場合でも，それを返す. 周期境界条件以外の外部境界に接している場合のみ 0 を返す.

BCMOctree.cpp の 271 行で定義されています。

参照先 EX_FAILURE, Exit, Node::getChild(), Pedigree::getLevel(), RootGrid::getNeighborRoot(), Node::getPedigree(), Pedigree::getRootID(), Pedigree::getUpperBound(), Pedigree::getX(), Pedigree::getY(), Pedigree::getZ(), Node::isLeafNode(), rootGrid, rootNodes, X_M, X_P, Y_M, Y_P, Z_M, と Z_P.

参照元 makeNeighborInfo().

6.1.4.6 std::vector<Node*>& BCMOctree::getLeafNodeArray ( ) [inline]

リーフノードリストを取得.

戻り値

リーフノードリスト (leafNodeArray)

BCMOctree.h の 112 行で定義されています。

参照元 cpm_VoxelInfoLMR::Init().

6.1.4.7 const std::vector<Node*>& BCMOctree::getLeafNodeArray ( ) const [inline]

リーフノードリストを取得.

戻り値

リーフノードリスト (leafNodeArray)

BCMOctree.h の 116 行で定義されています。

6.1.4.8 int BCMOctree::getNumLeafNode ( ) const [inline]

リーフノード総数を取得.

戻り値

リーフノード総数 (leafNodeArray のサイズ)

BCMOctree.h の 108 行で定義されています。

6.1.4.9 Vec3d BCMOctree::getOrigin (const Node * *node*) const

指定されたノードの原点位置を取得.

引数

in	*node*	ノード

戻り値

原点位置

BCMOctree.cpp の 402 行で定義されています。

参照先 Node::getPedigree(), Pedigree::getRootID(), Pedigree::getUpperBound(), Pedigree::getX(), Pedigree::getY(), Pedigree::getZ(), rootGrid, RootGrid::rootID2indexX(), RootGrid::rootID2indexY(), と RootGrid::rootID2indexZ().

参照元 cpm_VoxelInfoLMR::Init(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.1.4.10 const RootGrid* BCMOctree::getRootGrid ( ) const [inline]

ルートグリッドを取得

戻り値

ルートグリッド

BCMOctree.h の 104 行で定義されています。

参照元 cpm_VoxelInfoLMR::SetLocalDomainInfo(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.1.4.11 NeighborInfo * BCMOctree::makeNeighborInfo (const Node * *node,* const Partition * *partition*) const

指定されたノードの隣接情報を計算.

引数

in	*node*	ノード
in	*partition*	領域分割情報

戻り値

隣接情報クラス

BCMOctree.cpp の 417 行で定義されています。

参照先 NeighborInfo::childIdToSubface(), EX_FAILURE, Exit, findNeighborNode(), Node::getBlockID(), Node::getChild(), Pedigree::getChildId(), Node::getLevel(), NeighborInfo::getNeighborChildId(), Node::getPedigree(),

Partition::getRank(), Node::isActive(), Node::isLeafNode(), NUM_FACE, NUM_SUBFACE, NeighborInfo::setID(), NeighborInfo::setLevelDifference(), NeighborInfo::setNeighborSubface(), と NeighborInfo::setRank().

参照元 cpm_VoxelInfoLMR::SetNeighborInfo().

6.1.4.12 void BCMOctree::makeNode (Node * *node*) [private]

再帰呼び出しによりツリー生成.

引数

in	*node*	ノード

覚え書き

内部で divider クラスにより分割判定を行っている

BCMOctree.cpp の 105 行で定義されています。

参照先 Node::getChild(), Node::getPedigree(), Divider::LEAF_ACTIVE, Divider::LEAF_NO_ACTIVE, Node::makeChildNodes(), と Node::setActive().

参照元 BCMOctree().

6.1.4.13 void BCMOctree::packPedigrees (Node * *node*, size_t & *ip*, unsigned char * *buf*) [private]

Pedigree 情報を通信用バッファにパック.

引数

in	*node*	対象ノード
in	*ip*	バッファ内の位置インデクス
in	*buf*	バッファ配列

BCMOctree.cpp の 147 行で定義されています。

参照先 Node::getChild(), Node::getPedigree(), Pedigree::GetSerializeSize(), Node::isActive(), と Node::isLeafNode().

参照元 broadcast().

6.1.4.14 void BCMOctree::pickupLeafNodeHilbertOrdering (Node * *node*, int *orientation*) [private]

ヒルベルトオーダリング順にリーフノードリストを作成.

引数

in	*node*	ルートノード
in	*orientation*	回転向き

覚え書き

再帰呼び出しされる

BCMOctree.cpp の 242 行で定義されています。

参照先 Node::getChild(), HilbertOrdering, HilbertOrientation, Node::isActive(), Node::isLeafNode(), leafNodeArray, と Node::setBlockID().

参照元 BCMOctree().

6.1.4.15 void BCMOctree::pickupLeafNodeZOrdering (Node * *node*) [private]

Z オーダリング順にリーフノードリストを作成.

引数

in	*node*	ルートノード

覚え書き

再帰呼び出しされる

BCMOctree.cpp の 228 行で定義されています。

参照先 Node::getChild(), Node::isActive(), Node::isLeafNode(), leafNodeArray, と Node::setBlockID().

参照元 BCMOctree().

6.1.4.16 void BCMOctree::randomShuffle ( ) [private]

リーフノードリストをランダムシャッフル.

BCMOctree.cpp の 394 行で定義されています。

参照先 leafNodeArray.

参照元 BCMOctree().

6.1.4.17 BCMOctree * BCMOctree::ReceiveFromMaster (MPI::Intracomm & *comm* = MPI::COMM_WORLD) [static]

rank0 からOctree 情報を受信.

引数

in	*comm*	MPI コミュニケータ

戻り値

新たに生成したBCMOctree インスタンス

覚え書き

rank0 からは呼ばないこと

BCMOctree.cpp の 176 行で定義されています。

参照先 BCMOctree(), Pedigree::GetSerializeSize(), ordering, RootGrid::ReceiveFromMaster(), と rootGrid.

6.1.5 変数

6.1.5.1 Divider* BCMOctree::divider [private]

ブロック分割判定クラス

BCMOctree.h の 50 行で定義されています。

参照元 ~BCMOctree().

6.1.5.2 const int BCMOctree::HilbertOrdering [static], [private]

初期値:

```
= {
  {0, 1, 3, 2, 6, 7, 5, 4},
  {0, 4, 6, 2, 3, 7, 5, 1},
  {0, 1, 5, 4, 6, 7, 3, 2},
  {5, 1, 0, 4, 6, 2, 3, 7},
  {3, 7, 6, 2, 0, 4, 5, 1},
  {6, 7, 3, 2, 0, 1, 5, 4},
  {5, 1, 3, 7, 6, 2, 0, 4},
  {0, 4, 5, 1, 3, 7, 6, 2},
  {5, 4, 0, 1, 3, 2, 6, 7},
  {5, 4, 6, 7, 3, 2, 0, 1},
  {0, 2, 3, 1, 5, 7, 6, 4},
  {6, 4, 0, 2, 3, 1, 5, 7},
  {5, 7, 3, 1, 0, 2, 6, 4},
  {3, 7, 5, 1, 0, 4, 6, 2},
  {6, 4, 5, 7, 3, 1, 0, 2},
  {0, 2, 6, 4, 5, 7, 3, 1},
  {6, 2, 0, 4, 5, 1, 3, 7},
  {6, 2, 3, 7, 5, 1, 0, 4},
  {3, 2, 0, 1, 5, 4, 6, 7},
  {6, 7, 5, 4, 0, 1, 3, 2},
  {5, 7, 6, 4, 0, 2, 3, 1},
  {3, 2, 6, 7, 5, 4, 0, 1},
  {3, 1, 0, 2, 6, 4, 5, 7},
  {3, 1, 5, 7, 6, 4, 0, 2},
}
```

ヒルベルトオーダリング 子ノード選択順テーブル

BCMOctree.h の 58 行で定義されています。

参照元 pickupLeafNodeHilbertOrdering().

6.1.5.3 const int BCMOctree::HilbertOrientation [static], [private]

初期値:

```
= {
  {1, 2, 0, 3, 4, 0, 5, 6},
  {0, 7, 1, 8, 5, 1, 4, 9},
  {15, 0, 2, 22, 20, 2, 19, 23},
  {20, 6, 3, 23, 15, 3, 16, 22},
  {22, 13, 4, 12, 11, 4, 1, 20},
  {11, 19, 5, 20, 22, 5, 0, 12},
  {9, 3, 6, 2, 21, 6, 17, 0},
  {10, 1, 7, 11, 12, 7, 13, 14},
  {12, 9, 8, 14, 10, 8, 18, 11},
  {6, 8, 9, 7, 17, 9, 21, 1},
  {7, 15, 10, 16, 13, 10, 12, 17},
  {5, 14, 11, 9, 0, 11, 22, 8},
  {8, 20, 12, 19, 18, 12, 10, 5},
  {18, 4, 13, 5, 8, 13, 7, 19},
  {17, 11, 14, 1, 6, 14, 23, 7},
  {2, 10, 15, 18, 19, 15, 20, 21},
  {19, 17, 16, 21, 2, 16, 3, 18},
  {14, 16, 17, 15, 23, 17, 6, 10},
  {13, 21, 18, 17, 7, 18, 8, 16},
  {16, 5, 19, 4, 3, 19, 2, 13},
  {3, 12, 20, 13, 16, 20, 15, 4},
  {23, 18, 21, 10, 14, 21, 9, 15},
  {4, 23, 22, 6, 1, 22, 11, 3},
  {21, 22, 23, 0, 9, 23, 14, 2},
}
```

ヒルベルトオーダリング 回転テーブル

BCMOctree.h の 59 行で定義されています。

参照元 pickupLeafNodeHilbertOrdering().

6.1.5.4 std::vector<Node*> BCMOctree::leafNodeArray [private]

リーフノードリスト

BCMOctree.h の 56 行で定義されています。

参照元 BCMOctree(), broadcast(), pickupLeafNodeHilbertOrdering(), pickupLeafNodeZOrdering(), と randomShuffle().

6.1.5.5 Ordering BCMOctree::ordering [private]

オーダリング方法

BCMOctree.h の 52 行で定義されています。

参照元 broadcast(), と ReceiveFromMaster().

6.1.5.6 RootGrid* BCMOctree::rootGrid [private]

ルートノード配置情報

BCMOctree.h の 48 行で定義されています。

参照元 broadcast(), buildTreeFromPedigreeList(), checkOnOuterBoundary(), findNeighborNode(), getOrigin(), ReceiveFromMaster(), と ~BCMOctree().

6.1.5.7 Node** BCMOctree::rootNodes [private]

ルートノード配列

BCMOctree.h の 54 行で定義されています。

参照元 BCMOctree(), broadcast(), buildTreeFromPedigreeList(), findNeighborNode(), と ~BCMOctree().

このクラスの説明は次のファイルから生成されました:

- BCMOctree.h
- BCMOctree.cpp

6.2 クラス BCMFileIO::BitVoxel

ビットボクセル圧縮/展開ライブラリ

`#include <BitVoxel.h>`

Public 型

- typedef unsigned int bitVoxelCell
  ビットボクセル型の定義

Public メソッド

- BitVoxel ()
  コンストラクタ
- ~BitVoxel ()
  デストラクタ

Static Public メソッド

- static size_t GetSize (const size_t sourceSize, const unsigned char bitWidth)

- static bitVoxelCell * Compress (size_t *bitVoxelSize, const size_t voxelSize, const unsigned char *voxel, const unsigned char bitWidth)
- static unsigned char * Decompress (const size_t voxelSize, const bitVoxelCell *bitVoxel, const unsigned char bitWidth)

6.2.1 説明

ビットボクセル圧縮/展開ライブラリ

BitVoxel.h の 23 行で定義されています。

6.2.2 型定義

6.2.2.1 typedef unsigned int BCMFileIO::BitVoxel::bitVoxelCell

ビットボクセル型の定義

BitVoxel.h の 27 行で定義されています。

6.2.3 コンストラクタとデストラクタ

6.2.3.1 BCMFileIO::BitVoxel::BitVoxel ( )

コンストラクタ

6.2.3.2 BCMFileIO::BitVoxel::~BitVoxel ( )

デストラクタ

6.2.4 関数

6.2.4.1 static bitVoxelCell* BCMFileIO::BitVoxel::Compress (size_t * *bitVoxelSize*, const size_t *voxelSize*, const unsigned char * *voxel*, const unsigned char *bitWidth*) [static]

ビットボクセル圧縮

引数

out	*bitVoxelSize*	出力ビットボクセルサイズ
in	*boxelSize*	入力ボクセルサイズ
in	*voxel*	入力ボクセルの先頭ポインタ
in	*bitWidth*	ビット幅

戻り値

ビットボクセルの先頭ポインタ

覚え書き

return されたポインタは適宜解放 (delete) してください.

6.2.4.2 static unsigned char* BCMFileIO::BitVoxel::Decompress (const size_t *voxelSize*, const bitVoxelCell * *bitVoxel*, const unsigned char *bitWidth*) [static]

ビットボクセル展開

引数

in	*bitVoxelSize*	ボクセルサイズ (展開後のボクセル数)
in	*bitVoxel*	入力ビットボクセル
in	*bitWidth*	ビット幅

戻り値

展開されたボクセルの先頭ポインタ

覚え書き

return されたポインタは適宜解放 (delete) してください.

6.2.4.3 static size_t BCMFileIO::BitVoxel::GetSize (const size_t *sourceSize*, const unsigned char *bitWidth*) [static]

ボクセルをビットボクセル化した場合のビットボクセルサイズを出力

引数

in	*sourceSize*	ボクセル数
in	*bitWidth*	ビット幅

戻り値

ビットボクセルサイズ

覚え書き

ビットボクセルサイズはバイト単位ではない.

このクラスの説明は次のファイルから生成されました:

- BitVoxel.h

6.3 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

Additional Inherited Members

6.3.1 説明

CPM のサブ領域情報クラス

cpm_DomainInfo.h の 122 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_ActiveSubdomainInfo::cpm_ActiveSubdomainInfo ( )

デフォルトコンストラクタ

cpm_DomainInfo.cpp の 171 行で定義されています。

参照先 clear().

6.3.2.2 cpm_ActiveSubdomainInfo::cpm_ActiveSubdomainInfo (int *pos[3]*)

コンストラクタ

引数

in	*pos*	領域分割内での位置

cpm_DomainInfo.cpp の 179 行で定義されています。

参照先 SetPos().

6.3.2.3 cpm_ActiveSubdomainInfo::~cpm_ActiveSubdomainInfo ( ) [virtual]

デストラクタ

cpm_DomainInfo.cpp の 187 行で定義されています。

6.3.3 関数

6.3.3.1 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 194 行で定義されています。

参照先 m_pos.

参照元 cpm_LocalDomainInfo::clear(), と cpm_ActiveSubdomainInfo().

6.3.3.2 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 214 行で定義されています。

参照先 m_pos.

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

6.3.3.3 bool cpm_ActiveSubdomainInfo::operator!= (cpm_ActiveSubdomainInfo *dom*)

比較演算子

引数

in	*dom*	比較対象の活性サブドメイン情報

戻り値

true	違う位置情報を持つ
false	同じ位置情報を持つ

cpm_DomainInfo.cpp の 233 行で定義されています。

参照先 m_pos.

6.3.3.4 bool cpm_ActiveSubdomainInfo::operator== (cpm_ActiveSubdomainInfo *dom*)

比較演算子

引数

in	*dom*	比較対象の活性サブドメイン情報

戻り値

true	同じ位置情報を持つ
false	違う位置情報を持つ

cpm_DomainInfo.cpp の 222 行で定義されています。

参照先 m_pos.

6.3.3.5 void cpm_ActiveSubdomainInfo::SetPos (int *pos[3]*)

位置のセット

引数

in	*pos*	領域分割内での位置

cpm_DomainInfo.cpp の 204 行で定義されています。

参照先 m_pos.

参照元 cpm_ActiveSubdomainInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.3.4 変数

6.3.4.1 int cpm_ActiveSubdomainInfo::m_pos[3] [private]

領域分割内での位置

cpm_DomainInfo.h の 177 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

6.4 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 RealIsDouble ()
- 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 std::string getVersionInfo ()
- static std::string getRevisionInfo ()

Protected メソッド

- cpm_Base ()
- virtual ~cpm_Base ()

6.4.1 説明

CPM のベースクラス

cpm_Base.h の 48 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_Base::cpm_Base ( ) [inline], [protected]

コンストラクタ

cpm_Base.h の 243 行で定義されています。

6.4.2.2 virtual cpm_Base::~cpm_Base ( ) [inline], [protected], [virtual]

デストラクタ

cpm_Base.h の 246 行で定義されています。

6.4.3 関数

6.4.3.1 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	*str1*	文字列 1
in	*str2*	文字列 2
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 192 行で定義されています。

参照元 cpm_strCompareN(), cpm_TextParserDomainLMR::ReadBCMTree(), cpm_TextParserDomainLMR::ReadDomain(), cpm_TextParserDomain::ReadDomainInfo(), cpm_TextParserDomainLMR::ReadLeafBlock(), と cpm_TextParserDomain::ReadSubdomainInfo().

6.4.3.2 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 214 行で定義されています。

参照先 cpm_strCompare().

6.4.3.3 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 76 行で定義されています。

参照元 cpm_BaseParaManager::GetMPI_Comm().

6.4.3.4 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 153 行で定義されています。

6.4.3.5 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 57 行で定義されています。

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

6.4.3.6 static std::string cpm_Base::getRevisionInfo ( ) [inline], [static]

リビジョンを情報を返す

cpm_Base.h の 235 行で定義されています。

参照先 CPM_REVISION.

6.4.3.7 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 123 行で定義されています。
参照先 GetTime().

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

時刻の取得 (gettimeofday 版)

戻り値

時刻

cpm_Base.h の 108 行で定義されています。
参照元 GetSpanTime().

6.4.3.9 static std::string cpm_Base::getVersionInfo ( ) [inline], [static]

バージョンを情報を返す
cpm_Base.h の 227 行で定義されています。
参照先 CPM_VERSION_NO.

6.4.3.10 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 143 行で定義されています。
参照先 GetWTime().

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

時刻の取得 (MPI_Wtime 版)

戻り値

時刻

cpm_Base.h の 133 行で定義されています。
参照元 GetWSpanTime().

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

NULL のMPI_Comm かどうかを確認

戻り値

true	NULL
false	NULL ではない

cpm_Base.h の 85 行で定義されています。

参照元 cpm_BaseParaManager::Allgather(), cpm_BaseParaManager::Allgatherv(), cpm_BaseParaManager::-Allreduce(), cpm_BaseParaManager::Barrier(), cpm_BaseParaManager::Bcast(), cpm_BaseParaManager::Create-ProcessGroup(), cpm_BaseParaManager::Gather(), cpm_BaseParaManager::Gatherv(), cpm_BaseParaManager-::GetMyRankID(), cpm_BaseParaManager::GetNumRank(), cpm_VoxelInfoCART::Init(), cpm_VoxelInfoLMR::Init(), cpm_BaseParaManager::Irecv(), cpm_BaseParaManager::Isend(), cpm_BaseParaManager::Recv(), cpm_Base-ParaManager::Send(), cpm_ParaManager::VoxelInit(), と cpm_ParaManagerLMR::VoxelInit_LMR().

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

NULL のランクかどうかを確認

引数

in	*rankNo*	ランクID

戻り値

true	NULL
false	NULL ではない

cpm_Base.h の 67 行で定義されています。

参照元 cpm_VoxelInfoLMR::GetNeighborRankList(), cpm_VoxelInfoLMR::GetPeriodicRankList(), cpm_VoxelInfo-::IsInnerBoundary(), cpm_VoxelInfo::IsOuterBoundary(), cpm_ParaManager::packX(), cpm_ParaManager::pack-XEx(), cpm_ParaManager::packY(), cpm_ParaManager::packYEx(), cpm_ParaManager::packZ(), cpm_Para-Manager::packZEx(), cpm_ParaManager::sendrecv(), cpm_VoxelInfoLMR::SetNeighborInfo(), cpm_ParaManager-::unpackX(), cpm_ParaManager::unpackXEx(), cpm_ParaManager::unpackY(), cpm_ParaManager::unpackYEx(), cpm_ParaManager::unpackZ(), と cpm_ParaManager::unpackZEx().

6.4.3.14 static CPM_INLINE bool cpm_Base::RealIsDouble ( ) [inline], [static]

fortan の実数型（CPM_REAL）が倍精度かどうか確認

戻り値

true	倍精度
false	単精度

cpm_Base.h の 95 行で定義されています。

参照元 cpm_BaseParaManager::GetMPI_Datatype().

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

6.5 クラス cpm_BaseParaManager

`#include <cpm_BaseParaManager.h>`

cpm_BaseParaManager に対する継承グラフ

cpm_BaseParaManager 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
- int CreateProcessGroup (int nproc, int *proclist, int parentProcGrpNo=0)
- cpm_DomainType GetDomainType ()
- cpm_DefPointType GetDefPointType (int procGrpNo=0)
- virtual const cpm_VoxelInfo * FindVoxelInfo (int procGrpNo=0)=0
- const int * GetGlobalVoxelSize (int procGrpNo=0)
- const int * GetGlobalNodeSize (int procGrpNo=0)
- const int * GetGlobalArraySize (int procGrpNo=0)
- const double * GetGlobalOrigin (int procGrpNo=0)
- const double * GetGlobalRegion (int procGrpNo=0)
- const int * GetLocalVoxelSize (int procGrpNo=0)
- const int * GetLocalNodeSize (int procGrpNo=0)
- const int * GetLocalArraySize (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)
- virtual cpm_ErrorCode SetBndCommBuffer (size_t maxVC, size_t maxN, int procGrpNo=0)=0
- virtual size_t GetBndCommBufferSize (int procGrpNo=0)=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, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- float * AllocFloatS3D (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- int * AllocIntS3D (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- double * AllocDoubleV3D (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- float * AllocFloatV3D (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- int * AllocIntV3D (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- double * AllocDoubleV3DEx (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- float * AllocFloatV3DEx (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- int * AllocIntV3DEx (int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- double * AllocDoubleS4D (int nmax, int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- float * AllocFloatS4D (int nmax, int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- int * AllocIntS4D (int nmax, int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- double * AllocDoubleS4DEx (int nmax, int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- float * AllocFloatS4DEx (int nmax, int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- int * AllocIntS4DEx (int nmax, int vc, bool padding=false, int *pad_size=NULL, int procGrpNo=0)
- void flush (std::ostream &out, int procGrpNo=0)
- void flush (FILE *fp, int procGrpNo=0)
- template<class T >
  CPM_INLINE void InitArray (T *array, size_t size)
- template<class T >
  CPM_INLINE void CopyArray (T *source, T *dist, size_t size)

Static Public メソッド

- 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)
- static int GetPaddingSize1D (const int size, const int vc)
- static void GetPaddingSize (CPM_ARRAY_SHAPE atype, const int *size, const int vc, int *pad_size, int nmax=0)

Protected メソッド

- cpm_BaseParaManager ()
- virtual ~cpm_BaseParaManager ()
- virtual double * AllocDouble (int nmax, int sz[3], int vc, int procGrpNo)=0
- virtual float * AllocFloat (int nmax, int sz[3], int vc, int procGrpNo)=0
- virtual int * AllocInt (int nmax, int sz[3], int vc, int procGrpNo)=0

Protected 変数

- int m_nRank
- int m_rankNo
- cpm_DomainType m_domainType
- std::vector< MPI_Comm > m_procGrpList
- cpm_ObjList< MPI_Request > m_reqList
- DefPointMap m_defPointMap

6.5.1 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_BaseParaManager.h の 37 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_BaseParaManager::cpm_BaseParaManager ( ) [protected]

コンストラクタ

cpm_BaseParaManager.cpp の 21 行で定義されています。

参照先 CPM_DOMAIN_UNKNOWN, m_defPointMap, m_domainType, m_nRank, m_procGrpList, と m_rankNo.

6.5.2.2 cpm_BaseParaManager::~cpm_BaseParaManager ( ) [protected], [virtual]

デストラクタ

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

参照先 m_defPointMap, と m_procGrpList.

6.5.3 関数

6.5.3.1 void cpm_BaseParaManager::Abort (int *errorcode*)

Abort

- MPI_Abort のインターフェイス

引数

in	*errorcode*	MPI_Abort に渡すエラーコード

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

参照元 cpm_Abort_(), cpm_Abort_LMR_(), cpm_ParaManager::VoxelInit(), と cpm_ParaManagerLMR::VoxelInit_-LMR().

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

Allgather

- MPI_Allgather のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_inline.h の 205 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_(), と cpm_Allgather_LMR_().

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

Allgather

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 450 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_PROCGROU-P, CPM_SUCCESS, GetMPI_Comm(), と cpm_Base::IsCommNull().

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

Allgatherv

- MPI_Allgatherv のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_inline.h の 249 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Allgatherv_(), と cpm_Allgatherv_LMR_().

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

Allgatherv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 509 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().

6.5.3.6 virtual double* cpm_BaseParaManager::AllocDouble (int *nmax,* int *sz[3],* int *vc,* int *procGrpNo*) [protected], [pure virtual]

配列確保 (double)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_ParaManager, と cpm_ParaManagerLMRで実装されています。

参照元 AllocDoubleS4D(), と AllocDoubleS4DEx().

6.5.3.7 double * cpm_BaseParaManager::AllocDoubleS3D (int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 double(imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (3word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 98 行で定義されています。

参照先 AllocDoubleS4D().

6.5.3.8 double * cpm_BaseParaManager::AllocDoubleS4D (int *nmax*, int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 double(imax,jmax,kmax,nmax)

引数

in	*nmax*	成分数
in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 23 行で定義されています。

参照先 AllocDouble(), CPM_ARRAY_S4D, GetLocalArraySize(), と GetPaddingSize().

参照元 AllocDoubleS3D(), と AllocDoubleV3D().

6.5.3.9 double * cpm_BaseParaManager::AllocDoubleS4DEx (int *nmax*, int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 double(nmax,imax,jmax,kmax)

引数

in	*nmax*	成分数
in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 170 行で定義されています。

参照先 AllocDouble(), CPM_ARRAY_S4DEX, GetLocalArraySize(), と GetPaddingSize().

参照元 AllocDoubleV3DEx().

6.5.3.10 double * cpm_BaseParaManager::AllocDoubleV3D (int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 double(imax,jmax,kmax,3)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocDoubleS4D().

6.5.3.11 double * cpm_BaseParaManager::AllocDoubleV3DEx (int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 double(3,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocDoubleS4DEx().

6.5.3.12 virtual float* cpm_BaseParaManager::AllocFloat (int *nmax,* int *sz[3],* int *vc,* int *procGrpNo*) [protected], [pure virtual]

配列確保 (float)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_ParaManager, と cpm_ParaManagerLMRで実装されています。

参照元 AllocFloatS4D(), と AllocFloatS4DEx().

6.5.3.13 float * cpm_BaseParaManager::AllocFloatS3D (int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 float(imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (3word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 114 行で定義されています。

参照先 AllocFloatS4D().

6.5.3.14 float * cpm_BaseParaManager::AllocFloatS4D (int *nmax*, int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 float(imax,jmax,kmax,nmax)

引数

in	*nmax*	成分数
in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocFloat(), CPM_ARRAY_S4D, GetLocalArraySize(), と GetPaddingSize().

参照元 AllocFloatS3D(), と AllocFloatV3D().

6.5.3.15 float ∗ cpm_BaseParaManager::AllocFloatS4DEx (int *nmax*, int *vc*, bool *padding* = false, int ∗ *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 float(nmax,imax,jmax,kmax)

引数

in	*nmax*	成分数
in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 195 行で定義されています。

参照先 AllocFloat(), CPM_ARRAY_S4DEX, GetLocalArraySize(), と GetPaddingSize().

参照元 AllocFloatV3DEx().

6.5.3.16 float ∗ cpm_BaseParaManager::AllocFloatV3D (int *vc*, bool *padding* = false, int ∗ *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 float(imax,jmax,kmax,3)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 154 行で定義されています。

参照先 AllocFloatS4D().

6.5.3.17 float ∗ cpm_BaseParaManager::AllocFloatV3DEx (int *vc*, bool *padding* = false, int ∗ *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 float(3,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 253 行で定義されています。

参照先 AllocFloatS4DEx().

6.5.3.18 virtual int* cpm_BaseParaManager::AllocInt (int *nmax,* int *sz[3],* int *vc,* int *procGrpNo*) [protected], [pure virtual]

配列確保 (int)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_ParaManager, と cpm_ParaManagerLMRで実装されています。

参照元 AllocIntS4D(), と AllocIntS4DEx().

6.5.3.19 int * cpm_BaseParaManager::AllocIntS3D (int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 int(imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (3word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 130 行で定義されています。

参照先 AllocIntS4D().

6.5.3.20 int * cpm_BaseParaManager::AllocIntS4D (int *nmax*, int *vc*, bool *padding* = false, int * *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 int(imax,jmax,kmax,nmax)

引数

in	*nmax*	成分数
in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocInt(), CPM_ARRAY_S4D, GetLocalArraySize(), と GetPaddingSize().

参照元 AllocIntS3D(), と AllocIntV3D().

6.5.3.21 int ∗ cpm_BaseParaManager::AllocIntS4DEx (int *nmax*, int *vc*, bool *padding* = false, int ∗ *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 int(nmax,imax,jmax,kmax)

引数

in	*nmax*	成分数
in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 220 行で定義されています。

参照先 AllocInt(), CPM_ARRAY_S4DEX, GetLocalArraySize(), と GetPaddingSize().

参照元 AllocIntV3DEx().

6.5.3.22 int ∗ cpm_BaseParaManager::AllocIntV3D (int *vc*, bool *padding* = false, int ∗ *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 int(imax,jmax,kmax,3)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_BaseParaManager_Alloc.cpp の 162 行で定義されています。

参照先 AllocIntS4D().

6.5.3.23 int ∗ cpm_BaseParaManager::AllocIntV3DEx (int *vc*, bool *padding* = false, int ∗ *pad_size* = NULL, int *procGrpNo* = 0)

配列確保 int(3,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*padding*	パディングフラグ (true:する、false:しない)
out	*pad_size*	各次元のパディング数 (4word, NULL のとき格納しない)
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocIntS4DEx().

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

Allreduce

- MPI_Allreduce のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_inline.h の 166 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_(), cpm_Allreduce_LMR_(), cpm_ParaManager::GetBndIndexExtGc(), と cpm_ParaManagerLMR::GetNumLeaf().

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

Allreduce

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 394 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().

6.5.3.26 cpm_ErrorCode cpm_BaseParaManager::Barrier (int *procGrpNo* = 0)

Barrier

- MPI_Barrier のインターフェイス

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 174 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_(), cpm_Barrier_LMR_(), Initialize(), と cpm_ParaManagerLMR::SetBndCommBuffer().

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

Bcast

- MPI_Bcast のインターフェイス

引数

in,out	*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	*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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_inline.h の 81 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_(), と cpm_Bcast_LMR_().

6.5.3.28 cpm_ErrorCode cpm_BaseParaManager::Bcast (MPI_Datatype *dtype*, void * *buf*, int *count*, int *root*, int *procGrpNo* = 0)

Bcast

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

引数

in	*dtype*	送信バッファのMPI_Datatype
in,out	*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	*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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 252 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().

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

cpm_BaseParaManager_inline.h の 35 行で定義されています。

6.5.3.30 template<class T > void cpm_BaseParaManager::CopyArray (T * *source*, T * *dist*, size_t *size*)

配列のコピー

引数

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

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

cpm_Irecv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 639 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_(), と cpm_Irecv_LMR_().

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 602 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_(), と cpm_Isend_LMR_().

6.5.3.33 cpm_ErrorCode cpm_BaseParaManager::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_BaseParaManager_MPI.cpp の 538 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_(), と cpm_Wait_LMR_().

6.5.3.34 cpm_ErrorCode cpm_BaseParaManager::cpm_Waitall (int *count*, int *reqNoList[]*)

cpm_Waitall

- MPI_Waitall のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 561 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_(), と cpm_Waitall_LMR_().

6.5.3.35 int cpm_BaseParaManager::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	*nproc*	使用するプロセスの数
in	*proclist*	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	*parentProcGrpNo*	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_BaseParaManager.cpp の 158 行で定義されています。

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

6.5.3.36 virtual const cpm_VoxelInfo* cpm_BaseParaManager::FindVoxelInfo (int *procGrpNo* = 0) [pure virtual]

VOXEL 空間マップを検索

(LMR のときは自プロセス内先頭リーフのVoxelInfo を返す)

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

VOXEL 空間情報ポインタ

cpm_ParaManager, と cpm_ParaManagerLMRで実装されています。

参照元 GetGlobalNodeSize(), GetGlobalOrigin(), GetGlobalRegion(), GetGlobalVoxelSize(), GetLocalNodeSize(), と GetLocalVoxelSize().

6.5.3.37 void cpm_BaseParaManager::flush (std::ostream & *out*, int *procGrpNo* = 0)

flush

6.5.3.38 void cpm_BaseParaManager::flush (FILE * *fp,* int *procGrpNo* = 0)

flush

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

Gather

- MPI_Gather のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_inline.h の 183 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_(), と cpm_Gather_LMR_().

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 421 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().

6.5.3.41 template<class Ts , class Tr > CPM_INLINE cpm_ErrorCode cpm_BaseParaManager::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	*sendbuf*	送信データ
in	*sendcnt*	送信データのサイズ
out	*recvbuf*	受信データ
in	*recvcnts*	各ランクからの受信データサイズ
in	*displs*	各ランクからの受信データ配置位置
in	*root*	受信するランク番号 (procGrpNo 内でのランク番号)
in	*procGrpNo*	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_inline.h の 227 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_(), と cpm_Gatherv_LMR_().

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

Gatherv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 479 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().

6.5.3.43 virtual size_t cpm_BaseParaManager::GetBndCommBufferSize (int *procGrpNo* = 0) [pure virtual]

袖通信バッファサイズの取得

- 袖通信バッファとして確保されている配列サイズ (byte) を返す

引数

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

戻り値

バッファサイズ (byte)

cpm_ParaManager, と cpm_ParaManagerLMRで実装されています。

6.5.3.44 cpm_DefPointType cpm_BaseParaManager::GetDefPointType (int *procGrpNo* = 0)

定義点タイプを取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

定義点情報 (enum)

cpm_BaseParaManager.cpp の 208 行で定義されています。

参照先 CPM_DEFPOINTTYPE_UNKNOWN, と m_defPointMap.

参照元 cpm_GetDefPointType_(), cpm_GetDefPointType_LMR_(), cpm_ParaManagerLMR::GetArrayHeadIndex(), cpm_ParaManager::GetArrayHeadIndex(), cpm_ParaManagerLMR::GetArrayTailIndex(), cpm_ParaManager::GetArrayTailIndex(), GetGlobalArraySize(), GetLocalArraySize(), cpm_ParaManager::packX(), cpm_ParaManager::packXEx(), cpm_ParaManager::packY(), cpm_ParaManager::packYEx(), cpm_ParaManager::packZ(), と cpm_ParaManager::packZEx().

6.5.3.45 cpm_DomainType cpm_BaseParaManager::GetDomainType ( )

領域分割タイプを取得

戻り値

領域分割タイプ

cpm_BaseParaManager.cpp の 200 行で定義されています。

参照先 m_domainType.

6.5.3.46 const int * cpm_BaseParaManager::GetGlobalArraySize (int *procGrpNo* = 0)

全体ボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数を取得 (LMR のときは最大レベルでの数を返す)

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

全体ボクセル数または頂点数の整数配列ポインタ (3word)

cpm_BaseParaManager.cpp の 243 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, GetDefPointType(), GetGlobalNodeSize(), と GetGlobalVoxelSize().

参照元 cpm_GetGlobalArraySize_(), cpm_GetGlobalArraySize_LMR_(), と cpm_ParaManager::GetBndIndexExtGc().

6.5.3.47 const int * cpm_BaseParaManager::GetGlobalNodeSize (int *procGrpNo* = 0)

全体頂点数を取得

(LMR のときは最大レベルでの全体頂点数を返す)

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

全体頂点数の整数配列ポインタ (3word)

cpm_BaseParaManager.cpp の 230 行で定義されています。

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

参照元 cpm_GetGlobalNodeSize_(), cpm_GetGlobalNodeSize_LMR_(), と GetGlobalArraySize().

6.5.3.48 const double * cpm_BaseParaManager::GetGlobalOrigin (int *procGrpNo* = 0)

全体空間の原点を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

全体空間の原点実数配列のポインタ

cpm_BaseParaManager.cpp の 262 行で定義されています。

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

参照元 cpm_GetGlobalOrigin_(), と cpm_GetGlobalOrigin_LMR_().

6.5.3.49 const double * cpm_BaseParaManager::GetGlobalRegion (int *procGrpNo* = 0)

全体空間サイズを取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

全体空間サイズ実数配列のポインタ

cpm_BaseParaManager.cpp の 274 行で定義されています。

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

参照元 cpm_GetGlobalRegion_(), と cpm_GetGlobalRegion_LMR_().

6.5.3.50 const int * cpm_BaseParaManager::GetGlobalVoxelSize (int *procGrpNo* = 0)

全体ボクセル数を取得

(LMR のときは最大レベルでの全体ボクセル数を返す)

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

全体ボクセル数の整数配列ポインタ (3word)

cpm_BaseParaManager.cpp の 218 行で定義されています。

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

参照元 cpm_GetGlobalVoxelSize_(), cpm_GetGlobalVoxelSize_LMR_(), と GetGlobalArraySize().

6.5.3.51 std::string cpm_BaseParaManager::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_BaseParaManager_MPI.cpp の 138 行で定義されています。

6.5.3.52 const int * cpm_BaseParaManager::GetLocalArraySize (int *procGrpNo* = 0)

自ランクのボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

ローカル頂点数の整数配列ポインタ (3word)

cpm_BaseParaManager.cpp の 311 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, GetDefPointType(), GetLocalNodeSize(), と GetLocalVoxelSize().

参照元 AllocDoubleS4D(), AllocDoubleS4DEx(), AllocFloatS4D(), AllocFloatS4DEx(), AllocIntS4D(), AllocIntS4DEx(), cpm_GetLocalArraySize_(), cpm_GetLocalArraySize_LMR_(), cpm_ParaManager::GetBndIndexExtGc(), cpm_ParaManager::Global2LocalIndex(), と cpm_ParaManager::SetBndCommBuffer().

6.5.3.53 const int * cpm_BaseParaManager::GetLocalNodeSize (int *procGrpNo* = 0)

自ランクの頂点数を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

ローカル頂点数の整数配列ポインタ (3word)

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

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

参照元 cpm_GetLocalNodeSize_(), cpm_GetLocalNodeSize_LMR_(), と GetLocalArraySize().

6.5.3.54 const int * cpm_BaseParaManager::GetLocalVoxelSize (int *procGrpNo* = 0)

自ランクのボクセル数を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

ローカルボクセル数の整数配列ポインタ (3word)

cpm_BaseParaManager.cpp の 286 行で定義されています。

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

参照元 cpm_GetLocalVoxelSize_(), cpm_GetLocalVoxelSize_LMR_(), GetLocalArraySize(), と cpm_ParaManager-LMR::SetBndCommBuffer().

6.5.3.55 MPI_Comm cpm_BaseParaManager::GetMPI_Comm (int *procGrpNo* = 0)

MPI コミュニケータの取得

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

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

MPI コミュニケータ

cpm_BaseParaManager_MPI.cpp の 149 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(), cpm_ParaManager::VoxelInit(), と cpm_ParaManagerLMR::VoxelInit_LMR().

6.5.3.56 template<class T > CPM_INLINE MPI_Datatype cpm_BaseParaManager::GetMPI_Datatype (T * *ptr*) [static]

MPI_Datatype を取得

引数

in	*ptr*	取得したいデータのポインタ

戻り値

MPI_Datatype

cpm_BaseParaManager_inline.h の 45 行で定義されています。

参照元 Allgather(), Allgatherv(), Allreduce(), Bcast(), cpm_Allgather_(), cpm_Allgather_LMR_(), cpm_Allgatherv_(), cpm_Allgatherv_LMR_(), cpm_Allreduce_(), cpm_Allreduce_LMR_(), cpm_Bcast_(), cpm_Bcast_LMR_(), cpm_BndCommS3D_(), cpm_BndCommS3D_LMR_(), cpm_ParaManager::cpm_BndCommS3D_nowait(), cpm_BndCommS4D_(), cpm_BndCommS4D_LMR_(), cpm_ParaManager::cpm_BndCommS4D_nowait(), cpm_BndCommS4DEx_(), cpm_BndCommS4DEx_LMR_(), cpm_ParaManager::cpm_BndCommS4DEx_nowait(), cpm_BndCommV3D_(), cpm_BndCommV3D_LMR_(), cpm_ParaManager::cpm_BndCommV3D_nowait(), cpm_BndCommV3DEx_(), cpm_BndCommV3DEx_LMR_(), cpm_ParaManager::cpm_BndCommV3DEx_nowait(), cpm_Gather_(), cpm_Gather_LMR_(), cpm_Gatherv_(), cpm_Gatherv_LMR_(), cpm_Irecv(), cpm_Isend(), cpm_PeriodicCommS3D_(), cpm_PeriodicCommS3D_LMR_(), cpm_PeriodicCommS4D_(), cpm_PeriodicCommS4D_LMR_(), cpm_PeriodicCommS4DEx_(), cpm_PeriodicCommS4DEx_LMR_(), cpm_PeriodicCommV3D_(), cpm_PeriodicCommV3D_LMR_(), cpm_PeriodicCommV3DEx_(), cpm_PeriodicCommV3DEx_LMR_(), cpm_Recv_(), cpm_Recv_LMR_(), cpm_Send_(), cpm_Send_LMR_(), 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(), Gather(), Gatherv(), Irecv(), Isend(), Recv(), と Send().

6.5.3.57 MPI_Datatype cpm_BaseParaManager::GetMPI_Datatype (int *datatype*) [static]

MPI_Datatype を取得

- Fortran データタイプからMPI_Datatype を取得

引数

in	*datatype*	取得したいデータのポインタ

戻り値

MPI_Datatype

cpm_BaseParaManager_MPI.cpp の 27 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::RealIsDouble().

6.5.3.58 MPI_Op cpm_BaseParaManager::GetMPI_Op (int *op*) [static]

MPI_Op を取得

- Fortran オペレータタイプからMPI_Op を取得

引数

in	*op*	取得したいデータのポインタ

戻り値

MPI_Op

cpm_BaseParaManager_MPI.cpp の 61 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_(), と cpm_Allreduce_LMR_().

6.5.3.59 int cpm_BaseParaManager::GetMyRankID (int *procGrpNo* = 0)

ランク番号の取得

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

  のいずれか

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

ランク番号

cpm_BaseParaManager_MPI.cpp の 82 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_(), と cpm_GetMyRankID_LMR_().

6.5.3.60 int cpm_BaseParaManager::GetNumRank (int *procGrpNo* = 0)

ランク数の取得

- プロセスグループのランク数を取得する

引数

in	*procGrpNo*	プロセスグループ番号 (省略時 0)

戻り値

ランク数

cpm_BaseParaManager_MPI.cpp の 110 行で定義されています。

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

参照元 cpm_GetNumRank_(), cpm_GetNumRank_LMR_(), cpm_ParaManager::VoxelInit(), と cpm_ParaManager::VoxelInit_Subdomain().

6.5.3.61 void cpm_BaseParaManager::GetPaddingSize (CPM_ARRAY_SHAPE *atype,* const int * *size,* const int *vc,* int * *pad_size,* int *nmax* = 0) [static]

パディングサイズ取得処理 (静的関数)

引数

in	*atype*	配列形状タイプ
in	*size*	配列サイズ{imax,jmax,kmax}
in	*vc*	仮想セル数
out	*pad_size*	パディングサイズ (S3D のとき 3word、V3D,S4D のとき 4word{px,py,pz,pn}、V3DEx,S4DEx のとき 4word{pn,px,py,pz})
in	*nmax*	成分数 (S4D,S4DEx のとき必須)

cpm_BaseParaManager.cpp の 359 行で定義されています。

参照先 CPM_ARRAY_S3D, CPM_ARRAY_S4D, CPM_ARRAY_S4DEX, CPM_ARRAY_V3D, CPM_ARRAY_V3DEX, と GetPaddingSize1D().

参照元 AllocDoubleS4D(), AllocDoubleS4DEx(), AllocFloatS4D(), AllocFloatS4DEx(), AllocIntS4D(), AllocIntS4DEx(), cpm_ParaManager::BndCommS3D(), cpm_ParaManager::BndCommS3D_nowait(), cpm_ParaManager::BndCommS4D(), cpm_ParaManager::BndCommS4D_nowait(), cpm_ParaManager::BndCommS4DEx(), cpm_ParaManager::BndCommS4DEx_nowait(), cpm_ParaManager::BndCommV3D(), cpm_ParaManager::BndCommV3D_nowait(), cpm_ParaManager::BndCommV3DEx(), cpm_ParaManager::BndCommV3DEx_nowait(), cpm_ParaManager::GetBndIndexExtGc(), cpm_ParaManager::PeriodicCommS3D(), cpm_ParaManager::PeriodicCommS4D(), cpm_ParaManager::PeriodicCommS4DEx(), cpm_ParaManager::PeriodicCommV3D(), cpm_ParaManager::PeriodicCommV3DEx(), cpm_ParaManager::wait_BndCommS3D(), cpm_ParaManager::wait_BndCommS4D(), cpm_ParaManager::wait_BndCommS4DEx(), cpm_ParaManager::wait_BndCommV3D(), と cpm_ParaManager::wait_BndCommV3DEx().

6.5.3.62 int cpm_BaseParaManager::GetPaddingSize1D (const int *size,* const int *vc*) [static]

パディングサイズ取得処理 (静的関数)

引数

in	*size*	配列サイズ
in	*vc*	仮想セル数

戻り値

パディングサイズ

cpm_BaseParaManager.cpp の 344 行で定義されています。

参照元 GetPaddingSize().

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

cpm_BaseParaManager_inline.h の 25 行で定義されています。

6.5.3.64 template<class T > void cpm_BaseParaManager::InitArray (T * *array*, size_t *size*)

配列の初期化処理

引数

out	*array*	初期化する配列のポインタ
in	*size*	配列サイズ

6.5.3.65 cpm_ErrorCode cpm_BaseParaManager::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_BaseParaManager.cpp の 60 行で定義されています。

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

参照元 cpm_ParaManagerLMR::get_instance(), cpm_ParaManager::get_instance(), と Initialize().

6.5.3.66 cpm_ErrorCode cpm_BaseParaManager::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_BaseParaManager.cpp の 110 行で定義されています。

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

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

Irecv

- MPI_Irecv のインターフェイス

引数

out	*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_BaseParaManager_inline.h の 149 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Irecv(), cpm_ParaManagerLMR::recv_LMR(), と cpm_ParaManager::sendrecv().

6.5.3.68 cpm_ErrorCode cpm_BaseParaManager::Irecv (MPI_Datatype *dtype,* void * *buf,* int *count,* int *source,* MPI_Request * *request,* int *procGrpNo* = 0)

Irecv

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

引数

in	*dtype*	送信データのMPI_Datatype
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_BaseParaManager_MPI.cpp の 365 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().

6.5.3.69 template<class T > CPM_INLINE cpm_ErrorCode cpm_BaseParaManager::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	*buf*	送信データ
in	*count*	送信データのサイズ
in	*dest*	送信先のランク番号 (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_BaseParaManager_inline.h の 132 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(), cpm_ParaManagerLMR::send_LMR(), cpm_ParaManagerLMR::send_LMR_Ex(), と cpm_ParaManager::sendrecv().

6.5.3.70 cpm_ErrorCode cpm_BaseParaManager::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	*dtype*	送信データのMPI_Datatype
in	*buf*	送信データ
in	*count*	送信データのサイズ
in	*dest*	送信先のランク番号 (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_BaseParaManager_MPI.cpp の 336 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().

6.5.3.71 bool cpm_BaseParaManager::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

true	並列実行
false	逐次実行

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

参照先 m_nRank.

参照元 cpm_IsParallel_(), cpm_IsParallel_LMR_(), と Initialize().

6.5.3.72 bool cpm_BaseParaManager::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

true	並列実行
false	逐次実行

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

参照先 m_nRank.

6.5.3.73 template<class T > CPM_INLINE cpm_ErrorCode cpm_BaseParaManager::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	*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)
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_BaseParaManager_inline.h の 115 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_(), と cpm_Recv_LMR_().

6.5.3.74 cpm_ErrorCode cpm_BaseParaManager::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	*dtype*	送信データのMPI_Datatype
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)
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_BaseParaManager_MPI.cpp の 307 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().

6.5.3.75 template<class T > CPM_INLINE cpm_ErrorCode cpm_BaseParaManager::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	*buf*	送信データ
in	*count*	送信データのサイズ
in	*dest*	送信先のランク番号 (procGrpNo 内でのランク番号)
in	*procGrpNo*	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_inline.h の 98 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_(), と cpm_Send_LMR_().

6.5.3.76 cpm_ErrorCode cpm_BaseParaManager::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	*dtype*	送信データのMPI_Datatype
in	*buf*	送信データ
in	*count*	送信データのサイズ
in	*dest*	送信先のランク番号 (procGrpNo 内でのランク番号)
in	*procGrpNo*	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 279 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().

6.5.3.77 virtual cpm_ErrorCode cpm_BaseParaManager::SetBndCommBuffer (size_t *maxVC*, size_t *maxN*, int *procGrpNo* = 0) [pure virtual]

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager, と cpm_ParaManagerLMRで実装されています。

6.5.3.78 cpm_ErrorCode cpm_BaseParaManager::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_BaseParaManager_MPI.cpp の 195 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.

参照元 cpm_ParaManagerLMR::recv_LMR_Ex_wait(), cpm_ParaManagerLMR::recv_LMR_wait(), と cpm_ParaManagerLMR::send_LMR_wait().

6.5.3.79 cpm_ErrorCode cpm_BaseParaManager::Waitall (int *count*, MPI_Request *requests[]*)

Waitall

- MPI_Waitall のインターフェイス

引数

in	*count*	リクエストの数
in	*requests*	リクエストハンドル配列

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManager_MPI.cpp の 219 行で定義されています。

参照先 CPM_ERROR_MPI_WAITALL, と CPM_SUCCESS.

参照元 cpm_ParaManager::BndCommS4D(), cpm_ParaManager::BndCommS4DEx(), cpm_ParaManager::PeriodicCommS4D(), cpm_ParaManager::PeriodicCommS4DEx(), cpm_ParaManager::wait_BndCommS4D(), と cpm_ParaManager::wait_BndCommS4DEx().

6.5.4 変数

6.5.4.1 DefPointMap cpm_BaseParaManager::m_defPointMap [protected]

プロセスグループ毎の定義点タイプ管理マップ

- プロセスグループ番号をキーとした定義点タイプマップ
- 自ランクが含まれるプロセスグループのみを管理する

cpm_BaseParaManager.h の 2150 行で定義されています。

参照元 cpm_BaseParaManager(), GetDefPointType(), cpm_ParaManager::NodeInit(), cpm_ParaManager::NodeInit_Subdomain(), cpm_ParaManager::VoxelInit(), cpm_ParaManagerLMR::VoxelInit_LMR(), と ~cpm_BaseParaManager().

6.5.4.2 cpm_DomainType cpm_BaseParaManager::m_domainType [protected]

領域分割タイプ

cpm_BaseParaManager.h の 2130 行で定義されています。

参照元 cpm_BaseParaManager(), cpm_ParaManager::cpm_ParaManager(), cpm_ParaManagerLMR::cpm_ParaManagerLMR(), と GetDomainType().

6.5.4.3 int cpm_BaseParaManager::m_nRank [protected]

プロセス並列数

cpm_BaseParaManager.h の 2124 行で定義されています。

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

6.5.4.4 std::vector<MPI_Comm> cpm_BaseParaManager::m_procGrpList [protected]

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_BaseParaManager.h の 2139 行で定義されています。

参照元 cpm_BaseParaManager(), CreateProcessGroup(), GetMPI_Comm(), GetMyRankID(), GetNumRank(), cpm_ParaManager::VoxelInit(), cpm_ParaManagerLMR::VoxelInit_LMR(), と ~cpm_BaseParaManager().

6.5.4.5 int cpm_BaseParaManager::m_rankNo [protected]

MPI_COMM_WORLD での自ランク番号

cpm_BaseParaManager.h の 2127 行で定義されています。

参照元 cpm_ParaManagerLMR::copy_LMR(), cpm_ParaManagerLMR::copy_LMR_Ex(), cpm_BaseParaManager(), Initialize(), cpm_ParaManagerLMR::packMX(), cpm_ParaManagerLMR::packMXEx(), cpm_ParaManagerLMR::packMY(), cpm_ParaManagerLMR::packMYEx(), cpm_ParaManagerLMR::packMZ(), cpm_ParaManagerLMR::packMZEx(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::packPXEx(), cpm_ParaManagerLMR::packPY(), cpm_ParaManagerLMR::packPYEx(), cpm_ParaManagerLMR::packPZ(), cpm_ParaManagerLMR::packPZEx(), cpm_ParaManagerLMR::recv_LMR(), cpm_ParaManagerLMR::recv_LMR_Ex_wait(), cpm_ParaManagerLMR::recv_LMR_wait(), cpm_ParaManagerLMR::send_LMR(), cpm_ParaManagerLMR::send_LMR_Ex(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.5.4.6 cpm_ObjList<MPI_Request> cpm_BaseParaManager::m_reqList [protected]

MPI_Request の管理マップ

- Fortran インターフェイス用

cpm_BaseParaManager.h の 2144 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_Irecv(), cpm_Isend(), 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_Waitall().

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_BaseParaManager.h
- cpm_BaseParaManager.cpp
- cpm_BaseParaManager_Alloc.cpp
- cpm_BaseParaManager_MPI.cpp
- cpm_BaseParaManager_inline.h

6.6 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
- void SetNodNum (int nod[3])
- const int * GetNodNum () 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**
- int m_nodNum [3]
  頂点数

Additional Inherited Members

6.6.1 説明

CPM の領域情報クラス

cpm_DomainInfo.h の 26 行で定義されています。

6.6.2 コンストラクタとデストラクタ

6.6.2.1 cpm_DomainInfo::cpm_DomainInfo ( )

コンストラクタ

cpm_DomainInfo.cpp の 21 行で定義されています。

参照先 clear().

6.6.2.2 cpm_DomainInfo::~cpm_DomainInfo ( ) [virtual]

デストラクタ

cpm_DomainInfo.cpp の 29 行で定義されています。

6.6.3 関数

6.6.3.1 cpm_ErrorCode cpm_DomainInfo::CheckData ( )

領域情報のチェック

- VoxelInit を実行する上で必要な情報がセットされているかをチェックする。

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_DomainInfo.cpp の 155 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().

6.6.3.2 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 36 行で定義されています。

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

参照元 cpm_GlobalDomainInfo::clear(), cpm_LocalDomainInfo::clear(), と cpm_DomainInfo().

6.6.3.3 const int * cpm_DomainInfo::GetNodNum ( ) const

頂点数の取得

戻り値

頂点数情報整数配列のポインタ

cpm_DomainInfo.cpp の 147 行で定義されています。

参照先 m_nodNum.

参照元 cpm_VoxelInfo::GetGlobalArraySize(), cpm_VoxelInfo::GetGlobalNodeSize(), cpm_VoxelInfo::GetLocalArraySize(), と cpm_VoxelInfo::GetLocalNodeSize().

6.6.3.4 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 64 行で定義されています。

参照先 m_origin.

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

6.6.3.5 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 82 行で定義されています。

参照先 m_pitch.

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

6.6.3.6 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 100 行で定義されています。

参照先 m_region.

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

6.6.3.7 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 124 行で定義されています。

参照先 m_voxNum.

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoLMR::debugPrint(), cpm_VoxelInfo::GetGlobalArraySize(), cpm_VoxelInfo::GetGlobalVoxelSize(), cpm_VoxelInfo::GetLocalArraySize(), cpm_VoxelInfo::GetLocalVoxelSize(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.6.3.8 void cpm_DomainInfo::SetNodNum (int *nod[3]*)

頂点数のセット

引数

in	*nod*	頂点数情報

cpm_DomainInfo.cpp の 133 行で定義されています。

参照先 m_nodNum, と m_voxNum.

6.6.3.9 void cpm_DomainInfo::SetOrigin (double *org[3]*)

原点のセット

引数

in	*org*	原点情報

cpm_DomainInfo.cpp の 54 行で定義されています。

参照先 m_origin.

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_TextParserDomain::ReadDomainInfo(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), cpm_VoxelInfoLMR::SetLocalDomainInfo(), cpm_ParaManager::VoxelInit(), と cpm_ParaManager::VoxelInit_Subdomain().

6.6.3.10 void cpm_DomainInfo::SetPitch (double *pch[3]*)

ピッチのセット

引数

in	*pch*	ピッチ情報

cpm_DomainInfo.cpp の 72 行で定義されています。

参照先 m_pitch.

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_TextParserDomain::ReadDomainInfo(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), cpm_VoxelInfoLMR::SetLocalDomainInfo(), cpm_ParaManager::VoxelInit(), と cpm_ParaManager::VoxelInit_Subdomain().

6.6.3.11 void cpm_DomainInfo::SetRegion (double *rgn[3]*)

空間サイズのセット

引数

in	*rgn*	空間サイズ情報

cpm_DomainInfo.cpp の 90 行で定義されています。

参照先 m_region.

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_TextParserDomain::ReadDomainInfo(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), cpm_VoxelInfoLMR::SetLocalDomainInfo(), cpm_ParaManager::VoxelInit(), と cpm_ParaManager::VoxelInit_Subdomain().

6.6.3.12 void cpm_DomainInfo::SetVoxNum (int *vox[3]*)

VOXEL 数のセット

引数

in	*vox*	VOXEL 数情報

cpm_DomainInfo.cpp の 108 行で定義されています。

参照先 m_nodNum, と m_voxNum.

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_TextParserDomain::ReadDomainInfo(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), cpm_VoxelInfoLMR::SetLocalDomainInfo(), cpm_ParaManager::VoxelInit(), と cpm_ParaManager::VoxelInit_Subdomain().

6.6.4 変数

6.6.4.1 int cpm_DomainInfo::m_nodNum[3] [private]

頂点数

cpm_DomainInfo.h の 116 行で定義されています。

参照元 clear(), GetNodNum(), SetNodNum(), と SetVoxNum().

6.6.4.2 double cpm_DomainInfo::m_origin[3] [private]

原点

cpm_DomainInfo.h の 110 行で定義されています。

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

6.6.4.3 double cpm_DomainInfo::m_pitch[3] [private]

ピッチ

cpm_DomainInfo.h の 112 行で定義されています。

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

6.6.4.4 double cpm_DomainInfo::m_region[3] [private]

空間サイズ

cpm_DomainInfo.h の 111 行で定義されています。

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

6.6.4.5 int cpm_DomainInfo::m_voxNum[3] [private]

VOXEL 数

cpm_DomainInfo.h の 113 行で定義されています。

参照元 CheckData(), clear(), GetVoxNum(), SetNodNum(), 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

6.7 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

Additional Inherited Members

6.7.1 説明

CPM の全体領域情報クラス

cpm_DomainInfo.h の 183 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_GlobalDomainInfo::cpm_GlobalDomainInfo ( )

コンストラクタ

cpm_DomainInfo.cpp の 243 行で定義されています。

参照先 clear().

6.7.2.2 cpm_GlobalDomainInfo::~cpm_GlobalDomainInfo ( ) [virtual]

デストラクタ

cpm_DomainInfo.cpp の 251 行で定義されています。

6.7.3 関数

6.7.3.1 bool cpm_GlobalDomainInfo::AddSubdomain (cpm_ActiveSubdomainInfo *subDomain*)

活性サブドメイン情報の追加

引数

in	*subDomain*	追加する活性サブドメイン情報

戻り値

true	追加した
false	追加に失敗 (同じ領域分割位置で追加済み)

cpm_DomainInfo.cpp の 302 行で定義されています。

参照先 IsExistSubdomain(), と m_subDomainInfo.

参照元 CheckData(), と cpm_ParaManager::VoxelInit_Subdomain().

6.7.3.2 cpm_ErrorCode cpm_GlobalDomainInfo::CheckData (int *nRank*)

領域情報のチェック

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

引数

in	*nRank*	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 347 行で定義されています。

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

参照元 cpm_ParaManager::VoxelInit().

6.7.3.3 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 258 行で定義されています。

参照先 cpm_DomainInfo::clear(), m_divNum, と m_subDomainInfo.

参照元 cpm_GlobalDomainInfo().

6.7.3.4 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 281 行で定義されています。

参照先 m_divNum.

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

6.7.3.5 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

戻り値

活性サブドメイン情報配列サイズ

cpm_DomainInfo.cpp の 331 行で定義されています。

参照先 m_subDomainInfo.

6.7.3.6 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

in	*idx*	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 339 行で定義されています。

参照先 GetSubdomainNum(), と m_subDomainInfo.

参照元 cpm_VoxelInfoCART::CreateRankMap().

6.7.3.7 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 315 行で定義されています。

参照先 m_divNum, と m_subDomainInfo.

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

6.7.3.8 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

in	*subDomain*	チェックする活性サブドメイン情報

戻り値

true	存在する
false	存在しない

cpm_DomainInfo.cpp の 289 行で定義されています。

参照先 m_subDomainInfo.

参照元 AddSubdomain().

6.7.3.9 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

in	*ident*	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 394 行で定義されています。

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

6.7.3.10 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 421 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::VoxelInit_Subdomain().

6.7.3.11 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 449 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.

6.7.3.12 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 271 行で定義されています。

参照先 m_divNum.

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

6.7.4 変数

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

領域分割数

cpm_DomainInfo.h の 293 行で定義されています。

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

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

活性サブドメイン情報

cpm_DomainInfo.h の 294 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

6.8 クラス cpm_LeafCommInfo

`#include <cpm_LeafCommInfo.h>`

cpm_LeafCommInfo に対する継承グラフ

cpm_LeafCommInfo のコラボレーション図

構成

- struct stCommInfo

Public メソッド

- cpm_LeafCommInfo (int distRankID)
- virtual ~cpm_LeafCommInfo ()
- void AddCommInfo (stCommInfo *commInfo)
- void Sort (int type)
- bool SetBndCommBuffer (int myRankNo, size_t sz[2], size_t maxVC, size_t maxN)
- void * GetBndCommSendBufferPtr ()
- void * GetBndCommRecvBufferPtr ()
- size_t GetBndCommBufferSize ()
- stCommInfo * SearchDistCommInfo (stCommInfo *commInfo)

Public 変数

- int m_iDistRankNo

  通信相手のランク番号
- std::vector< stCommInfo * > m_vecCommInfo

  経路情報
- MPI_Request m_reqSend
- MPI_Request m_reqRecv

Protected メソッド

- void Qsort (int type, std::vector< stCommInfo * > &vecCommInfo, int startIndex, int endIndex)

Protected 変数

- REAL_BUF_TYPE * m_pCommSendBuf

  送信バッファ
- REAL_BUF_TYPE * m_pCommRecvBuf

  受信バッファ
- size_t m_CommSendBufSize

  送信バッファサイズ *(WORD* **数***)*
- size_t m_CommRecvBufSize

  受信バッファサイズ *(WORD* **数***)*

Additional Inherited Members

6.8.1 説明

LMR の袖通信情報管理クラス

- 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_LeafCommInfo.h の 29 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_LeafCommInfo::cpm_LeafCommInfo (int *distRankID*)

コンストラクタ

cpm_LeafCommInfo.cpp の 22 行で定義されています。

参照先 m_iDistRankNo, m_pCommRecvBuf, m_pCommSendBuf, m_reqRecv, m_reqSend, と m_vecCommInfo.

6.8.2.2 cpm_LeafCommInfo::~cpm_LeafCommInfo ( ) [virtual]

デストラクタ

cpm_LeafCommInfo.cpp の 34 行で定義されています。

参照先 m_pCommRecvBuf, m_pCommSendBuf, と m_vecCommInfo.

6.8.3 関数

6.8.3.1 void cpm_LeafCommInfo::AddCommInfo (stCommInfo * *commInfo*)

CommInfo を追加

cpm_LeafCommInfo.cpp の 51 行で定義されています。

参照先 m_vecCommInfo.

参照元 cpm_ParaManagerLMR::SetBndCommBuffer().

6.8.3.2 size_t cpm_LeafCommInfo::GetBndCommBufferSize ( ) [inline]

袖通信バッファサイズの取得

戻り値

バッファサイズ (word)

cpm_LeafCommInfo.h の 172 行で定義されています。

参照先 m_CommRecvBufSize, と m_CommSendBufSize.

参照元 cpm_ParaManagerLMR::GetBndCommBufferSize().

6.8.3.3 void* cpm_LeafCommInfo::GetBndCommRecvBufferPtr ( ) [inline]

袖通信の受信バッファの取得

戻り値

バッファのポインタ

cpm_LeafCommInfo.h の 167 行で定義されています。

参照先 m_pCommRecvBuf.

参照元 cpm_ParaManagerLMR::recv_LMR(), cpm_ParaManagerLMR::recv_LMR_Ex_wait(), と cpm_ParaManagerLMR::recv_LMR_wait().

6.8.3.4 void* cpm_LeafCommInfo::GetBndCommSendBufferPtr ( ) [inline]

袖通信の送信バッファの取得

戻り値

バッファのポインタ

cpm_LeafCommInfo.h の 162 行で定義されています。

参照先 m_pCommSendBuf.

参照元 cpm_ParaManagerLMR::copy_LMR(), cpm_ParaManagerLMR::copy_LMR_Ex(), cpm_ParaManagerLMR::send_LMR(), と cpm_ParaManagerLMR::send_LMR_Ex().

6.8.3.5 void cpm_LeafCommInfo::Qsort (int *type*, std::vector< stCommInfo * > & *vecCommInfo*, int *startIndex*, int *endIndex*) [protected]

CommInfo リストのクイックソート

引数

in	*type*	ソートタイプ (0:自身のリーフ番号でソート, 1:相手のリーフ番号でソート)
in,out	*vecCommInfo*	ソート対象の配列
in	*startIndex*	開始インデクス

in	*endIndex*	終了インデクス

cpm_LeafCommInfo.cpp の 107 行で定義されています。

参照先 cpm_LeafCommInfo::stCommInfo::GetLeafID().

参照元 Sort().

6.8.3.6 cpm_LeafCommInfo::stCommInfo * cpm_LeafCommInfo::SearchDistCommInfo (cpm_LeafCommInfo::stCommInfo * *commInfo*)

対となる通信情報を検索

(OwnLeaf とDistLeaf が入れ替わりで bPeriodic が同じ通信情報)

引数

in	*commInfo*	検索したい元の通信情報

戻り値

対となる通信情報 (存在しない場合NULL)

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

参照先 cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::iDistLeafID, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, と m_vecCommInfo.

参照元 cpm_ParaManagerLMR::copy_LMR(), と cpm_ParaManagerLMR::copy_LMR_Ex().

6.8.3.7 bool cpm_LeafCommInfo::SetBndCommBuffer (int *myRankNo,* size_t *sz[2],* size_t *maxVC,* size_t *maxN*)

袖通信バッファのセット

引数

in	*myRankNo*	自身のランク番号
in	*sz_face*	1 リーフの格子数 (平面内 2 軸)
in	*maxVC*	送受信バッファの最大袖数
in	*maxN*	送受信バッファの最大成分数

cpm_LeafCommInfo.cpp の 137 行で定義されています。

参照先 m_CommRecvBufSize, m_CommSendBufSize, m_pCommRecvBuf, m_pCommSendBuf, m_vecCommInfo, と REAL_BUF_TYPE.

参照元 cpm_ParaManagerLMR::SetBndCommBuffer().

6.8.3.8 void cpm_LeafCommInfo::Sort (int *type*)

CommInfo リストのソート

引数

in	*type*	ソートタイプ (0:自身のリーフ番号でソート, 1:相手のリーフ番号でソート)

cpm_LeafCommInfo.cpp の 62 行で定義されています。

参照先 m_vecCommInfo, と Qsort().

参照元 cpm_ParaManagerLMR::SetBndCommBuffer().

6.8.4 変数

6.8.4.1 size_t cpm_LeafCommInfo::m_CommRecvBufSize [protected]

受信バッファサイズ (WORD 数)

cpm_LeafCommInfo.h の 226 行で定義されています。

参照元 GetBndCommBufferSize(), と SetBndCommBuffer().

6.8.4.2 size_t cpm_LeafCommInfo::m_CommSendBufSize [protected]

送信バッファサイズ (WORD 数)

cpm_LeafCommInfo.h の 223 行で定義されています。

参照元 GetBndCommBufferSize(), と SetBndCommBuffer().

6.8.4.3 int cpm_LeafCommInfo::m_iDistRankNo

通信相手のランク番号

cpm_LeafCommInfo.h の 203 行で定義されています。

参照元 cpm_LeafCommInfo().

6.8.4.4 REAL_BUF_TYPE* cpm_LeafCommInfo::m_pCommRecvBuf [protected]

受信バッファ

cpm_LeafCommInfo.h の 220 行で定義されています。

参照元 cpm_LeafCommInfo(), GetBndCommRecvBufferPtr(), SetBndCommBuffer(), と ~cpm_LeafCommInfo().

6.8.4.5 REAL_BUF_TYPE* cpm_LeafCommInfo::m_pCommSendBuf [protected]

送信バッファ

cpm_LeafCommInfo.h の 217 行で定義されています。

参照元 cpm_LeafCommInfo(), GetBndCommSendBufferPtr(), SetBndCommBuffer(), と ~cpm_LeafCommInfo().

6.8.4.6 MPI_Request cpm_LeafCommInfo::m_reqRecv

cpm_LeafCommInfo.h の 212 行で定義されています。

参照元 cpm_LeafCommInfo(), cpm_ParaManagerLMR::recv_LMR(), cpm_ParaManagerLMR::recv_LMR_Ex_wait(), cpm_ParaManagerLMR::recv_LMR_wait(), と cpm_ParaManagerLMR::send_LMR_wait().

6.8.4.7 MPI_Request cpm_LeafCommInfo::m_reqSend

cpm_LeafCommInfo.h の 209 行で定義されています。

参照元 cpm_LeafCommInfo(), cpm_ParaManagerLMR::send_LMR(), cpm_ParaManagerLMR::send_LMR_Ex(), と cpm_ParaManagerLMR::send_LMR_wait().

6.8.4.8 std::vector<stCommInfo*> cpm_LeafCommInfo::m_vecCommInfo

経路情報

cpm_LeafCommInfo.h の 206 行で定義されています。

参照元 AddCommInfo(), cpm_ParaManagerLMR::copy_LMR(), cpm_ParaManagerLMR::copy_LMR_Ex(), cpm_LeafCommInfo(), cpm_ParaManagerLMR::recv_LMR(), cpm_ParaManagerLMR::recv_LMR_Ex_wait(), cpm_ParaManagerLMR::recv_LMR_wait(), SearchDistCommInfo(), cpm_ParaManagerLMR::send_LMR(), cpm_ParaManagerLMR::send_LMR_Ex(), SetBndCommBuffer(), cpm_ParaManagerLMR::SetBndCommBuffer(), Sort(), と ~cpm_LeafCommInfo().

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_LeafCommInfo.h
- cpm_LeafCommInfo.cpp

6.9 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 ()

Additional Inherited Members

6.9.1 説明

CPM のローカル領域情報クラス

cpm_DomainInfo.h の 300 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_LocalDomainInfo::cpm_LocalDomainInfo ( )

コンストラクタ

cpm_DomainInfo.cpp の 534 行で定義されています。

6.9.2.2 cpm_LocalDomainInfo::~cpm_LocalDomainInfo ( ) [virtual]

デストラクタ

cpm_DomainInfo.cpp の 541 行で定義されています。

6.9.3 関数

6.9.3.1 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 548 行で定義されています。

参照先 cpm_DomainInfo::clear(), と cpm_ActiveSubdomainInfo::clear().

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

6.10 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

Additional Inherited Members

6.10.1 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 32 行で定義されています。

6.10.2 型定義

6.10.2.1 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 46 行で定義されています。

6.10.2.2 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 42 行で定義されています。

6.10.3 コンストラクタとデストラクタ

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

コンストラクタ

cpm_ObjList.h の 59 行で定義されています。

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

デストラクタ

cpm_ObjList.h の 67 行で定義されています。

6.10.4 関数

6.10.4.1 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 93 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_BaseParaManager::cpm_Irecv(), と cpm_BaseParaManager::cpm_Isend().

6.10.4.2 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 83 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_BaseParaManager::cpm_Irecv(), と cpm_BaseParaManager::cpm_Isend().

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

オブジェクトの削除

引数

in	*key*	Add の戻り値である登録番号

戻り値

CPM 終了コード (0,CPM_SUCCESS=正常終了)

cpm_ObjList.h の 123 行で定義されています。

参照元 cpm_BaseParaManager::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_BaseParaManager::cpm_Waitall().

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

オブジェクトの取得

引数

in	*key*	Add の戻り値である登録番号

戻り値

オブジェクトのポインタ

cpm_ObjList.h の 142 行で定義されています。

参照元 cpm_BaseParaManager::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_BaseParaManager::cpm_Waitall(), と cpm_ObjList< MPI_Request >::Delete().

6.10.5 変数

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

cpm_ObjList.h の 47 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().

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

使用可能な登録番号

cpm_ObjList.h の 50 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().

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

cpm_ObjList.h の 43 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

6.11 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

- virtual cpm_ErrorCode VoxelInit (cpm_GlobalDomainInfo *domainInfo, size_t maxVC=1, size_t maxN=3, int procGrpNo=0)
- virtual cpm_ErrorCode VoxelInit (int div[3], int vox[3], double origin[3], double region[3], size_t maxVC=1, size_t maxN=3, cpm_DivPolicy divPolicy=DIV_COMM_SIZE, int procGrpNo=0)
- virtual cpm_ErrorCode VoxelInit (int vox[3], double origin[3], double region[3], size_t maxVC=1, size_t maxN=3, cpm_DivPolicy divPolicy=DIV_COMM_SIZE, int procGrpNo=0)
- virtual cpm_ErrorCode NodeInit (int div[3], int nod[3], double origin[3], double region[3], size_t maxVC=1, size_t maxN=3, cpm_DivPolicy divPolicy=DIV_COMM_SIZE, int procGrpNo=0)
- virtual cpm_ErrorCode NodeInit (int nod[3], double origin[3], double region[3], size_t maxVC=1, size_t maxN=3, cpm_DivPolicy divPolicy=DIV_COMM_SIZE, int procGrpNo=0)
- virtual cpm_ErrorCode VoxelInit_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)
- virtual cpm_ErrorCode VoxelInit_Subdomain (int vox[3], double origin[3], double region[3], std::string subDomainFile, size_t maxVC=1, size_t maxN=3, int procGrpNo=0)
- virtual cpm_ErrorCode NodeInit_Subdomain (int div[3], int nod[3], double origin[3], double region[3], std::string subDomainFile, size_t maxVC=1, size_t maxN=3, int procGrpNo=0)
- virtual cpm_ErrorCode NodeInit_Subdomain (int nod[3], double origin[3], double region[3], std::string subDomainFile, size_t maxVC=1, size_t maxN=3, int procGrpNo=0)
- virtual const cpm_VoxelInfo * FindVoxelInfo (int procGrpNo=0)
- const int * GetDivNum (int procGrpNo=0)
- const double * GetPitch (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 * GetNodeHeadIndex (int procGrpNo=0)
- const int * GetArrayHeadIndex (int procGrpNo=0)
- const int * GetVoxelTailIndex (int procGrpNo=0)
- const int * GetNodeTailIndex (int procGrpNo=0)
- const int * GetArrayTailIndex (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, CPM_PADDING padding=CPM_PADDING_OFF)
- 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 pad_size[3], int procGrpNo=0)
- bool Global2LocalIndex (int iG, int jG, int kG, int &iL, int &jL, int &kL, int procGrpNo=0)
- bool IsOuterBoundary (cpm_FaceFlag face, int procGrpNo=0)
- bool IsInnerBoundary (cpm_FaceFlag face, int procGrpNo=0)
- virtual cpm_ErrorCode SetBndCommBuffer (size_t maxVC, size_t maxN, int procGrpNo=0)
- virtual 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_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommS3D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)

- 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_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommV3D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommS4D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommS4D (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], int procGrpNo)
- cpm_ErrorCode BndCommS4D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], int procGrpNo)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommS3D_nowait (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommV3D_nowait (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int pad_size[4], int procGrpNo)
- 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[48], int pad_size[4], int procGrpNo)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode wait_BndCommS3D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode wait_BndCommV3D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int pad_size[4], int procGrpNo)
- 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[48], int pad_size[4], int procGrpNo)

- 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_PADDING padding=CPM_PADDING_OFF)
- 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, CPM_PADDING padding=CPM_PADDING_OFF)
- 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_PADDING padding=CPM_PADDING_OFF)
- 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, CPM_PADDING padding=CPM_PADDING_OFF)
- 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_PADDING padding=CPM_PADDING_OFF)
- 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, CPM_PADDING padding=CPM_PADDING_OFF)
- 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 pad_size[4], int procGrpNo)
- 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 pad_size[4], int procGrpNo)
- 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_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommV3DEx (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommS4DEx (MPI_Datatype dtype, void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommS4DEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], 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 pad_size[4], 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode BndCommV3DEx_nowait (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int pad_size[4], 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[48], int pad_size[4], 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- cpm_ErrorCode wait_BndCommV3DEx (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)

- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int procGrpNo=0, CPM_PADDING padding=CPM_PADDING_OFF)
- 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[48], int pad_size[4], 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[48], int pad_size[4], 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_PADDING padding=CPM_PADDING_OFF)
- 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, CPM_PADDING padding=CPM_PADDING_OFF)
- 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_PADDING padding=CPM_PADDING_OFF)
- 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, CPM_PADDING padding=CPM_PADDING_OFF)
- 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 pad_size[4], 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 pad_size[4], int procGrpNo=0)
- cpm_ErrorCode cpm_BndCommS3D_nowait (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_BndCommV3D_nowait (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_BndCommS4D_nowait (void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_wait_BndCommS3D (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_wait_BndCommV3D (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_wait_BndCommS4D (void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_BndCommV3DEx_nowait (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_BndCommS4DEx_nowait (void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_wait_BndCommV3DEx (void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_wait_BndCommS4DEx (void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int datatype, int *reqNo, int procGrpNo=0)
- 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 pad_size[4], int procGrpNo)
- 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 pad_size[4], int procGrpNo)

- template<class T >
  CPM_INLINE cpm_ErrorCode packX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode sendrecv (T *sendm, T *recvm, T *sendp, T *recvp, size_t nw, MPI_Request *req, int nIDsm, int nIDrm, int nIDsp, int nIDrp, int procGrpNo)
- 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 pad_size[4], int procGrpNo)
- 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 pad_size[4], int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)

Static Public メソッド

- static cpm_ParaManager * get_instance ()
- static cpm_ParaManager * get_instance (int &argc, char **&argv)

Protected 型

- typedef std::map< int,
  S_BNDCOMM_BUFFER * > BndCommInfoMap

Protected メソッド

- cpm_ParaManager ()
- virtual ~cpm_ParaManager ()
- virtual double * AllocDouble (int nmax, int sz[3], int vc, int procGrpNo)
- virtual float * AllocFloat (int nmax, int sz[3], int vc, int procGrpNo)
- virtual int * AllocInt (int nmax, int sz[3], int vc, int procGrpNo)
- cpm_ErrorCode DecideDivPattern_CommSize (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_ErrorCode DecideDivPattern_Cube (int divNum, int voxSize[3], int divPttn[3]) const
- long long CheckCube (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, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  cpm_ErrorCode unpackX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], 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, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  cpm_ErrorCode unpackY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  cpm_ErrorCode unpackZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  cpm_ErrorCode unpackXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  cpm_ErrorCode unpackYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *sendm, T *sendp, int nIDm, int nIDp, int procGrpNo)
- template<class T >
  cpm_ErrorCode unpackZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int pad_size[4], T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode sendrecv (T *sendm, T *recvm, T *sendp, T *recvp, size_t nw, MPI_Request *req, int nIDsm, int nIDrm, int nIDsp, int nIDrp, int procGrpNo=0)

Protected 変数

- VoxelInfoMap m_voxelInfoMap
- BndCommInfoMap m_bndCommInfoMap

フレンド

- class cpm_BaseParaManager

6.11.1 説明

カーテシアン用の並列管理クラス

- cpm_BaseParaManager クラスからの派生
- get_instance 関数の引数の domainType がCPM_DOMAIN_CARTESIAN のとき、 このクラスがインスタンスされる

cpm_ParaManager.h の 74 行で定義されています。

6.11.2 型定義

6.11.2.1 typedef std::map<int, S_BNDCOMM_BUFFER*> cpm_ParaManager::BndCommInfoMap [protected]

プロセスグループ毎の袖通信バッファ情報マップの定義

cpm_ParaManager.h の 1807 行で定義されています。

6.11.3 コンストラクタとデストラクタ

6.11.3.1 cpm_ParaManager::cpm_ParaManager ( ) [protected]

コンストラクタ

cpm_ParaManager.cpp の 67 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, m_bndCommInfoMap, cpm_BaseParaManager::m_domainType, と m_voxelInfoMap.

6.11.3.2 cpm_ParaManager::~cpm_ParaManager ( ) [protected], [virtual]

デストラクタ

cpm_ParaManager.cpp の 82 行で定義されています。

参照先 m_bndCommInfoMap, と m_voxelInfoMap.

6.11.4 関数

6.11.4.1 double * cpm_ParaManager::AllocDouble (int *nmax*, int *sz[3]*, int *vc*, int *procGrpNo*) [protected], [virtual]

配列確保 (double)

引数

in	*nmax*	成分数

in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_BaseParaManagerを実装しています。

cpm_ParaManager_Alloc.cpp の 23 行で定義されています。

6.11.4.2 float * cpm_ParaManager::AllocFloat (int *nmax*, int *sz[3]*, int *vc*, int *procGrpNo*) [protected], [virtual]

配列確保 (float)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_BaseParaManagerを実装しています。

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

6.11.4.3 int * cpm_ParaManager::AllocInt (int *nmax*, int *sz[3]*, int *vc*, int *procGrpNo*) [protected], [virtual]

配列確保 (int)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_BaseParaManagerを実装しています。

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

6.11.4.4 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

袖通信 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の袖通信を行う

引数

in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 45 行で定義されています。

参照先 BndCommS4D(), CPM_ARRAY_S3D, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_BndCommS3D_().

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

袖通信 (Scalar3D 版, MPI_Datatype 指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BndCommS4D(), CPM_ARRAY_S3D, と cpm_BaseParaManager::GetPaddingSize().

6.11.4.6 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[48]*, int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

非同期版袖通信 (Scalar3D 版)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 90 行で定義されています。

参照先 BndCommS4D_nowait(), CPM_ARRAY_S3D, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_BndCommS3D_nowait().

6.11.4.7 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.8 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う

引数

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 75 行で定義されています。

参照先 CPM_ARRAY_S4D, と cpm_BaseParaManager::GetPaddingSize().

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

6.11.4.9 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

袖通信 (Scalar4D 版, MPI_Datatype 指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BndCommS4D(), CPM_ARRAY_S4D, と cpm_BaseParaManager::GetPaddingSize().

6.11.4.10 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 *pad_size[4]*, int *procGrpNo*)

袖通信 (Scalar4D 版, パディングサイズ指定)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う

引数

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (i,j,k,n)
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 234 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(), cpm_BaseParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.11.4.11 cpm_ErrorCode cpm_ParaManager::BndCommS4D (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* int *pad_size[4],* int *procGrpNo*)

袖通信 (Scalar4D 版, MPI_Datatype 指定, パディングサイズ指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (i,j,k,n)
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 72 行で定義されています。

参照先 BndCommS4D(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.11.4.12 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 *pad_size[4],* int *procGrpNo*)

cpm_ParaManager_BndComm.h の 339 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.

6.11.4.13 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 122 行で定義されています。

参照先 CPM_ARRAY_S4D, と cpm_BaseParaManager::GetPaddingSize().

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

6.11.4.14 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

非同期版袖通信 (Scalar4D 版, MPI_Datatype 指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.15 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[48],* int *pad_size[4],* int *procGrpNo*)

非同期版袖通信 (Scalar4D 版, パディングサイズ指定)

- (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 (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (i,j,k,n)
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)

6.11.4.16 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[48],* int *pad_size[4],* int *procGrpNo*)

非同期版袖通信 (Scalar4D 版, MPI_Datatype 指定, パディングサイズ指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (i,j,k,n)
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)

6.11.4.17 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

in,out	*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	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 67 行で定義されています。

参照先 CPM_ARRAY_S4DEX, と cpm_BaseParaManager::GetPaddingSize().

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

6.11.4.18 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in,out	*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	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 402 行で定義されています。

参照先 BndCommS4DEx(), CPM_ARRAY_S4DEX, と cpm_BaseParaManager::GetPaddingSize().

6.11.4.19 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 *pad_size[4],* int *procGrpNo* = 0)

袖通信 (Scalar4DEx 版, パディングサイズ指定)

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

in,out	*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	*pad_size*	パディングサイズ (n,i,j,k)
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 178 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(), cpm_BaseParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.11.4.20 cpm_ErrorCode cpm_ParaManager::BndCommS4DEx (MPI_Datatype *dtype,* void * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *pad_size[4],* int *procGrpNo* = 0)

袖通信 (Scalar4DEx 版, MPI_Datatype 指定, パディングサイズ指定)

- (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	*dtype*	袖通信データのMPI_Datatype
in,out	*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	*pad_size*	パディングサイズ (n,i,j,k)
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 417 行で定義されています。

参照先 BndCommS4DEx(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.11.4.21 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 *pad_size[4],* int *procGrpNo*)

cpm_ParaManager_BndCommEx.h の 282 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.

6.11.4.22 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

非同期版袖通信 (Scalar4DEx 版)

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

引数

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

in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 98 行で定義されています。

参照先 CPM_ARRAY_S4DEX, と cpm_BaseParaManager::GetPaddingSize().

参照元 BndCommV3DEx_nowait(), と cpm_BndCommS4DEx_nowait().

6.11.4.23 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.24 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[48],* int *pad_size[4],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4DEx 版, パディングサイズ指定)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (n,i,j,k)
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)

6.11.4.25 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[48],* int *pad_size[4],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4DEx 版, MPI_Datatype 指定, パディングサイズ指定)

- (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	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (n,i,j,k)
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)

6.11.4.26 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

袖通信 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う

引数

in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 60 行で定義されています。

参照先 BndCommS4D(), CPM_ARRAY_V3D, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_BndCommV3D_().

6.11.4.27 cpm_ErrorCode cpm_ParaManager::BndCommV3D (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BndCommS4D(), CPM_ARRAY_V3D, と cpm_BaseParaManager::GetPaddingSize().

6.11.4.28 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 106 行で定義されています。

参照先 BndCommS4D_nowait(), CPM_ARRAY_V3D, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_BndCommV3D_nowait().

6.11.4.29 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[48]*, int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.30 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BndCommS4DEx(), CPM_ARRAY_V3DEX, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_BndCommV3DEx_().

6.11.4.31 cpm_ErrorCode cpm_ParaManager::BndCommV3DEx (MPI_Datatype *dtype*, void * *array*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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
in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 387 行で定義されています。

参照先 BndCommS4DEx(), CPM_ARRAY_V3DEX, と cpm_BaseParaManager::GetPaddingSize().

6.11.4.32 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[48]*, int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 82 行で定義されています。

参照先 BndCommS4DEx_nowait(), CPM_ARRAY_V3DEX, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_BndCommV3DEx_nowait().

6.11.4.33 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.34 unsigned long long cpm_ParaManager::CalcCommSize (unsigned long long *iDiv,* unsigned long long *jDiv,* unsigned long long *kDiv,* unsigned long long *voxSize[3]*) const [protected]

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

戻り値

袖通信点数

cpm_ParaManager.cpp の 527 行で定義されています。

参照元 DecideDivPattern_CommSize().

6.11.4.35 long long cpm_ParaManager::CheckCube (unsigned long long *iDiv,* unsigned long long *jDiv,* unsigned long long *kDiv,* unsigned long long *voxSize[3]*) const [protected]

I,J,K 分割を行った時のI,J,K ボクセル数の最大/最小の差を取得する

引数

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

戻り値

***0* 以上**	I,J,K ボクセル数の最大/最小の差
負値	領域分割不可のパターン

cpm_ParaManager.cpp の 632 行で定義されています。

参照元 DecideDivPattern_Cube().

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

cpm_BndCommS3D_nowait

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

引数

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

out	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3034 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_PADDING_OFF, CPM_SUCCESS, cpm_ObjList< T >::Create(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::m_reqList.

参照元 cpm_BndCommS3D_nowait_().

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

cpm_BndCommS4D_nowait

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3118 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(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::m_reqList.

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

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

cpm_BndCommS4DEx_nowait

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3332 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(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::m_reqList.

参照元 cpm_BndCommS4DEx_nowait_(), と cpm_BndCommV3DEx_nowait().

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

cpm_BndCommV3D_nowait

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3076 行で定義されています。

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

参照元 cpm_BndCommV3D_nowait_().

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

cpm_BndCommV3DEx_nowait

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
out	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3290 行で定義されています。

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

参照元 cpm_BndCommV3DEx_nowait_().

6.11.4.41 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)
in	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3156 行で定義されています。

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

参照元 cpm_wait_BndCommS3D_().

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

cpm_wait_BndCommS4D

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
in	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3248 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(), cpm_BaseParaManager::GetMPI_Datatype(), cpm_BaseParaManager::m_reqList, と wait_BndCommS4D().

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

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

cpm_wait_BndCommS4DEx

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
in	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3416 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(), cpm_BaseParaManager::GetMPI_Datatype(), cpm_BaseParaManager::m_reqList, と wait_BndCommS4DEx().

参照元 cpm_wait_BndCommS4DEx_(), と cpm_wait_BndCommV3DEx().

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

cpm_wait_BndCommV3D

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
in	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3202 行で定義されています。

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

参照元 cpm_wait_BndCommV3D_().

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

cpm_wait_BndCommV3DEx

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
in	*reqNo*	リクエスト番号配列 (サイズ 48,CART の場合 12 でも良い)
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 3370 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(), cpm_BaseParaManager::GetMPI_Datatype(), cpm_BaseParaManager::m_reqList, と wait_BndCommV3DEx().

参照元 cpm_wait_BndCommV3DEx_().

6.11.4.46 cpm_ErrorCode cpm_ParaManager::DecideDivPattern_CommSize (int *divNum*, int *voxSize[3]*, int *divPttn[3]*) const [protected]

並列プロセス数からI,J,K 方向の分割数を取得する

- 通信面のトータルサイズが小さい分割パターンを採用する

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 451 行で定義されています。

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

参照元 VoxelInit().

6.11.4.47 cpm_ErrorCode cpm_ParaManager::DecideDivPattern_Cube (int *divNum,* int *voxSize[3],* int *divPttn[3]*) const [protected]

並列プロセス数からI,J,K 方向の分割数を取得する

- １つのサブドメインが立方体に一番近い分割パターンを採用する

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 556 行で定義されています。

参照先 CheckCube(), CPM_ERROR_DECIDE_DIV_PATTERN, CPM_ERROR_INVALID_PTR, CPM_ERROR_INVALID_VOXELSIZE, と CPM_SUCCESS.

参照元 VoxelInit().

6.11.4.48 const cpm_VoxelInfo * cpm_ParaManager::FindVoxelInfo (int *procGrpNo* = 0) [virtual]

VOXEL 空間マップを検索

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

VOXEL 空間情報ポインタ

cpm_BaseParaManagerを実装しています。

cpm_ParaManager.cpp の 657 行で定義されています。

参照先 m_voxelInfoMap.

参照元 GetDivNum(), GetDivPos(), GetLocalOrigin(), GetLocalRegion(), GetNeighborRankID(), GetNodeHeadIndex(), GetNodeTailIndex(), GetPeriodicRankID(), GetPitch(), GetVoxelHeadIndex(), GetVoxelTailIndex(), IsInnerBoundary(), と IsOuterBoundary().

6.11.4.49 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 24 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_GetArrayHeadIndex_(), cpm_GetArrayTailIndex_(), cpm_GetDefPointType_(), cpm_GetDivNum_(), cpm_GetDivPos_(), cpm_GetGlobalArraySize_(), cpm_GetGlobalNodeSize_(), cpm_GetGlobalOrigin_(), cpm_GetGlobalRegion_(), cpm_GetGlobalVoxelSize_(), cpm_GetLocalArraySize_(), cpm_GetLocalNodeSize_(), cpm_GetLocalOrigin_(), cpm_GetLocalRegion_(), cpm_GetLocalVoxelSize_(), cpm_GetMyRankID_(), cpm_GetNeighborRankID_(), cpm_GetNodeHeadIndex_(), cpm_GetNodeTailIndex_(), cpm_GetNumRank_(), cpm_GetPeriodicRankID_(), cpm_GetPitch_(), cpm_GetVoxelHeadIndex_(), cpm_GetVoxelTailIndex_(), cpm_Initialize_(), cpm_Irecv_(), cpm_Isend_(), cpm_IsParallel_(), cpm_NodeInit_(), cpm_NodeInit_nodiv_(), cpm_PeriodicCommS3D_(), cpm_PeriodicCommS4D_(), cpm_PeriodicCommS4DEx_(), cpm_PeriodicCommV3D_(), cpm_PeriodicCommV3DEx_(), cpm_Recv_(), cpm_Send_(), cpm_SetBndCommBuffer_(), cpm_VoxelInit_(), cpm_VoxelInit_nodiv_(), cpm_Wait_(), cpm_wait_BndCommS3D_(), cpm_wait_BndCommS4D_(), cpm_wait_BndCommS4DEx_(), cpm_wait_BndCommV3D_(), cpm_wait_BndCommV3DEx_(), cpm_Waitall_(), と get_instance().

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

唯一のインスタンスの取得 (initialize 処理も実行)

引数

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

戻り値

インスタンスのポインタ

cpm_ParaManager.cpp の 36 行で定義されています。

参照先 CPM_SUCCESS, get_instance(), と cpm_BaseParaManager::Initialize().

6.11.4.51 const int * cpm_ParaManager::GetArrayHeadIndex (int *procGrpNo* = 0)

自ランクの始点VOXEL または頂点の全体空間でのインデクスを取得

- FVM のときはボクセル数、FDM のときは頂点数を取得
- 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

in	*procGrpNo*	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 752 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, cpm_BaseParaManager::GetDefPointType(), GetNodeHeadIndex(), と GetVoxelHeadIndex().

参照元 cpm_GetArrayHeadIndex_(), GetBndIndexExtGc(), と Global2LocalIndex().

6.11.4.52 const int * cpm_ParaManager::GetArrayTailIndex (int *procGrpNo* = 0)

自ランクの終点ボクセルまたは頂点の全体空間でのインデクスを取得

- FVM のときはボクセル数、FDM のときは頂点数を取得
- 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

in	*procGrpNo*	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 795 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, cpm_BaseParaManager::GetDefPointType(), GetNodeTailIndex(), と GetVoxelTailIndex().

参照元 cpm_GetArrayTailIndex_().

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

袖通信バッファの取得

- 袖通信バッファ情報の取得

引数

in	*procGrpNo*	プロセスグループ番号

戻り値

袖通信バッファ情報のポインタ

cpm_ParaManager.h の 1906 行で定義されています。

参照先 m_bndCommInfoMap.

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), wait_BndCommS4D(), と wait_BndCommS4DEx().

6.11.4.54 size_t cpm_ParaManager::GetBndCommBufferSize (int *procGrpNo* = 0) [virtual]

袖通信バッファサイズの取得

- 袖通信バッファとして確保されている配列サイズ (byte) を返す

引数

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

戻り値

バッファサイズ (byte)

cpm_BaseParaManagerを実装しています。

cpm_ParaManager.cpp の 1054 行で定義されています。

参照先 S_BNDCOMM_BUFFER::CalcBufferSize(), GetBndCommBuffer(), と m_bndCommInfoMap.

6.11.4.55 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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クスを 0 としたときの，i,j,k 各方向の スタートインデクスと長さを取得する．

引数

in	*id*	判定する id
in	*array*	判定対象の配列ポインタ
in	*vc*	仮想セル数
out	*ista*	I 方向範囲のスタートインデクス
out	*jsta*	J 方向範囲のスタートインデクス
out	*ksta*	K 方向範囲のスタートインデクス
out	*ilen*	I 方向範囲の長さ
out	*jlen*	J 方向範囲の長さ
out	*klen*	K 方向範囲の長さ
in	*procGrpNo*	プロセスグループ番号 (省略時=0)
in	*padding*	パディングフラグ

戻り値

true	指定 id を含むセルが存在した
false	指定 id を含むセルが存在しない

cpm_ParaManager.cpp の 837 行で定義されています。

参照先 CPM_ARRAY_S3D, cpm_BaseParaManager::GetLocalArraySize(), と cpm_BaseParaManager::GetPaddingSize().

6.11.4.56 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 *pad_size[3],* 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クスを 0 としたときの，i,j,k 各方向の スタートインデクスと長さを取得する．

引数

in	*id*	判定する id
in	*array*	判定対象の配列ポインタ
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
out	*ista*	I 方向範囲のスタートインデクス
out	*jsta*	J 方向範囲のスタートインデクス
out	*ksta*	K 方向範囲のスタートインデクス
out	*ilen*	I 方向範囲の長さ

out	*jlen*	J 方向範囲の長さ
out	*klen*	K 方向範囲の長さ
in	*pad_size*	パディングサイズ (i,j,k)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

true	指定 id を含むセルが存在した
false	指定 id を含むセルが存在しない

cpm_ParaManager.cpp の 864 行で定義されています。

参照先 _IDX_S3D, cpm_BaseParaManager::Allreduce(), CPM_SUCCESS, GetArrayHeadIndex(), と cpm_BaseParaManager::GetGlobalArraySize().

6.11.4.57 const int ∗ cpm_ParaManager::GetDivNum (int *procGrpNo* = 0)

領域分割数を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

領域分割数整数配列のポインタ

cpm_ParaManager.cpp の 667 行で定義されています。

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

参照元 cpm_GetDivNum_().

6.11.4.58 const int ∗ cpm_ParaManager::GetDivPos (int *procGrpNo* = 0)

自ランクの領域分割位置を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

自ランクの領域分割位置整数配列のポインタ

cpm_ParaManager.cpp の 715 行で定義されています。

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

参照元 cpm_GetDivPos_().

6.11.4.59 const double ∗ cpm_ParaManager::GetLocalOrigin (int *procGrpNo* = 0)

自ランクの空間原点を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

自ランクの空間原点実数配列のポインタ

cpm_ParaManager.cpp の 691 行で定義されています。

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

参照元 cpm_GetLocalOrigin_().

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

自ランクの空間サイズを取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

自ランクの空間サイズ実数配列のポインタ

cpm_ParaManager.cpp の 703 行で定義されています。

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

参照元 cpm_GetLocalRegion_().

6.11.4.61 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

in	*procGrpNo*	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 813 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().

6.11.4.62 const int * cpm_ParaManager::GetNodeHeadIndex (int *procGrpNo* = 0)

自ランクの始点頂点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

in	*procGrpNo*	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 739 行で定義されています。

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

参照元 cpm_GetNodeHeadIndex_(), と GetArrayHeadIndex().

6.11.4.63 const int * cpm_ParaManager::GetNodeTailIndex (int *procGrpNo* = 0)

自ランクの終点頂点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

in	*procGrpNo*	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 782 行で定義されています。

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

参照元 cpm_GetNodeTailIndex_(), と GetArrayTailIndex().

6.11.4.64 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	*procGrpNo*	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 825 行で定義されています。

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

参照元 cpm_GetPeriodicRankID_(), PeriodicCommS4D(), と PeriodicCommS4DEx().

6.11.4.65 const double * cpm_ParaManager::GetPitch (int *procGrpNo* = 0)

ピッチを取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

ピッチ実数配列のポインタ

cpm_ParaManager.cpp の 679 行で定義されています。

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

参照元 cpm_GetPitch_().

6.11.4.66 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

in	*procGrpNo*	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 727 行で定義されています。

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

参照元 cpm_GetVoxelHeadIndex_(), と GetArrayHeadIndex().

6.11.4.67 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

in	*procGrpNo*	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 770 行で定義されています。

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

参照元 cpm_GetVoxelTailIndex_(), と GetArrayTailIndex().

6.11.4.68 bool cpm_ParaManager::Global2LocalIndex (int *iG*, int *jG*, int *kG*, int & *iL*, int & *jL*, int & *kL*, int *procGrpNo* = 0)

グローバルインデクスからローカルインデクスを計算

- 全体空間実セルのスタートインデクスを 0 としたときの，i,j,k 各方向の ローカルインデクスの計算と自領域に含まれるかの判定を行う．

引数

in	*iG*	グローバルの i インデクス
in	*jG*	グローバルの j インデクス
in	*kG*	グローバルの k インデクス
out	*iL*	ローカルの i インデクス
out	*jL*	ローカルの i インデクス
out	*kL*	ローカルの i インデクス
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

true	自領域の実計算セルに含まれる
false	自領域の実計算セルに含まれない

cpm_ParaManager.cpp の 937 行で定義されています。

参照先 GetArrayHeadIndex(), と cpm_BaseParaManager::GetLocalArraySize().

6.11.4.69 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

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

戻り値

true	内部境界
false	内部境界でない

cpm_ParaManager.cpp の 974 行で定義されています。

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

6.11.4.70 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

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

戻り値

true	外部境界
false	外部境界でない

cpm_ParaManager.cpp の 961 行で定義されています。

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

6.11.4.71 cpm_ErrorCode cpm_ParaManager::NodeInit (int *div[3]*, int *nod[3]*, double *origin[3]*, double *region[3]*, size_t *maxVC* = 1, size_t *maxN* = 3, cpm_DivPolicy *divPolicy* = DIV_COMM_SIZE, int *procGrpNo* = 0) [virtual]

領域分割 (FDM 用)

- 領域分割の各種情報を引数で渡して領域分割を行う

- プロセスグループの全てのランクが活性ドメインになる
- I,J,K 方向の領域分割数を指定するバージョン

引数

in	*div*	領域分割数
in	*nod*	空間全体の頂点数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*maxVC*	最大の袖数 (袖通信用)
in	*maxN*	最大の成分数 (袖通信用)
in	*divPolicy*	自動分割ポリシー
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 265 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_ERROR_ALREADY_NODEINIT, CPM_ERROR_ALREADY_VOXELINIIT, CPM_ERROR_INSERT_DEFPOINTTYPEMAP, CPM_ERROR_INVALID_NODESIZE, CPM_ERROR_INVALID_VOXELSIZE, CPM_SUCCESS, cpm_BaseParaManager::m_defPointMap, と VoxelInit().

参照元 cpm_NodeInit_(), cpm_NodeInit_nodiv_(), と NodeInit().

6.11.4.72 cpm_ErrorCode cpm_ParaManager::NodeInit (int *nod[3],* double *origin[3],* double *region[3],* size_t *maxVC* = 1, size_t *maxN* = 3, cpm_DivPolicy *divPolicy* = DIV_COMM_SIZE, int *procGrpNo* = 0) [virtual]

領域分割 (FDM 用)

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

引数

in	*nod*	空間全体の頂点数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*maxVC*	最大の袖数 (袖通信用)
in	*maxN*	最大の成分数 (袖通信用)
in	*divPolicy*	自動分割ポリシー
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 299 行で定義されています。

参照先 NodeInit().

6.11.4.73 cpm_ErrorCode cpm_ParaManager::NodeInit_Subdomain (int *div[3],* int *nod[3],* double *origin[3],* double *region[3],* std::string *subDomainFile,* size_t *maxVC* = 1, size_t *maxN* = 3, int *procGrpNo* = 0) [virtual]

領域分割 (ActiveSubdomain 指定)(FDM 用)

- 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	*div*	領域分割数
in	*nod*	空間全体の頂点数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*subDomainFile*	ActiveSubdomain ファイル名
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 406 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_ERROR_INSERT_DEFPOINTTYPEMAP, CPM_SUCCESS, cpm_-BaseParaManager::m_defPointMap, と VoxelInit_Subdomain().

参照元 NodeInit_Subdomain().

6.11.4.74 cpm_ErrorCode cpm_ParaManager::NodeInit_Subdomain (int *nod[3],* double *origin[3],* double *region[3],* std::string *subDomainFile,* size_t *maxVC* = 1, size_t *maxN* = 3, int *procGrpNo* = 0) [virtual]

領域分割 (ActiveSubdomain 指定)(FDM 用)

- 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	*nod*	空間全体の頂点数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*subDomainFile*	ActiveSubdomain ファイル名
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 437 行で定義されています。

参照先 NodeInit_Subdomain().

6.11.4.75 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*, int *pad_size[4]*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*, int *procGrpNo*)

cpm_ParaManager_BndComm.h の 656 行で定義されています。

参照先 _IDX_S4D_PAD, _IDXFX, CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, CPM_SUCCESS, cpm_BaseParaManager::GetDefPointType(), と cpm_Base::IsRankNull().

6.11.4.76 template<class T > cpm_ErrorCode cpm_ParaManager::packX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, int *pad_size[4]*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*, int *procGrpNo*) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) のX 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
in	*pad_size*	パディングサイズ (i,j,k,n)
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
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)

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

6.11.4.77 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*, int *pad_size[4]*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*, int *procGrpNo*)

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

参照先 _IDX_S4DEX_PAD, _IDXFX, CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, CPM_SUCCESS, cpm_BaseParaManager::GetDefPointType(), と cpm_Base::IsRankNull().

6.11.4.78 template<class T > cpm_ErrorCode cpm_ParaManager::packXEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *pad_size[4]*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*, int *procGrpNo*) [protected]

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

引数

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

in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (n,i,j,k)
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
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)

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

6.11.4.79 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,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*)

cpm_ParaManager_BndComm.h の 739 行で定義されています。

参照先 _IDX_S4D_PAD, _IDXFY, CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, CPM_SUCCESS, cpm_BaseParaManager::GetDefPointType(), と cpm_Base::IsRankNull().

6.11.4.80 template<class T > cpm_ErrorCode cpm_ParaManager::packY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*) [protected]

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
in	*pad_size*	パディングサイズ (i,j,k,n)
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
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)

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

6.11.4.81 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,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*)

cpm_ParaManager_BndCommEx.h の 682 行で定義されています。

参照先 _IDX_S4DEX_PAD, _IDXFY, CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, CPM_SUCCESS, cpm_BaseParaManager::GetDefPointType(), と cpm_Base::IsRankNull().

6.11.4.82 template<class T > cpm_ErrorCode cpm_ParaManager::packYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*) [protected]

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (n,i,j,k)
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
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)

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

6.11.4.83 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,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*)

cpm_ParaManager_BndComm.h の 822 行で定義されています。

参照先 _IDX_S4D_PAD, _IDXFZ, CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, CPM_SUCCESS, cpm_BaseParaManager::GetDefPointType(), と cpm_Base::IsRankNull().

6.11.4.84 template<class T > cpm_ErrorCode cpm_ParaManager::packZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*) [protected]

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

引数

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

in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (i,j,k,n)
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
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)

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

6.11.4.85 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,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*)

cpm_ParaManager_BndCommEx.h の 765 行で定義されています。

参照先 _IDX_S4DEX_PAD, _IDXFZ, CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, CPM_SUCCESS, cpm_BaseParaManager::GetDefPointType(), と cpm_Base::IsRankNull().

6.11.4.86 template<class T > cpm_ErrorCode cpm_ParaManager::packZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp,* int *procGrpNo*) [protected]

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (n,i,j,k)
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
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)

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

6.11.4.87 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 186 行で定義されています。

参照先 CPM_ARRAY_S3D, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4D().

参照元 cpm_PeriodicCommS3D_().

6.11.4.88 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ARRAY_S3D, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4D().

6.11.4.89 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*dir*	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	*pm*	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 218 行で定義されています。

参照先 CPM_ARRAY_S4D, と cpm_BaseParaManager::GetPaddingSize().

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

6.11.4.90 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

周期境界袖通信 (Scalar4D 版, MPI_Datatype 指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*dir*	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	*pm*	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 326 行で定義されています。

参照先 CPM_ARRAY_V3D, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4D().

6.11.4.91 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 *pad_size[4],* int *procGrpNo*)

周期境界袖通信 (Scalar4D 版, パディングサイズ指定)

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

引数

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*dir*	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	*pm*	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	*pad_size*	パディングサイズ (i,j,k,n)
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 506 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(), cpm_BaseParaManager::Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

6.11.4.92 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 *pad_size[4],* int *procGrpNo*)

周期境界袖通信 (Scalar4D 版, MPI_Datatype 指定, パディングサイズ指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*dir*	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	*pm*	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	*pad_size*	パディングサイズ (i,j,k,n)
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 342 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4D().

6.11.4.93 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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

in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 162 行で定義されています。

参照先 CPM_ARRAY_S4DEX, と cpm_BaseParaManager::GetPaddingSize().

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

6.11.4.94 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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
in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 625 行で定義されています。

参照先 CPM_ARRAY_S4DEX, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4DEx().

6.11.4.95 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 *pad_size[4],* int *procGrpNo* = 0)

周期境界袖通信 (Scalar4DEx 版, パディングサイズ指定)

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

引数

in,out	*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	*pad_size*	パディングサイズ (n,i,j,k)
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 449 行で定義されています。

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

6.11.4.96 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 *pad_size[4]*, int *procGrpNo* = 0)

周期境界袖通信 (Scalar4DEx 版, MPI_Datatype 指定, パディングサイズ指定)

- (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
in,out	*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	*pad_size*	パディングサイズ (n,i,j,k)
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 641 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4DEx().

6.11.4.97 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ARRAY_V3D, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4D().

参照元 cpm_PeriodicCommV3D_().

6.11.4.98 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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
in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ARRAY_V3D, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4D().

6.11.4.99 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ARRAY_V3DEX, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4DEx().

参照元 cpm_PeriodicCommV3DEx_().

6.11.4.100 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, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in,out	*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
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 609 行で定義されています。

参照先 CPM_ARRAY_V3DEX, cpm_BaseParaManager::GetPaddingSize(), と PeriodicCommS4DEx().

6.11.4.101 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*)

cpm_ParaManager_BndComm.h の 906 行で定義されています。

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

6.11.4.102 template<class T > 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) [protected]

1 方向 (プラス、マイナス) の双方向袖通信処理

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), PeriodicCommS4D(), と PeriodicCommS4DEx().

6.11.4.103 cpm_ErrorCode cpm_ParaManager::SetBndCommBuffer (size_t *maxVC*, size_t *maxN*, int *procGrpNo* = 0) [virtual]

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManagerを実装しています。

cpm_ParaManager.cpp の 987 行で定義されています。

参照先 CPM_ERROR_BNDCOMM, CPM_ERROR_BNDCOMM_ALLOC_BUFFER, CPM_ERROR_BNDCOMM_VOXELSIZE, CPM_SUCCESS, cpm_BaseParaManager::GetLocalArraySize(), 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().

6.11.4.104 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*, int *pad_size[4]*, T * *recvm*, T * *recvp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndComm.h の 703 行で定義されています。

参照先 _IDX_S4D_PAD, _IDXFX, CPM_SUCCESS, と cpm_Base::IsRankNull().

6.11.4.105 template<class T > cpm_ErrorCode cpm_ParaManager::unpackX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, int *pad_size[4]*, T * *recvm*, T * *recvp*, int *nIDm*, int *nIDp*) [protected]

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

引数

in,out	*array*	袖通信をした配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (i,j,k,n)
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	*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)

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

6.11.4.106 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,* int *pad_size[4],* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndCommEx.h の 646 行で定義されています。

参照先 _IDX_S4DEX_PAD, _IDXFX, CPM_SUCCESS, と cpm_Base::IsRankNull().

6.11.4.107 template<class T > cpm_ErrorCode cpm_ParaManager::unpackXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*) [protected]

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

引数

in,out	*array*	袖通信をした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	*pad_size*	パディングサイズ (n,i,j,k)
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	*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)

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

6.11.4.108 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,* int *pad_size[4],* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndComm.h の 786 行で定義されています。

参照先 _IDX_S4D_PAD, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

6.11.4.109 template<class T > cpm_ErrorCode cpm_ParaManager::unpackY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*) [protected]

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

引数

in,out	*array*	袖通信をした配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (i,j,k,n)
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	*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)

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

6.11.4.110 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*, int *pad_size[4]*, T * *recvm*, T * *recvp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndCommEx.h の 729 行で定義されています。

参照先 _IDX_S4DEX_PAD, _IDXFY, CPM_SUCCESS, と cpm_Base::IsRankNull().

6.11.4.111 template<class T > cpm_ErrorCode cpm_ParaManager::unpackYEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *pad_size[4]*, T * *recvm*, T * *recvp*, int *nIDm*, int *nIDp*) [protected]

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

引数

in,out	*array*	袖通信をした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	*pad_size*	パディングサイズ (n,i,j,k)
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	*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)

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

6.11.4.112 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*, int *pad_size[4]*, T * *recvm*, T * *recvp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndComm.h の 870 行で定義されています。

参照先 _IDX_S4D_PAD, _IDXFZ, CPM_SUCCESS, と cpm_Base::IsRankNull().

6.11.4.113 template<class T > cpm_ErrorCode cpm_ParaManager::unpackZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*) [protected]

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

引数

in,out		
in	*array*	袖通信をした配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (i,j,k,n)
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	*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)

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

6.11.4.114 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,* int *pad_size[4],* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndCommEx.h の 812 行で定義されています。

参照先 _IDX_S4DEX_PAD, _IDXFZ, CPM_SUCCESS, と cpm_Base::IsRankNull().

6.11.4.115 template<class T > cpm_ErrorCode cpm_ParaManager::unpackZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *pad_size[4],* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*) [protected]

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

引数

in,out		
in	*array*	袖通信をした配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*pad_size*	パディングサイズ (n,i,j,k)
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	*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)

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

6.11.4.116 cpm_ErrorCode cpm_ParaManager::VoxelInit (cpm_GlobalDomainInfo * *domainInfo*, size_t *maxVC* = 1, size_t *maxN* = 3, int *procGrpNo* = 0) [virtual]

領域分割 (FVM 用)

- 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 111 行で定義されています。

参照先 cpm_BaseParaManager::Abort(), cpm_GlobalDomainInfo::CheckData(), CPM_DEFPOINTTYPE_FVM, CPM_ERROR_ALREADY_VOXELINIIT, CPM_ERROR_INSERT_DEFPOINTTYPEMAP, CPM_ERROR_INSERT_VOXELMAP, CPM_ERROR_INVALID_PTR, CPM_ERROR_MISMATCH_NP_SUBDOMAIN, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, cpm_BaseParaManager::GetMPI_Comm(), cpm_GlobalDomainInfo::GetSubdomainNum(), cpm_VoxelInfoCART::Init(), cpm_Base::IsCommNull(), cpm_BaseParaManager::m_defPointMap, cpm_BaseParaManager::m_procGrpList, m_voxelInfoMap, と SetBndCommBuffer().

参照元 cpm_VoxelInit_(), cpm_VoxelInit_nodiv_(), NodeInit(), VoxelInit(), と VoxelInit_Subdomain().

6.11.4.117 cpm_ErrorCode cpm_ParaManager::VoxelInit (int *div[3]*, int *vox[3]*, double *origin[3]*, double *region[3]*, size_t *maxVC* = 1, size_t *maxN* = 3, cpm_DivPolicy *divPolicy* = DIV_COMM_SIZE, int *procGrpNo* = 0) [virtual]

領域分割 (FVM 用)

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

引数

in	*div*	領域分割数

in	*vox*	空間全体のボクセル数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*maxVC*	最大の袖数 (袖通信用)
in	*maxN*	最大の成分数 (袖通信用)
in	*divPolicy*	自動分割ポリシー
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 195 行で定義されています。

参照先 CPM_ERROR_INVALID_REGION, CPM_ERROR_INVALID_VOXELSIZE, CPM_SUCCESS, DecideDivPattern_CommSize(), DecideDivPattern_Cube(), DIV_COMM_SIZE, cpm_BaseParaManager::GetNumRank(), cpm_GlobalDomainInfo::SetDivNum(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), と VoxelInit().

6.11.4.118 cpm_ErrorCode cpm_ParaManager::VoxelInit (int *vox[3],* double *origin[3],* double *region[3],* size_t *maxVC* = 1, size_t *maxN* = 3, cpm_DivPolicy *divPolicy* = DIV_COMM_SIZE, int *procGrpNo* = 0) [virtual]

領域分割 (FVM 用)

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

引数

in	*vox*	空間全体のボクセル数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*maxVC*	最大の袖数 (袖通信用)
in	*maxN*	最大の成分数 (袖通信用)
in	*divPolicy*	自動分割ポリシー
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 253 行で定義されています。

参照先 VoxelInit().

6.11.4.119 cpm_ErrorCode cpm_ParaManager::VoxelInit_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) [virtual]

領域分割 (ActiveSubdomain 指定)(FVM 用)

- 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	*div*	領域分割数
in	*vox*	空間全体のボクセル数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*subDomainFile*	ActiveSubdomain ファイル名
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 313 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, cpm_BaseParaManager::GetNumRank(), cpm_GlobalDomainInfo::ReadActiveSubdomainFile(), cpm_GlobalDomainInfo::SetDivNum(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), と VoxelInit().

参照元 NodeInit_Subdomain(), と VoxelInit_Subdomain().

6.11.4.120 cpm_ErrorCode cpm_ParaManager::VoxelInit_Subdomain (int *vox[3],* double *origin[3],* double *region[3],* std::string *subDomainFile,* size_t *maxVC* = 1, size_t *maxN* = 3, int *procGrpNo* = 0) [virtual]

領域分割 (ActiveSubdomain 指定)(FVM 用)

- 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	*vox*	空間全体のボクセル数
in	*origin*	空間全体の原点
in	*region*	空間全体のサイズ
in	*subDomainFile*	ActiveSubdomain ファイル名
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 393 行で定義されています。

参照先 VoxelInit_Subdomain().

6.11.4.121 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 138 行で定義されています。

参照先 CPM_ARRAY_S3D, cpm_BaseParaManager::GetPaddingSize(), と wait_BndCommS4D().

参照元 cpm_wait_BndCommS3D().

6.11.4.122 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[48]*, int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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
in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.123 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 *pad_size[4]*, int *procGrpNo*)

cpm_ParaManager_BndComm.h の 428 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(), GetNeighbor-RankID(), 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(), cpm_BaseParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.11.4.124 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 170 行で定義されています。

参照先 CPM_ARRAY_S4D, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_wait_BndCommS4D(), wait_BndCommS3D(), と wait_BndCommV3D().

6.11.4.125 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)

in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.126 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[48],* int *pad_size[4],* int *procGrpNo*)

非同期版袖通信の wait、展開 (Scalar4D 版, パディングサイズ指定)

- (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

引数

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (n,i,j,k)
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)

6.11.4.127 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[48],* int *pad_size[4],* int *procGrpNo*)

非同期版袖通信の wait、展開 (Scalar4D 版, MPI_Datatype 指定, パディングサイズ指定)

- (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	*dtype*	袖通信データのMPI_Datatype
in,out	*array*	袖通信をする配列の先頭ポインタ

in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*req*	MPI リクエスト (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (n,i,j,k)
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)

6.11.4.128 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 *pad_size[4],* int *procGrpNo*)

cpm_ParaManager_BndCommEx.h の 371 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(), GetNeighbor-RankID(), 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(), cpm_BaseParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.11.4.129 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::wait_BndCommS4DEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 130 行で定義されています。

参照先 CPM_ARRAY_S4DEX, と cpm_BaseParaManager::GetPaddingSize().

参照元 cpm_wait_BndCommS4DEx(), と wait_BndCommV3DEx().

6.11.4.130 cpm_ErrorCode cpm_ParaManager::wait_BndCommS4DEx (MPI_Datatype *dtype*, void * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, MPI_Request *req[48]*, int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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
in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.131 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[48]*, int *pad_size[4]*, int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar4DEx 版, パディングサイズ指定)

- (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

引数

in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (n,i,j,k)
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)

6.11.4.132 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[48]*, int *pad_size[4]*, int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar4DEx 版, MPI_Datatype 指定, パディングサイズ指定)

- (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
in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*pad_size*	パディングサイズ (n,i,j,k)
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)

6.11.4.133 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 154 行で定義されています。

参照先 CPM_ARRAY_V3D, cpm_BaseParaManager::GetPaddingSize(), と wait_BndCommS4D().

参照元 cpm_wait_BndCommV3D().

6.11.4.134 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.4.135 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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

in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 114 行で定義されています。

参照先 CPM_ARRAY_V3DEX, cpm_BaseParaManager::GetPaddingSize(), と wait_BndCommS4DEx().

参照元 cpm_wait_BndCommV3DEx().

6.11.4.136 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[48],* int *procGrpNo* = 0, CPM_PADDING *padding* = CPM_PADDING_OFF)

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	*dtype*	袖通信データのMPI_Datatype
in,out	*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 (サイズ 48、CART の場合 12 でも良い)
in	*procGrpNo*	プロセスグループ番号
in	*padding*	パディングフラグ (true:ON、false:OFF)

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.11.5 フレンドと関連する関数

6.11.5.1 friend class cpm_BaseParaManager [friend]

cpm_ParaManager.h の 76 行で定義されています。

6.11.6 変数

6.11.6.1 BndCommInfoMap cpm_ParaManager::m_bndCommInfoMap [protected]

プロセスグループ毎の袖通信バッファ情報

cpm_ParaManager.h の 2201 行で定義されています。

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

6.11.6.2 VoxelInfoMap cpm_ParaManager::m_voxelInfoMap [protected]

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 2197 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

6.12 クラス cpm_ParaManagerLMR

`#include <cpm_ParaManagerLMR.h>`

cpm_ParaManagerLMR に対する継承グラフ

cpm_ParaManagerLMR のコラボレーション図

Public メソッド

- virtual cpm_ErrorCode VoxelInit_LMR (std::string treeFile, size_t maxVC=1, size_t maxN=3, int procGrpNo=0)
- virtual const cpm_VoxelInfo * FindVoxelInfo (int procGrpNo=0)
- const cpm_VoxelInfoLMR * FindLeafVoxelInfo_byID (int leafID, int procGrpNo=0)
- const cpm_VoxelInfoLMR * FindLeafVoxelInfo (int leafIndex, int procGrpNo=0)
- int GetNumLeaf (int procGrpNo=0)
- int GetLocalNumLeaf (int procGrpNo=0)
- std::vector< int > GetLocalLeafIDs (int procGrpNo=0)
- int GetLeafID (int leafIndex, int procGrpNo=0)
- int GetLocalLeafIndex_byID (int leafID, int procGrpNo=0)
- const int * GetDivNum (int leafIndex, int procGrpNo=0)
- const double * GetPitch (int leafIndex, int procGrpNo=0)
- const double * GetLocalOrigin (int leafIndex, int procGrpNo=0)
- const double * GetLocalRegion (int leafIndex, int procGrpNo=0)
- const int * GetDivPos (int leafIndex, int procGrpNo=0)
- const int * GetVoxelHeadIndex (int leafIndex, int procGrpNo=0)
- const int * GetNodeHeadIndex (int leafIndex, int procGrpNo=0)
- const int * GetArrayHeadIndex (int leafIndex, int procGrpNo=0)
- const int * GetVoxelTailIndex (int leafIndex, int procGrpNo=0)
- const int * GetNodeTailIndex (int leafIndex, int procGrpNo=0)
- const int * GetArrayTailIndex (int leafIndex, int procGrpNo=0)
- const int * GetNeighborLeafList (int leafIndex, cpm_FaceFlag face, int &num, int procGrpNo=0)
- const int * GetPeriodicLeafList (int leafIndex, cpm_FaceFlag face, int &num, int procGrpNo=0)
- const int * GetNeighborRankList (int leafIndex, cpm_FaceFlag face, int &num, int procGrpNo=0)
- const int * GetPeriodicRankList (int leafIndex, cpm_FaceFlag face, int &num, int procGrpNo=0)
- int GetNeighborLevelDiff (int leafIndex, cpm_FaceFlag face, int procGrpNo=0)
- bool IsOuterBoundary (int leafIndex, cpm_FaceFlag face, int procGrpNo=0)
- bool IsInnerBoundary (int leafIndex, cpm_FaceFlag face, int procGrpNo=0)
- virtual size_t GetBndCommBufferSize (int procGrpNo=0)
- cpm_ErrorCode SetBndCommBuffer (size_t maxVC, size_t maxN, 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 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 BndCommS3D_nowait (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- cpm_ErrorCode BndCommS3D_nowait (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_nowait (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- cpm_ErrorCode BndCommV3D_nowait (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 BndCommV3DEx_nowait (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- cpm_ErrorCode BndCommV3DEx_nowait (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 wait_BndCommS3D (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, 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, int procGrpNo=0)
- cpm_ErrorCode wait_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 wait_BndCommV3DEx (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, 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 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 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 BndCommS4D_nowait (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, 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, 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, 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 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 BndCommS4DEx_nowait (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, 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, 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, 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 >
  CPM_INLINE cpm_ErrorCode recv_LMR (LeafCommInfoMap &commInfoMap, size_t sz_face[2], int nmax, int vc_comm, bool bPeriodic, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode send_LMR (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode copy_LMR (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMapM, LeafCommInfoMap &commInfoMapP, bool bPeriodic, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode recv_LMR_wait (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode send_LMR_wait (LeafCommInfoMap &commInfoMap)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)

- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode send_LMR_Ex (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode copy_LMR_Ex (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMapM, LeafCommInfoMap &commInfoMapP, bool bPeriodic, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode recv_LMR_Ex_wait (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)

- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo)

Static Public メソッド

- static cpm_ParaManagerLMR * get_instance ()
- static cpm_ParaManagerLMR * get_instance (int &argc, char **&argv)
- static int GetNumLeaf (std::string treeFile)

Protected メソッド

- cpm_ParaManagerLMR ()
- virtual ~cpm_ParaManagerLMR ()
- virtual double * AllocDouble (int nmax, int sz[3], int vc, int procGrpNo)
- virtual float * AllocFloat (int nmax, int sz[3], int vc, int procGrpNo)
- virtual int * AllocInt (int nmax, int sz[3], int vc, int procGrpNo)
- template<class T >
  cpm_ErrorCode recv_LMR (LeafCommInfoMap &commInfoMap, size_t sz_face[2], int nmax, int vc_comm, bool bPeriodic, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode send_LMR (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode copy_LMR (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMapM, LeafCommInfoMap &commInfoMapP, bool bPeriodic, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode recv_LMR_wait (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode send_LMR_wait (LeafCommInfoMap &commInfoMap)
- template<class T >
  cpm_ErrorCode packMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)

- template<class T >
  cpm_ErrorCode unpackMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode send_LMR_Ex (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode copy_LMR_Ex (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMapM, LeafCommInfoMap &commInfoMapP, bool bPeriodic, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode recv_LMR_Ex_wait (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, LeafCommInfoMap &commInfoMap, bool bPeriodic, cpm_FaceFlag face, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *sendbuf, size_t nw, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode unpackPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_LeafCommInfo::stCommInfo *commInfo, T *recvbuf, int procGrpNo=0)

Protected 変数

- VoxelInfoMapLMR m_voxelInfoMap
- BndCommInfoMap m_bndCommInfoMapMX
- BndCommInfoMap m_bndCommInfoMapMY
- BndCommInfoMap m_bndCommInfoMapMZ
- BndCommInfoMap m_bndCommInfoMapPX
- BndCommInfoMap m_bndCommInfoMapPY
- BndCommInfoMap m_bndCommInfoMapPZ

フレンド

- class cpm_BaseParaManager

6.12.1 説明

LMR 用の並列管理クラス

- 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_ParaManagerLMR.h の 41 行で定義されています。

6.12.2 コンストラクタとデストラクタ

6.12.2.1 cpm_ParaManagerLMR::cpm_ParaManagerLMR ( ) [protected]

コンストラクタ

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

参照先 CPM_DOMAIN_LMR, cpm_BaseParaManager::m_domainType, と m_voxelInfoMap.

6.12.2.2 cpm_ParaManagerLMR::~cpm_ParaManagerLMR ( ) [protected], [virtual]

デストラクタ

cpm_ParaManagerLMR.cpp の 91 行で定義されています。

参照先 m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, と m_bndCommInfoMapPZ.

6.12.3 関数

6.12.3.1 double * cpm_ParaManagerLMR::AllocDouble (int *nmax*, int *sz[3]*, int *vc*, int *procGrpNo*) [protected], [virtual]

配列確保 (double)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_BaseParaManagerを実装しています。

cpm_ParaManagerLMR_Alloc.cpp の 23 行で定義されています。

参照先 GetLocalNumLeaf().

6.12.3.2 float * cpm_ParaManagerLMR::AllocFloat (int *nmax*, int *sz[3]*, int *vc*, int *procGrpNo*) [protected], [virtual]

配列確保 (float)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_BaseParaManagerを実装しています。

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

参照先 GetLocalNumLeaf().

6.12.3.3 int * cpm_ParaManagerLMR::AllocInt (int *nmax*, int *sz[3]*, int *vc*, int *procGrpNo*) [protected], [virtual]

配列確保 (int)

引数

in	*nmax*	成分数
in	*sz*	配列サイズ
in	*vc*	仮想セル数
in	*procGrpNo*	領域分割を行うプロセスグループ番号

戻り値

確保した配列のポインタ

cpm_BaseParaManagerを実装しています。

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

参照先 GetLocalNumLeaf().

6.12.3.4 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の袖通信を行う

引数

in,out	*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_ParaManagerLMR_BndComm.h の 48 行で定義されています。

参照先 BndCommS4D().

参照元 cpm_BndCommS3D_LMR_().

6.12.3.5 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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_ParaManagerLMR_MPI.cpp の 27 行で定義されています。

参照先 BndCommS4D().

6.12.3.6 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::BndCommS3D_nowait (T * *array*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *procGrpNo* = 0)

非同期版袖通信 (Scalar3D 版)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
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_ParaManagerLMR_BndComm.h の 66 行で定義されています。

参照先 BndCommS4D_nowait().

6.12.3.7 cpm_ErrorCode cpm_ParaManagerLMR::BndCommS3D_nowait (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信 (Scalar3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,nLeaf) 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 と展開は行わない
- wait、展開は wait_BndCommS3D をコールする

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*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_ParaManagerLMR_MPI.cpp の 89 行で定義されています。

参照先 BndCommS4D_nowait().

6.12.3.8 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の袖通信を行う

引数

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*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_ParaManagerLMR_BndComm.h の 129 行で定義されています。

参照先 BOTH, copy_LMR(), CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, recv_LMR_wait(), send_LMR(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 BndCommS3D(), BndCommS4D(), BndCommV3D(), と cpm_BndCommS4D_LMR_().

6.12.3.9 cpm_ErrorCode cpm_ParaManagerLMR::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)

- (imax,jmax,kmax,nmax,nLeaf) 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
in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*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_ParaManagerLMR_MPI.cpp の 45 行で定義されています。

参照先 BndCommS4D(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.12.3.10 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::BndCommS4D_nowait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax,nLeaf) の形式の配列の非同期袖通信を行う
- wait と展開は行わない
- 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
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_ParaManagerLMR_BndComm.h の 367 行で定義されています。

参照先 BOTH, copy_LMR(), CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, send_LMR(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

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

6.12.3.11 cpm_ErrorCode cpm_ParaManagerLMR::BndCommS4D_nowait (MPI_Datatype *dtype*, void * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, int *procGrpNo* = 0)

非同期版袖通信 (Scalar4D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,nmax,nLeaf) 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 と展開は行わない
- wait、展開は wait_BndCommS4D をコールする

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
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_ParaManagerLMR_MPI.cpp の 107 行で定義されています。

参照先 BndCommS4D_nowait(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.12.3.12 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の袖通信を行う

引数

in,out	*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	*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_ParaManagerLMR_BndCommEx.h の 91 行で定義されています。

参照先 BOTH, copy_LMR_Ex(), CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, recv_LMR_Ex_wait(), send_LMR_Ex(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 BndCommS4DEx(), BndCommV3DEx(), と cpm_BndCommS4DEx_LMR_().

6.12.3.13 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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	*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_ParaManagerLMR_MPI.cpp の 221 行で定義されています。

参照先 BndCommS4DEx(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.12.3.14 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::BndCommS4DEx_nowait (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4DEx 版)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
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_ParaManagerLMR_BndCommEx.h の 329 行で定義されています。

参照先 BOTH, copy_LMR_Ex(), CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, send_LMR_Ex(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 BndCommS4DEx_nowait(), と BndCommV3DEx_nowait().

6.12.3.15 cpm_ErrorCode cpm_ParaManagerLMR::BndCommS4DEx_nowait (MPI_Datatype *dtype,* void * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

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

- (nmax,imax,jmax,kmax,nLeaf) 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 と展開は行わない
- wait、展開は wait_BndCommS4DEx をコールする

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*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	*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_ParaManagerLMR_MPI.cpp の 274 行で定義されています。

参照先 BndCommS4DEx_nowait(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.12.3.16 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の袖通信を行う

引数

in,out	*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_ParaManagerLMR_BndComm.h の 57 行で定義されています。

参照先 BndCommS4D().

参照元 cpm_BndCommV3D_LMR_().

6.12.3.17 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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_ParaManagerLMR_MPI.cpp の 36 行で定義されています。

参照先 BndCommS4D().

6.12.3.18 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::BndCommV3D_nowait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信 (Vector3D 版)

- (imax,jmax,kmax,3,nLeaf) の形式の配列の非同期袖通信を行う
- wait と展開は行わない
- wait、展開は wait_BndCommV3D 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
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_ParaManagerLMR_BndComm.h の 76 行で定義されています。

参照先 BndCommS4D_nowait().

6.12.3.19 cpm_ErrorCode cpm_ParaManagerLMR::BndCommV3D_nowait (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信 (Vector3D 版, MPI_Datatype 指定)

- (imax,jmax,kmax,3,nLeaf) 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 と展開は行わない
- wait、展開は wait_BndCommV3D をコールする

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*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_ParaManagerLMR_MPI.cpp の 98 行で定義されています。

参照先 BndCommS4D_nowait().

6.12.3.20 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の袖通信を行う

引数

in,out	*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_ParaManagerLMR_BndCommEx.h の 52 行で定義されています。

参照先 BndCommS4DEx().

参照元 cpm_BndCommV3DEx_LMR_().

6.12.3.21 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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_ParaManagerLMR_MPI.cpp の 213 行で定義されています。

参照先 BndCommS4DEx().

6.12.3.22 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::BndCommV3DEx_nowait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax,nLeaf) の形式の配列の非同期袖通信を行う
- wait と展開は行わない
- 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
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_ParaManagerLMR_BndCommEx.h の 61 行で定義されています。

参照先 BndCommS4DEx_nowait().

6.12.3.23 cpm_ErrorCode cpm_ParaManagerLMR::BndCommV3DEx_nowait (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

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

- (3,imax,jmax,kmax,nLeaf) 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 と展開は行わない
- wait、展開は wait_BndCommV3DEx をコールする

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*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_ParaManagerLMR_MPI.cpp の 265 行で定義されています。

参照先 BndCommS4DEx_nowait().

6.12.3.24 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::copy_LMR (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMapM,* LeafCommInfoMap & *commInfoMapP,* bool *bPeriodic,* cpm_DirFlag *dir,* cpm_PMFlag *pm,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1006 行で定義されています。

参照先 BOTH, cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::CalcSendBufferSize(), CPM_ERROR, CPM_SUCCESS, cpm_LeafCommInfo::GetBndCommSendBufferPtr(), cpm_BaseParaManager::m_rankNo, cpm_LeafCommInfo::m_vecCommInfo, MINUS2PLUS, packMX(), packMY(), packMZ(), packPX(), packPY(), packPZ(), PLUS2MINUS, cpm_LeafCommInfo::SearchDistCommInfo(), unpackMX(), unpackMY(), unpackMZ(), unpackPX(), unpackPY(), unpackPZ(), X_DIR, Y_DIR, と Z_DIR.

6.12.3.25 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::copy_LMR (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMapM,* LeafCommInfoMap & *commInfoMapP,* bool *bPeriodic,* cpm_DirFlag *dir,* cpm_PMFlag *pm,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) のランク内コピー処理

引数

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfoMapM*	通信情報マップ (マイナス側)
in	*commInfoMapP*	通信情報マップ (プラス側)
in	*bPeriodic*	周期境界フラグ (true:周期境界通信、false:内部袖通信のみ)
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

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

6.12.3.26 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::copy_LMR_Ex (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMapM,* LeafCommInfoMap & *commInfoMapP,* bool *bPeriodic,* cpm_DirFlag *dir,* cpm_PMFlag *pm,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 915 行で定義されています。

参照先 BOTH, cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::CalcSendBufferSize(), CPM_ERROR, CPM_SUCCESS, cpm_LeafCommInfo::GetBndCommSendBufferPtr(), cpm_BaseParaManager::m_rankNo, cpm_LeafCommInfo::m_vecCommInfo, MINUS2PLUS, packMXEx(), packMYEx(), packMZEx(), packPXEx(), packPYEx(), packPZEx(), PLUS2MINUS, cpm_LeafCommInfo::SearchDistCommInfo(), unpackMXEx(), unpackMYEx(), unpackMZEx(), unpackPXEx(), unpackPYEx(), unpackPZEx(), X_DIR, Y_DIR, と Z_DIR.

6.12.3.27 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::copy_LMR_Ex (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMapM,* LeafCommInfoMap & *commInfoMapP,* bool *bPeriodic,* cpm_DirFlag *dir,* cpm_PMFlag *pm,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) のランク内コピー処理

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfoMapM*	通信情報マップ (マイナス側)
in	*commInfoMapP*	通信情報マップ (プラス側)
in	*bPeriodic*	周期境界フラグ (true:周期境界通信、false:内部袖通信のみ)
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

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

6.12.3.28 const cpm_VoxelInfoLMR * cpm_ParaManagerLMR::FindLeafVoxelInfo (int *leafIndex,* int *procGrpNo* = 0)

VOXEL 空間マップを検索 (ランク内のリーフ順番号指定)

引数

in	*leafIndex*	リーフ順番号 (0～)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

VOXEL 空間情報ポインタ

cpm_ParaManagerLMR.cpp の 426 行で定義されています。

参照先 FindLeafVoxelInfo_byID(), と GetLocalLeafIDs().

参照元 FindVoxelInfo(), GetDivNum(), GetDivPos(), GetLocalOrigin(), GetLocalRegion(), GetNeighborLeafList(), GetNeighborLevelDiff(), GetNeighborRankList(), GetNodeHeadIndex(), GetNodeTailIndex(), GetPeriodicLeafList(), GetPeriodicRankList(), GetPitch(), GetVoxelHeadIndex(), GetVoxelTailIndex(), IsInnerBoundary(), と IsOuterBoundary().

6.12.3.29 const cpm_VoxelInfoLMR * cpm_ParaManagerLMR::FindLeafVoxelInfo_byID (int *leafID,* int *procGrpNo* = 0)

VOXEL 空間マップを検索 (リーフID 指定)

引数

in	*leafID*	リーフID
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

VOXEL 空間情報ポインタ

cpm_ParaManagerLMR.cpp の 439 行で定義されています。

参照先 m_voxelInfoMap.

参照元 FindLeafVoxelInfo(), と SetBndCommBuffer().

6.12.3.30 const cpm_VoxelInfo * cpm_ParaManagerLMR::FindVoxelInfo (int *procGrpNo* = 0) [virtual]

VOXEL 空間マップを検索 (0 番目のリーフ情報を取得)

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

VOXEL 空間情報ポインタ

cpm_BaseParaManagerを実装しています。

cpm_ParaManagerLMR.cpp の 418 行で定義されています。

参照先 FindLeafVoxelInfo().

6.12.3.31 cpm_ParaManagerLMR * cpm_ParaManagerLMR::get_instance ( ) [static]

インスタンスの取得

戻り値

インスタンスのポインタ

cpm_ParaManagerLMR.cpp の 23 行で定義されています。

参照元 cpm_Abort_LMR_(), cpm_Allgather_LMR_(), cpm_Allgatherv_LMR_(), cpm_Allreduce_LMR_(), cpm_Barrier_LMR_(), cpm_Bcast_LMR_(), cpm_BndCommS3D_LMR_(), cpm_BndCommS4D_LMR_(), cpm_BndCommS4DEx_LMR_(), cpm_BndCommV3D_LMR_(), cpm_BndCommV3DEx_LMR_(), cpm_Gather_LMR_(), cpm_Gatherv_LMR_(), cpm_GetArrayHeadIndex_LMR_(), cpm_GetArrayTailIndex_LMR_(), cpm_GetDefPointType_LMR_(), cpm_GetDivNum_LMR_(), cpm_GetDivPos_LMR_(), cpm_GetGlobalArraySize_LMR_(), cpm_GetGlobalNodeSize_LMR_(), cpm_GetGlobalOrigin_LMR_(), cpm_GetGlobalRegion_LMR_(), cpm_GetGlobalVoxelSize_LMR_(), cpm_GetLeafID_LMR_(), cpm_GetLocalArraySize_LMR_(), cpm_GetLocalNodeSize_LMR_(), cpm_GetLocalNumLeaf_LMR_(), cpm_GetLocalOrigin_LMR_(), cpm_GetLocalRegion_LMR_(), cpm_GetLocalVoxelSize_LMR_(), cpm_GetMyRankID_LMR_(), cpm_GetNeighborLeafList_LMR_(), cpm_GetNeighborRankList_LMR_(), cpm_GetNodeHeadIndex_LMR_(), cpm_GetNodeTailIndex_LMR_(), cpm_GetNumLeaf_LMR_(), cpm_GetNumRank_LMR_(), cpm_GetPeriodicLeafList_LMR_(), cpm_GetPeriodicRankList_LMR_(), cpm_GetPitch_LMR_(), cpm_GetVoxelHeadIndex_LMR_(), cpm_GetVoxelTailIndex_LMR(), cpm_Initialize_LMR_(), cpm_Irecv_LMR_(), cpm_Isend_LMR_(), cpm_IsParallel_LMR_(), cpm_PeriodicCommS3D_LMR_(), cpm_PeriodicCommS4D_LMR_(), cpm_PeriodicCommS4DEx_LMR_(), cpm_PeriodicCommV3D_LMR_(), cpm_PeriodicCommV3DEx_LMR_(), cpm_Recv_LMR_(), cpm_Send_LMR_(), cpm_Wait_LMR_(), cpm_Waitall_LMR_(), と get_instance().

6.12.3.32 cpm_ParaManagerLMR * cpm_ParaManagerLMR::get_instance (int & *argc,* char **& *argv*) [static]

インスタンスの取得 (initialize 処理も実行)

引数

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

戻り値

インスタンスのポインタ

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

参照先 CPM_SUCCESS, get_instance(), と cpm_BaseParaManager::Initialize().

6.12.3.33 const int ∗ cpm_ParaManagerLMR::GetArrayHeadIndex (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの始点VOXEL または頂点の全体空間でのインデクスを取得

- FVM のときはボクセル数、FDM のときは頂点数を取得
- 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

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの始点インデクス整数配列のポインタ

cpm_ParaManagerLMR.cpp の 602 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, cpm_BaseParaManager::GetDefPointType(), GetNodeHeadIndex(), と GetVoxelHeadIndex().

参照元 cpm_GetArrayHeadIndex_LMR_().

6.12.3.34 const int ∗ cpm_ParaManagerLMR::GetArrayTailIndex (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの終点VOXEL または頂点の全体空間でのインデクスを取得

- FVM のときはボクセル数、FDM のときは頂点数を取得
- 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

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの終点インデクス整数配列のポインタ

cpm_ParaManagerLMR.cpp の 646 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, cpm_BaseParaManager::GetDefPointType(), GetNodeTailIndex(), と GetVoxelTailIndex().

参照元 cpm_GetArrayTailIndex_LMR_().

6.12.3.35 size_t cpm_ParaManagerLMR::GetBndCommBufferSize (int *procGrpNo* = 0) [virtual]

袖通信バッファサイズの取得

- 袖通信バッファとして確保されている配列サイズ (byte) を返す

引数

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

戻り値

バッファサイズ (byte)

cpm_BaseParaManagerを実装しています。

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

参照先 cpm_LeafCommInfo::GetBndCommBufferSize(), m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, と REAL_BUF_TYPE.

6.12.3.36 const int * cpm_ParaManagerLMR::GetDivNum (int *leafIndex*, int *procGrpNo* = 0)

領域分割数を取得 (指定リーフのレベルにおける値)

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフのリーフレベルにおける領域分割数整数配列のポインタ

cpm_ParaManagerLMR.cpp の 518 行で定義されています。

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

参照元 cpm_GetDivNum_LMR_().

6.12.3.37 const int * cpm_ParaManagerLMR::GetDivPos (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの領域分割位置を取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの領域分割位置整数配列のポインタ

cpm_ParaManagerLMR.cpp の 566 行で定義されています。

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

参照元 cpm_GetDivPos_LMR_().

6.12.3.38 int cpm_ParaManagerLMR::GetLeafID (int *leafIndex*, int *procGrpNo* = 0)

指定リーフのリーフID を取得する

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

リーフID

cpm_ParaManagerLMR.cpp の 493 行で定義されています。

参照先 GetLocalLeafIDs().

参照元 cpm_GetLeafID_LMR_().

6.12.3.39 std::vector< int > cpm_ParaManagerLMR::GetLocalLeafIDs (int *procGrpNo* = 0)

自ランクが担当するリーフID リストを取得する

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

自ランクが担当するリーフ数

cpm_ParaManagerLMR.cpp の 473 行で定義されています。

参照先 m_voxelInfoMap.

参照元 FindLeafVoxelInfo(), GetLeafID(), GetLocalLeafIndex_byID(), GetLocalNumLeaf(), と SetBndCommBuffer().

6.12.3.40 int cpm_ParaManagerLMR::GetLocalLeafIndex_byID (int *leafID,* int *procGrpNo* = 0)

自ランクが担当するリーフID のインデクスを取得する

引数

in	*leafID*	リーフID
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

自ランク内での順番号 (0〜、ID が存在しない場合-1)

cpm_ParaManagerLMR.cpp の 502 行で定義されています。

参照先 GetLocalLeafIDs().

参照元 packMX(), packMXEx(), packMY(), packMYEx(), packMZ(), packMZEx(), packPX(), packPXEx(), packPY(), packPYEx(), packPZ(), packPZEx(), unpackMX(), unpackMXEx(), unpackMY(), unpackMYEx(), unpackMZ(), unpackMZEx(), unpackPX(), unpackPXEx(), unpackPY(), unpackPYEx(), unpackPZ(), と unpackPZEx().

6.12.3.41 int cpm_ParaManagerLMR::GetLocalNumLeaf (int *procGrpNo* = 0)

自ランクが担当するリーフ数を取得する

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

自ランクが担当するリーフ数

cpm_ParaManagerLMR.cpp の 464 行で定義されています。

参照先 GetLocalLeafIDs().

参照元 AllocDouble(), AllocFloat(), AllocInt(), cpm_GetLocalNumLeaf_LMR_(), と GetNumLeaf().

6.12.3.42 const double * cpm_ParaManagerLMR::GetLocalOrigin (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの空間原点を取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの空間原点実数配列のポインタ

cpm_ParaManagerLMR.cpp の 542 行で定義されています。

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

参照元 cpm_GetLocalOrigin_LMR_().

6.12.3.43 const double * cpm_ParaManagerLMR::GetLocalRegion (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの空間サイズを取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの空間サイズ実数配列のポインタ

cpm_ParaManagerLMR.cpp の 554 行で定義されています。

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

参照元 cpm_GetLocalRegion_LMR_().

6.12.3.44 const int * cpm_ParaManagerLMR::GetNeighborLeafList (int *leafIndex*, cpm_FaceFlag *face*, int & *num*, int *procGrpNo* = 0)

指定面における自リーフの隣接リーフ番号を取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*face*	面方向
out	*num*	面の数 (0 or 1 or 4)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定面における自リーフの隣接リーフ番号整数配列のポインタ

cpm_ParaManagerLMR.cpp の 666 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfoLMR::GetNeighborLeafList().

参照元 cpm_GetNeighborLeafList_LMR_().

6.12.3.45 int cpm_ParaManagerLMR::GetNeighborLevelDiff (int *leafIndex*, cpm_FaceFlag *face*, int *procGrpNo* = 0)

指定リーフの指定面におけるレベル差を取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*face*	面方向
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

レベル差 (0:同じレベル, 1:fine, -1:coarse)

cpm_ParaManagerLMR.cpp の 714 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfoLMR::GetNeighborLevelDiff().

6.12.3.46 const int * cpm_ParaManagerLMR::GetNeighborRankList (int *leafIndex*, cpm_FaceFlag *face*, int & *num*, int *procGrpNo* = 0)

指定リーフの指定面における自リーフの隣接ランク番号を取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*face*	面方向
out	*num*	面の数 (0 or 1 or 4)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの指定面における自リーフの隣接ランク番号整数配列のポインタ

cpm_ParaManagerLMR.cpp の 690 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfoLMR::GetNeighborRankList().

参照元 cpm_GetNeighborRankList_LMR_().

6.12.3.47 const int * cpm_ParaManagerLMR::GetNodeHeadIndex (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの始点頂点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

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの始点インデクス整数配列のポインタ

cpm_ParaManagerLMR.cpp の 590 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfo::GetNodeHeadIndex().

参照元 cpm_GetNodeHeadIndex_LMR_(), と GetArrayHeadIndex().

6.12.3.48 const int * cpm_ParaManagerLMR::GetNodeTailIndex (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの終点頂点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

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの終点インデクス整数配列のポインタ

cpm_ParaManagerLMR.cpp の 634 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfo::GetNodeTailIndex().

参照元 cpm_GetNodeTailIndex_LMR_(), と GetArrayTailIndex().

6.12.3.49 int cpm_ParaManagerLMR::GetNumLeaf (std::string *treeFile*) [static]

木情報ファイルからリーフ数を取得する

引数

in	*treefile*	木情報ファイル

戻り値

リーフ数

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

参照先 cpm_VoxelInfoLMR::GetNumLeaf().

参照元 cpm_GetNumLeaf_LMR_().

6.12.3.50 int cpm_ParaManagerLMR::GetNumLeaf (int *procGrpNo* = 0)

全リーフ数を取得する

引数

in	*procGrpNo*	プロセスグループ番号

戻り値

リーフ数

cpm_ParaManagerLMR.cpp の 453 行で定義されています。

参照先 cpm_BaseParaManager::Allreduce(), と GetLocalNumLeaf().

6.12.3.51 const int * cpm_ParaManagerLMR::GetPeriodicLeafList (int *leafIndex*, cpm_FaceFlag *face*, int & *num*, int *procGrpNo* = 0)

指定リーフの指定面における自リーフの周期境界の隣接リーフ番号を取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*face*	面方向
out	*num*	面の数 (0 or 1 or 4)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの指定面における自リーフの周期境界の隣接リーフ番号整数配列のポインタ

cpm_ParaManagerLMR.cpp の 678 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfoLMR::GetPeriodicLeafList().

参照元 cpm_GetPeriodicLeafList_LMR_().

6.12.3.52 const int * cpm_ParaManagerLMR::GetPeriodicRankList (int *leafIndex*, cpm_FaceFlag *face*, int & *num*, int *procGrpNo* = 0)

指定リーフの指定面における自リーフの周期境界の隣接ランク番号を取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*face*	面方向
out	*num*	面の数 (0 or 1 or 4)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの指定面における自リーフの周期境界の隣接ランク番号整数配列のポインタ

cpm_ParaManagerLMR.cpp の 702 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfoLMR::GetPeriodicRankList().

参照元 cpm_GetPeriodicRankList_LMR_().

6.12.3.53 const double * cpm_ParaManagerLMR::GetPitch (int *leafIndex*, int *procGrpNo* = 0)

指定リーフのピッチを取得

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフのピッチ実数配列のポインタ

cpm_ParaManagerLMR.cpp の 530 行で定義されています。

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

参照元 cpm_GetPitch_LMR_().

6.12.3.54 const int * cpm_ParaManagerLMR::GetVoxelHeadIndex (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの始点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

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの始点インデクス整数配列のポインタ

cpm_ParaManagerLMR.cpp の 578 行で定義されています。

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

参照元 cpm_GetVoxelHeadIndex_LMR_(), と GetArrayHeadIndex().

6.12.3.55 const int * cpm_ParaManagerLMR::GetVoxelTailIndex (int *leafIndex*, int *procGrpNo* = 0)

指定リーフの終点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

in	*leafIndex*	リーフ順番号 (0〜)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定リーフの終点インデクス整数配列のポインタ

cpm_ParaManagerLMR.cpp の 622 行で定義されています。

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

参照元 cpm_GetVoxelTailIndex_LMR(), と GetArrayTailIndex().

6.12.3.56 bool cpm_ParaManagerLMR::IsInnerBoundary (int *leafIndex*, cpm_FaceFlag *face*, int *procGrpNo* = 0)

指定リーフの境界が内部境界 (隣が不活性ドメイン) かどうかを判定

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*face*	面方向
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

true	内部境界
false	内部境界でない

cpm_ParaManagerLMR.cpp の 738 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfoLMR::IsInnerBoundary().

6.12.3.57 bool cpm_ParaManagerLMR::IsOuterBoundary (int *leafIndex*, cpm_FaceFlag *face*, int *procGrpNo* = 0)

指定リーフの境界が外部境界かどうかを判定

引数

in	*leafIndex*	リーフ順番号 (0〜)
in	*face*	面方向
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

true	外部境界
false	外部境界でない

cpm_ParaManagerLMR.cpp の 726 行で定義されています。

参照先 FindLeafVoxelInfo(), と cpm_VoxelInfoLMR::IsOuterBoundary().

6.12.3.58 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, cpm_LeafCommInfo::stCommInfo * *commInfo*, T * *sendbuf*, size_t *nw*, int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の-X 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と send_LMR().

6.12.3.59 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packMX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1324 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFX, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.60 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packMXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1199 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFX, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.61 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の-X 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と send_LMR_Ex().

6.12.3.62 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の-Y 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)

in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と send_LMR().

6.12.3.63 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packMY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1484 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.64 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packMYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1355 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.65 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の-Y 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と send_LMR_Ex().

6.12.3.66 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の-Z 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と send_LMR().

6.12.3.67 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packMZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1640 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.68 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の-Z 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と send_LMR_Ex().

6.12.3.69 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packMZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1511 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.70 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の+X 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と send_LMR().

6.12.3.71 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packPX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1402 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFX, cpm_LeafCommInfo::stCommInfo::CalcSendBufferSize(), CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.72 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packPXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1277 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFX, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.73 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の+X 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報

out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と send_LMR_Ex().

6.12.3.74 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の+Y 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と send_LMR().

6.12.3.75 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packPY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1562 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.76 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の+Y 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と send_LMR_Ex().

6.12.3.77 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packPYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1433 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.78 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の+Z 面への送信データのパック (通信面毎)

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と send_LMR().

6.12.3.79 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packPZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1719 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.80 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の+Z 面への送信データのパック (通信面毎)

引数

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

in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と send_LMR_Ex().

6.12.3.81 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::packPZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *sendbuf,* size_t *nw,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1589 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, cpm_BaseParaManager::m_rankNo, と stmpd_printf.

6.12.3.82 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の周期境界方向の袖通信を行う

引数

in,out	*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_ParaManagerLMR_BndComm.h の 106 行で定義されています。

参照先 PeriodicCommS4D().

参照元 cpm_PeriodicCommS3D_LMR_().

6.12.3.83 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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_ParaManagerLMR_MPI.cpp の 151 行で定義されています。

参照先 PeriodicCommS4D().

6.12.3.84 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の周期境界方向の袖通信を行う

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManagerLMR_BndComm.h の 686 行で定義されています。

参照先 BOTH, copy_LMR(), CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, MINUS2PLUS, PLUS2MINUS, recv_LMR_wait(), send_LMR(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_PeriodicCommS4D_LMR_(), PeriodicCommS3D(), PeriodicCommS4D(), と PeriodicCommV3D().

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManagerLMR_MPI.cpp の 169 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4D().

6.12.3.86 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の周期境界方向の袖通信を行う

引数

in,out	*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_ParaManagerLMR_BndCommEx.h の 648 行で定義されています。

参照先 BOTH, copy_LMR_Ex(), CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, MINUS2PLUS, PLUS2MINUS, recv_LMR_Ex_wait(), send_LMR_Ex(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_PeriodicCommS4DEx_LMR_(), PeriodicCommS4DEx(), と PeriodicCommV3DEx().

6.12.3.87 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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_ParaManagerLMR_MPI.cpp の 327 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4DEx().

6.12.3.88 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の周期境界方向の袖通信を行う

引数

in,out	*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_ParaManagerLMR_BndComm.h の 116 行で定義されています。

参照先 PeriodicCommS4D().

参照元 cpm_PeriodicCommV3D_LMR_().

6.12.3.89 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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_ParaManagerLMR_MPI.cpp の 160 行で定義されています。

参照先 PeriodicCommS4D().

6.12.3.90 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) の形式の配列の周期境界方向の袖通信を行う

引数

in,out	*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_ParaManagerLMR_BndCommEx.h の 81 行で定義されています。

参照先 PeriodicCommS4DEx().

参照元 cpm_PeriodicCommV3DEx_LMR_().

6.12.3.91 cpm_ErrorCode cpm_ParaManagerLMR::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,nLeaf) 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
in,out	*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_ParaManagerLMR_MPI.cpp の 318 行で定義されています。

参照先 PeriodicCommS4DEx().

6.12.3.92 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR (LeafCommInfoMap & *commInfoMap,* size_t *sz_face[2],* int *nmax,* int *vc_comm,* bool *bPeriodic,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 830 行で定義されています。

参照先 cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::CalcRecvBufferSize(), CPM_SUCCESS, cpm_LeafCommInfo::GetBndCommRecvBufferPtr(), cpm_BaseParaManager::Irecv(), cpm_BaseParaManager::m_rankNo, cpm_LeafCommInfo::m_reqRecv, と cpm_LeafCommInfo::m_vecCommInfo.

6.12.3.93 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR (LeafCommInfoMap & *commInfoMap,* size_t *sz_face[2],* int *nmax,* int *vc_comm,* bool *bPeriodic,* int *procGrpNo* = 0) [protected]

1 方向の非同期受信処理

引数

in	*commInfoMap*	通信情報マップ
in	*sz_face*	面内の格子数
in	*nmax*	成分数
in	*vc_comm*	通信層数
in	*bPeriodic*	周期境界フラグ (true:周期境界通信、false:内部袖通信のみ)
in	*procGrpNo*	プロセスグループ番号

6.12.3.94 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR_Ex_wait (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1080 行で定義されています。

参照先 cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::CalcRecvBufferSize(), CPM_SUCCESS, cpm_LeafCommInfo::GetBndCommRecvBufferPtr(), cpm_BaseParaManager::m_rankNo, cpm_LeafCommInfo::m_reqRecv, cpm_LeafCommInfo::m_vecCommInfo, unpackMXEx(), unpackMYEx(), unpackMZEx(), unpackPXEx(), unpackPYEx(), unpackPZEx(), cpm_BaseParaManager::Wait(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.12.3.95 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR_Ex_wait (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の 1 面の受信待機とデータの展開

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfoMap*	通信情報マップ
in	*bPeriodic*	周期境界フラグ (true:周期境界通信、false:内部袖通信のみ)
in	*face*	受信方向
in	*procGrpNo*	プロセスグループ番号

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

6.12.3.96 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR_wait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の 1 面の受信待機とデータの展開

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfoMap*	通信情報マップ
in	*bPeriodic*	周期境界フラグ (true:周期境界通信、false:内部袖通信のみ)
in	*face*	受信方向
in	*procGrpNo*	プロセスグループ番号

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

6.12.3.97 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR_wait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1174 行で定義されています。

参照先 cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::CalcRecvBufferSize(), CPM_SUCCESS, cpm_LeafCommInfo::GetBndCommRecvBufferPtr(), cpm_BaseParaManager::m_rankNo, cpm-

_LeafCommInfo::m_reqRecv, cpm_LeafCommInfo::m_vecCommInfo, unpackMX(), unpackMY(), unpackMZ(), unpackPX(), unpackPY(), unpackPZ(), cpm_BaseParaManager::Wait(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.12.3.98 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::send_LMR (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 883 行で定義されています。

参照先 cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::CalcSendBufferSize(), CPM_SUCCESS, cpm_LeafCommInfo::GetBndCommSendBufferPtr(), cpm_BaseParaManager::Isend(), cpm_BaseParaManager::m_rankNo, cpm_LeafCommInfo::m_reqSend, cpm_LeafCommInfo::m_vecCommInfo, packMX(), packMY(), packMZ(), packPX(), packPY(), packPZ(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.12.3.99 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::send_LMR (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の 1 面の送信データのパックと送信

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfoMap*	通信情報マップ
in	*bPeriodic*	周期境界フラグ (true:周期境界通信、false:内部袖通信のみ)
in	*face*	送信方向
in	*procGrpNo*	プロセスグループ番号

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

6.12.3.100 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::send_LMR_Ex (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 792 行で定義されています。

参照先 cpm_LeafCommInfo::stCommInfo::bPeriodic, cpm_LeafCommInfo::stCommInfo::CalcSendBufferSize(), CPM_SUCCESS, cpm_LeafCommInfo::GetBndCommSendBufferPtr(), cpm_BaseParaManager::Isend(), cpm_BaseParaManager::m_rankNo, cpm_LeafCommInfo::m_reqSend, cpm_LeafCommInfo::m_vecCommInfo, packMXEx(), packMYEx(), packMZEx(), packPXEx(), packPYEx(), packPZEx(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.12.3.101 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::send_LMR_Ex (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* LeafCommInfoMap & *commInfoMap,* bool *bPeriodic,* cpm_FaceFlag *face,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の 1 面の送信データのパックと送信

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfoMap*	通信情報マップ
in	*bPeriodic*	周期境界フラグ (true:周期境界通信、false:内部袖通信のみ)
in	*face*	送信方向
in	*procGrpNo*	プロセスグループ番号

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

6.12.3.102 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::send_LMR_wait (LeafCommInfoMap & *commInfoMap*) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の 1 面の送信待機

引数

in	*commInfoMap*	通信情報マップ

6.12.3.103 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::send_LMR_wait (LeafCommInfoMap & *commInfoMap*)

cpm_ParaManagerLMR_BndComm.h の 1293 行で定義されています。

参照先 CPM_SUCCESS, cpm_LeafCommInfo::m_reqRecv, cpm_LeafCommInfo::m_reqSend, と cpm_BaseParaManager::Wait().

6.12.3.104 cpm_ErrorCode cpm_ParaManagerLMR::SetBndCommBuffer (size_t *maxVC*, size_t *maxN*, int *procGrpNo* = 0) [virtual]

袖通信バッファのセット (LMR 用の袖通信情報も生成する)

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_BaseParaManagerを実装しています。

cpm_ParaManagerLMR.cpp の 239 行で定義されています。

参照先 cpm_LeafCommInfo::AddCommInfo(), cpm_BaseParaManager::Barrier(), cpm_LeafCommInfo::stCommInfo::bPeriodic, CPM_ERROR_BNDCOMM, CPM_ERROR_BNDCOMM_VOXELSIZE, CPM_SUCCESS, FindLeafVoxelInfo_byID(), GetLocalLeafIDs(), cpm_BaseParaManager::GetLocalVoxelSize(), cpm_VoxelInfoLMR::GetNeighborLeafList(), cpm_VoxelInfoLMR::GetNeighborLevelDiff(), cpm_VoxelInfoLMR::GetNeighborRankList(), cpm_VoxelInfoLMR::GetPeriodicLeafList(), cpm_VoxelInfoLMR::GetPeriodicRankList(), cpm_LeafCommInfo::stCommInfo::iDistLeafID, cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, cpm_LeafCommInfo::stCommInfo::iOwnLeafID, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bnd-

CommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, cpm_BasePara-Manager::m_rankNo, cpm_LeafCommInfo::m_vecCommInfo, cpm_LeafCommInfo::SetBndCommBuffer(), cpm_-LeafCommInfo::Sort(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 VoxelInit_LMR().

6.12.3.105 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の-X 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と recv_LMR_wait().

6.12.3.106 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackMX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1797 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFX, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stComm-Info::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.107 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の-X 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と recv_LMR_Ex_wait().

6.12.3.108 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackMXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1667 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFX, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.109 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の-Y 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と recv_LMR_wait().

6.12.3.110 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackMY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 2025 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFY, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.111 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の-Y 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と recv_LMR_Ex_wait().

6.12.3.112 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackMYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1895 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFY, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.113 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の-Z 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と recv_LMR_wait().

6.12.3.114 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackMZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 2254 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFZ, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.115 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の-Z 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と recv_LMR_Ex_wait().

6.12.3.116 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackMZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 2123 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFZ, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.117 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の+X 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と recv_LMR_wait().

6.12.3.118 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackPX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 1910 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFX, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.119 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の+X 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と recv_LMR_Ex_wait().

6.12.3.120 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackPXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 1780 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFX, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.121 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の+Y 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と recv_LMR_wait().

6.12.3.122 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackPY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 2139 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFY, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.123 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の+Y 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と recv_LMR_Ex_wait().

6.12.3.124 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackPYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 2008 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFY, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.125 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar3D,4D,Vector3D 版) の+Z 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR(), と recv_LMR_wait().

6.12.3.126 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackPZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo*)

cpm_ParaManagerLMR_BndComm.h の 2367 行で定義されています。

参照先 _IDX_S4D_LMR, _IDXFZ, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.127 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* cpm_LeafCommInfo::stCommInfo * *commInfo,* T * *recvbuf,* int *procGrpNo* = 0) [protected]

袖通信 (Scalar4DEx,Vector3DEx 版) の+Z 面からの受信データの展開 (通信面毎)

引数

in,out		
in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*commInfo*	リーフ間の通信情報
in	*recvbuf*	受信バッファ
in	*procGrpNo*	プロセスグループ番号

参照元 copy_LMR_Ex(), と recv_LMR_Ex_wait().

6.12.3.128 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::unpackPZEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, cpm_LeafCommInfo::stCommInfo * *commInfo*, T * *recvbuf*, int *procGrpNo*)

cpm_ParaManagerLMR_BndCommEx.h の 2236 行で定義されています。

参照先 _IDX_S4DEX_LMR, _IDXFZ, CPM_SUCCESS, GetLocalLeafIndex_byID(), cpm_LeafCommInfo::stCommInfo::iFaceIdx, cpm_LeafCommInfo::stCommInfo::iLevelDiff, と cpm_LeafCommInfo::stCommInfo::iOwnLeafID.

6.12.3.129 cpm_ErrorCode cpm_ParaManagerLMR::VoxelInit_LMR (std::string *treeFile*, size_t *maxVC* = 1, size_t *maxN* = 3, int *procGrpNo* = 0) [virtual]

LMR 用の領域分割

- FXgen 出力の領域情報ファイル、木情報ファイルを渡して領域分割情報を生成する

引数

in	*treefile*	木情報ファイル
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_ParaManagerLMR.cpp の 119 行で定義されています。

参照先 cpm_BaseParaManager::Abort(), CPM_DEFPOINTTYPE_FVM, CPM_ERROR_ALREADY_VOXELINIIT, CPM_ERROR_INSERT_DEFPOINTTYPEMAP, CPM_ERROR_INSERT_VOXELMAP, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, cpm_BaseParaManager::GetMPI_Comm(), cpm_VoxelInfoLMR::Init(), cpm_Base::IsCommNull(), cpm_BaseParaManager::m_defPointMap, cpm_BaseParaManager::m_procGrpList, m_voxelInfoMap, と SetBndCommBuffer().

6.12.3.130 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::wait_BndCommS3D (T * *array*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar3D 版)

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

引数

in,out	*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_ParaManagerLMR_BndComm.h の 86 行で定義されています。

参照先 wait_BndCommS4D().

6.12.3.131 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::wait_BndCommS4D (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar4D 版)

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

引数

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*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_ParaManagerLMR_BndComm.h の 538 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, recv_LMR_wait(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 wait_BndCommS3D(), と wait_BndCommV3D().

6.12.3.132 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::wait_BndCommS4DEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar4DEx 版)

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

引数

in,out	*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	*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_ParaManagerLMR_BndCommEx.h の 500 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, m_bndCommInfoMapMX, m_bndCommInfoMapMY, m_bndCommInfoMapMZ, m_bndCommInfoMapPX, m_bndCommInfoMapPY, m_bndCommInfoMapPZ, recv_LMR_Ex_wait(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 wait_BndCommV3DEx().

6.12.3.133 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::wait_BndCommV3D (T * *array*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Vector3D 版)

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

引数

in,out	*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_ParaManagerLMR_BndComm.h の 96 行で定義されています。

参照先 wait_BndCommS4D().

6.12.3.134 cpm_ErrorCode cpm_ParaManagerLMR::wait_BndCommV3D (MPI_Datatype *dtype*, void * *array*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *procGrpNo* = 0)

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

- (imax,jmax,kmax,3,nLeaf) 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
in,out	*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)

6.12.3.135 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::wait_BndCommV3DEx (T * *array*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Vector3DEx 版)

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

引数

in,out	*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_ParaManagerLMR_BndCommEx.h の 71 行で定義されています。

参照先 wait_BndCommS4DEx().

6.12.4 フレンドと関連する関数

6.12.4.1 friend class cpm_BaseParaManager [friend]

cpm_ParaManagerLMR.h の 43 行で定義されています。

6.12.5 変数

6.12.5.1 BndCommInfoMap cpm_ParaManagerLMR::m_bndCommInfoMapMX [protected]

-X 方向袖通信情報

cpm_ParaManagerLMR.h の 1569 行で定義されています。

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), SetBndCommBuffer(), wait_BndCommS4D(), wait_BndCommS4DEx(), と ~cpm_ParaManagerLMR().

6.12.5.2 BndCommInfoMap cpm_ParaManagerLMR::m_bndCommInfoMapMY [protected]

-Y 方向袖通信情報

cpm_ParaManagerLMR.h の 1572 行で定義されています。

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), SetBndCommBuffer(), wait_BndCommS4D(), wait_BndCommS4DEx(), と ~cpm_ParaManagerLMR().

6.12.5.3 BndCommInfoMap cpm_ParaManagerLMR::m_bndCommInfoMapMZ [protected]

-Z 方向袖通信情報

cpm_ParaManagerLMR.h の 1575 行で定義されています。

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), SetBndCommBuffer(), wait_BndCommS4D(), wait_BndCommS4DEx(), と ~cpm_ParaManagerLMR().

6.12.5.4 BndCommInfoMap cpm_ParaManagerLMR::m_bndCommInfoMapPX [protected]

+X 方向袖通信情報

cpm_ParaManagerLMR.h の 1578 行で定義されています。

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), SetBndCommBuffer(), wait_BndCommS4D(), wait_BndCommS4DEx(), と ~cpm_ParaManagerLMR().

6.12.5.5 BndCommInfoMap cpm_ParaManagerLMR::m_bndCommInfoMapPY [protected]

+Y 方向袖通信情報

cpm_ParaManagerLMR.h の 1581 行で定義されています。

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), SetBndCommBuffer(), wait_BndCommS4D(), wait_BndCommS4DEx(), と ~cpm_ParaManagerLMR().

6.12.5.6 BndCommInfoMap cpm_ParaManagerLMR::m_bndCommInfoMapPZ [protected]

+Z 方向袖通信情報

cpm_ParaManagerLMR.h の 1584 行で定義されています。

参照元 BndCommS4D(), BndCommS4D_nowait(), BndCommS4DEx(), BndCommS4DEx_nowait(), GetBndCommBufferSize(), PeriodicCommS4D(), PeriodicCommS4DEx(), SetBndCommBuffer(), wait_BndCommS4D(), wait_BndCommS4DEx(), と ~cpm_ParaManagerLMR().

6.12.5.7 VoxelInfoMapLMR cpm_ParaManagerLMR::m_voxelInfoMap [protected]

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
- １プロセス複数リーフに対応

cpm_ParaManagerLMR.h の 1566 行で定義されています。

参照元 cpm_ParaManagerLMR(), FindLeafVoxelInfo_byID(), GetLocalLeafIDs(), と VoxelInit_LMR().

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_ParaManagerLMR.h
- cpm_ParaManagerLMR.cpp
- cpm_ParaManagerLMR_Alloc.cpp
- cpm_ParaManagerLMR_MPI.cpp
- cpm_ParaManagerLMR_BndComm.h
- cpm_ParaManagerLMR_BndCommEx.h

6.13 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

Additional Inherited Members

6.13.1 説明

CPM のテキストパーサークラス

cpm_TextParser.h の 25 行で定義されています。

6.13.2 コンストラクタとデストラクタ

6.13.2.1 cpm_TextParser::cpm_TextParser ( ) [protected]

コンストラクタ

cpm_TextParser.cpp の 21 行で定義されています。

参照先 m_tp.

6.13.2.2 cpm_TextParser::~cpm_TextParser ( ) [protected], [virtual]

デストラクタ

cpm_TextParser.cpp の 30 行で定義されています。

参照先 m_tp.

6.13.3 関数

6.13.3.1 int cpm_TextParser::Read (std::string *filename*) [protected]

読み込み処理

- ユーザは直接コールできない

引数

in	*filename*	読み込むファイル名

戻り値

TextParser クラスの終了コード

cpm_TextParser.cpp の 37 行で定義されています。

参照先 m_tp.

参照元 cpm_TextParserDomain::ReadMain(), と cpm_TextParserDomainLMR::ReadMain().

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

ベクトルデータの読み込み (単精度実数版)

引数

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

戻り値

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

cpm_TextParser.cpp の 56 行で定義されています。

参照先 m_tp.

参照元 cpm_TextParserDomainLMR::ReadDomain(), cpm_TextParserDomain::ReadDomainInfo(), と cpm_TextParserDomainLMR::ReadLeafBlock().

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

ベクトルデータの読み込み (倍精度実数版)

引数

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

戻り値

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

cpm_TextParser.cpp の 91 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.

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

ベクトルデータの読み込み (整数版)

引数

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

戻り値

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

cpm_TextParser.cpp の 126 行で定義されています。

参照先 m_tp.

6.13.4 変数

6.13.4.1 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 95 行で定義されています。

参照元 cpm_TextParser(), Read(), cpm_TextParserDomainLMR::ReadBCMTree(), cpm_TextParserDomainLMR::ReadDomain(), cpm_TextParserDomain::ReadDomainInfo(), cpm_TextParserDomainLMR::ReadLeafBlock(), 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

6.14 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)

Additional Inherited Members

6.14.1 説明

CPM の領域情報テキストパーサークラス

cpm_TextParserDomain.h の 26 行で定義されています。

6.14.2 コンストラクタとデストラクタ

6.14.2.1 cpm_TextParserDomain::cpm_TextParserDomain ( )

コンストラクタ

cpm_TextParserDomain.cpp の 22 行で定義されています。

6.14.2.2 cpm_TextParserDomain::∼cpm_TextParserDomain ( ) [virtual]

デストラクタ

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

6.14.3 関数

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

読み込み処理

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

引数

in	*filename*	読み込むファイル名
out	*errorcode*	CPM エラーコード

戻り値

領域情報ポインタ

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

参照先 ReadMain().

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

DomainInfo の読み込み

引数

in,out	*dInfo*	領域情報

戻り値

CPM エラーコード

cpm_TextParserDomain.cpp の 86 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().

6.14.3.3 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	*filename*	読み込むファイル名
out	*errorcode*	CPM エラーコード

戻り値

領域情報ポインタ

cpm_TextParserDomain.cpp の 48 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().

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

ActiveSubdomainInfo の読み込み

引数

in,out	*dInfo*	領域情報
in	*tpfname*	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 248 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

6.15 クラス cpm_TextParserDomainLMR

`#include <cpm_TextParserDomainLMR.h>`

cpm_TextParserDomainLMR に対する継承グラフ

cpm_TextParserDomainLMR のコラボレーション図

Public メソッド

- cpm_TextParserDomainLMR ()
- virtual ~cpm_TextParserDomainLMR ()

Static Public メソッド

- static cpm_ErrorCode Read (std::string filename, S_OCT_DOMAIN_INFO &domainInfo)

Private メソッド

- cpm_ErrorCode ReadMain (std::string filename, S_OCT_DOMAIN_INFO &domainInfo)
- int ReadDomain (S_OCT_DOMAIN_INFO &domainInfo)
- int ReadBCMTree (S_OCT_DOMAIN_INFO &domainInfo, std::string tpFile)
- int ReadLeafBlock (S_OCT_DOMAIN_INFO &domainInfo)

Additional Inherited Members

6.15.1 説明

LMR 用領域情報テキストパーサークラス

cpm_TextParserDomainLMR.h の 55 行で定義されています。

6.15.2 コンストラクタとデストラクタ

6.15.2.1 cpm_TextParserDomainLMR::cpm_TextParserDomainLMR ( )

コンストラクタ

cpm_TextParserDomainLMR.cpp の 22 行で定義されています。

6.15.2.2 cpm_TextParserDomainLMR::~cpm_TextParserDomainLMR ( ) [virtual]

デストラクタ

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

6.15.3 関数

6.15.3.1 cpm_ErrorCode cpm_TextParserDomainLMR::Read (std::string *filename*, S_OCT_DOMAIN_INFO & *domainInfo*) [static]

読み込み処理

- TextParser クラスを用いて領域分割情報ファイルを読み込む

- TextParser クラスのインスタンスはクリア (remove) される

引数

in	*filename*	読み込むファイル名
out	*domainInfo*	領域情報

戻り値

CPM エラーコード

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

参照先 ReadMain().

参照元 cpm_VoxelInfoLMR::GetNumLeaf(), と cpm_VoxelInfoLMR::Init().

6.15.3.2 int cpm_TextParserDomainLMR::ReadBCMTree (S_OCT_DOMAIN_INFO & *domainInfo,* std::string *tpFile*) [private]

BCMTree の読み込み

引数

in,out	*domainInfo*	領域情報
in	*tpFile*	元のテキストパーサー形式ファイル名

戻り値

CPM エラーコード

cpm_TextParserDomainLMR.cpp の 155 行で定義されています。

参照先 CPM_ERROR_TP_LMR_BCMTREE, cpm_Base::cpm_strCompare(), CPM_SUCCESS, CPM_PATH::cpmPath_concat(), CPM_PATH::cpmPath_isAbsolute(), CES::DirName(), cpm_TextParser::m_tp, と S_OCT_DOMAIN_INFO::octFile.

参照元 ReadMain().

6.15.3.3 int cpm_TextParserDomainLMR::ReadDomain (S_OCT_DOMAIN_INFO & *domainInfo*) [private]

Domain の読み込み

引数

in,out	*domainInfo*	領域情報

戻り値

CPM エラーコード

cpm_TextParserDomainLMR.cpp の 82 行で定義されています。

参照先 CPM_ERROR_TP_INVALID_G_ORG, CPM_ERROR_TP_INVALID_G_RGN, cpm_Base::cpm_strCompare(), CPM_SUCCESS, cpm_TextParser::m_tp, S_OCT_DOMAIN_INFO::origin, cpm_TextParser::readVector(), と S_OCT_DOMAIN_INFO::region.

参照元 ReadMain().

6.15.3.4 int cpm_TextParserDomainLMR::ReadLeafBlock (S_OCT_DOMAIN_INFO & *domainInfo*) [private]

LeafBlock の読み込み

引数

in,out	*domainInfo*	領域情報

戻り値

CPM エラーコード

cpm_TextParserDomainLMR.cpp の 235 行で定義されています。

参照先 CPM_ERROR_TP_LMR_LEAFBLOCK, CPM_ERROR_TP_LMR_SIZE_NOT_EVEN, CPM_ERROR_TP_LMR_UNIT, cpm_Base::cpm_strCompare(), CPM_SUCCESS, cpm_TextParser::m_tp, cpm_TextParser::readVector(), S_OCT_DOMAIN_INFO::size, と S_OCT_DOMAIN_INFO::unitLength.

参照元 ReadMain().

6.15.3.5 cpm_ErrorCode cpm_TextParserDomainLMR::ReadMain (std::string *filename*, S_OCT_DOMAIN_INFO & *domainInfo*) [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	*filename*	読み込むファイル名
in,out	*domainInfo*	領域情報

戻り値

CPM エラーコード

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

参照先 CPM_ERROR_TP_LMR_BCMTREE, CPM_ERROR_TP_LMR_DOMAIN, CPM_ERROR_TP_LMR_DOMAINFILE, CPM_ERROR_TP_LMR_LEAFBLOCK, CPM_SUCCESS, cpm_TextParser::Read(), ReadBCMTree(), ReadDomain(), と ReadLeafBlock().

参照元 Read().

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_TextParserDomainLMR.h
- cpm_TextParserDomainLMR.cpp

6.16 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

Public メソッド

- virtual ~cpm_VoxelInfo ()
- const int * GetDivNum () const

- const int * GetDivPos () const
- const double * GetPitch () const
- const double * GetGlobalPitch () const
- const int * GetGlobalVoxelSize () const
- const int * GetGlobalNodeSize () const
- const int * GetGlobalArraySize (cpm_DefPointType dtype) const
- const double * GetGlobalOrigin () const
- const double * GetGlobalRegion () const
- const int * GetLocalVoxelSize () const
- const int * GetLocalNodeSize () const
- const int * GetLocalArraySize (cpm_DefPointType dtype) const
- const double * GetLocalOrigin () const
- const double * GetLocalRegion () const
- const int * GetVoxelHeadIndex () const
- const int * GetNodeHeadIndex () const
- const int * GetArrayHeadIndex (cpm_DefPointType dtype) const
- const int * GetVoxelTailIndex () const
- const int * GetNodeTailIndex () const
- const int * GetArrayTailIndex (cpm_DefPointType dtype) const
- const int * GetNeighborRankID () const
- const int * GetPeriodicRankID () const
- virtual bool IsOuterBoundary (cpm_FaceFlag face) const
- virtual bool IsInnerBoundary (cpm_FaceFlag face) const

Protected メソッド

- cpm_VoxelInfo ()

Protected 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
- int m_nodeHeadIndex [3]
  自ランクの始点頂点インデックス
- int m_nodeTailIndex [3]
  自ランクの終点頂点インデックス
- 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

フレンド

- class cpm_BaseParaManager
- class cpm_ParaManagerCART

Additional Inherited Members

6.16.1 説明

CPM のVOXEL 空間情報管理クラス

cpm_VoxelInfo.h の 26 行で定義されています。

6.16.2 コンストラクタとデストラクタ

6.16.2.1 cpm_VoxelInfo::~cpm_VoxelInfo ( ) [virtual]

デストラクタ

cpm_VoxelInfo.cpp の 50 行で定義されています。

6.16.2.2 cpm_VoxelInfo::cpm_VoxelInfo ( ) [protected]

コンストラクタ

cpm_VoxelInfo.cpp の 21 行で定義されています。

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

6.16.3 関数

6.16.3.1 const int * cpm_VoxelInfo::GetArrayHeadIndex (cpm_DefPointType *dtype*) const

自ランクの始点頂点の全体空間でのインデクスを取得

- FVM のときはボクセル、FDM のときは頂点の始点インデックスを取得

引数

in	*dtype*	定義点タイプ (enum)

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 211 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, m_nodeHeadIndex, と m_voxelHeadIndex.

6.16.3.2 const int * cpm_VoxelInfo::GetArrayTailIndex (cpm_DefPointType *dtype*) const

自ランクの終点頂点の全体空間でのインデクスを取得

- FVM のときはボクセル、FDM のときは頂点の終点インデックスを取得

引数

in	*dtype*	定義点タイプ (enum)

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 247 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, m_nodeTailIndex, と m_voxelTailIndex.

6.16.3.3 const int ∗ cpm_VoxelInfo::GetDivNum ( ) const

領域分割数を取得

- LMR のときは最大レベルにおける分割数を返す

戻り値

領域分割数整数配列のポインタ

cpm_VoxelInfo.cpp の 57 行で定義されています。

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

参照元 cpm_VoxelInfoLMR::debugPrint(), cpm_ParaManagerLMR::GetDivNum(), cpm_ParaManager::GetDivNum(), IsInnerBoundary(), cpm_VoxelInfoLMR::IsInnerBoundary(), IsOuterBoundary(), と cpm_VoxelInfoLMR::IsOuterBoundary().

6.16.3.4 const int ∗ cpm_VoxelInfo::GetDivPos ( ) const

自ランクの領域分割位置を取得

- LMR のときは最大レベルにおける分割位置を返す

戻り値

自ランクの領域分割位置整数配列のポインタ

cpm_VoxelInfo.cpp の 65 行で定義されています。

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

参照元 cpm_VoxelInfoLMR::debugPrint(), cpm_ParaManagerLMR::GetDivPos(), cpm_ParaManager::GetDivPos(), IsInnerBoundary(), cpm_VoxelInfoLMR::IsInnerBoundary(), IsOuterBoundary(), と cpm_VoxelInfoLMR::IsOuterBoundary().

6.16.3.5 const int ∗ cpm_VoxelInfo::GetGlobalArraySize (cpm_DefPointType *dtype*) const

全体ボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数を取得

引数

in	*dtype*	定義点タイプ (enum)

戻り値

全体ボクセル数または頂点数の整数配列ポインタ (3word)

cpm_VoxelInfo.cpp の 107 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, cpm_DomainInfo::GetNodNum(), cpm_DomainInfo::GetVoxNum(), と m_globalDomainInfo.

6.16.3.6 const int * cpm_VoxelInfo::GetGlobalNodeSize ( ) const

全体頂点数を取得

戻り値

全体頂点数整数配列のポインタ

cpm_VoxelInfo.cpp の 99 行で定義されています。

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

参照元 cpm_BaseParaManager::GetGlobalNodeSize().

6.16.3.7 const double * cpm_VoxelInfo::GetGlobalOrigin ( ) const

全体空間の原点を取得

戻り値

全体空間の原点実数配列のポインタ

cpm_VoxelInfo.cpp の 125 行で定義されています。

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

参照元 cpm_BaseParaManager::GetGlobalOrigin().

6.16.3.8 const double * cpm_VoxelInfo::GetGlobalPitch ( ) const

グローバルピッチを取得

- カーテシアンのときはGetPitch と同じ
- LMR のときは最大レベルにおけるピッチ

戻り値

ピッチ実数配列のポインタ

cpm_VoxelInfo.cpp の 81 行で定義されています。

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

6.16.3.9 const double * cpm_VoxelInfo::GetGlobalRegion ( ) const

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 133 行で定義されています。

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

参照元 cpm_BaseParaManager::GetGlobalRegion().

6.16.3.10 const int * cpm_VoxelInfo::GetGlobalVoxelSize ( ) const

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 89 行で定義されています。

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

参照元 cpm_BaseParaManager::GetGlobalVoxelSize().

6.16.3.11 const int * cpm_VoxelInfo::GetLocalArraySize (cpm_DefPointType *dtype*) const

自ランクのボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数を取得

引数

in	*dtype*	定義点タイプ (enum)

戻り値

自ランクのボクセル数または頂点数の整数配列ポインタ (3word)

cpm_VoxelInfo.cpp の 159 行で定義されています。

参照先 CPM_DEFPOINTTYPE_FDM, CPM_DEFPOINTTYPE_FVM, cpm_DomainInfo::GetNodNum(), cpm_DomainInfo::GetVoxNum(), と m_localDomainInfo.

6.16.3.12 const int * cpm_VoxelInfo::GetLocalNodeSize ( ) const

自ランクの頂点数を取得

戻り値

自ランクの頂点数整数配列のポインタ

cpm_VoxelInfo.cpp の 151 行で定義されています。

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

参照元 cpm_BaseParaManager::GetLocalNodeSize().

6.16.3.13 const double * cpm_VoxelInfo::GetLocalOrigin ( ) const

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 177 行で定義されています。
参照先 cpm_DomainInfo::GetOrigin(), と m_localDomainInfo.
参照元 cpm_ParaManagerLMR::GetLocalOrigin(), と cpm_ParaManager::GetLocalOrigin().

6.16.3.14 const double * cpm_VoxelInfo::GetLocalRegion ( ) const

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 185 行で定義されています。
参照先 cpm_DomainInfo::GetRegion(), と m_localDomainInfo.
参照元 cpm_ParaManagerLMR::GetLocalRegion(), と cpm_ParaManager::GetLocalRegion().

6.16.3.15 const int * cpm_VoxelInfo::GetLocalVoxelSize ( ) const

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 141 行で定義されています。
参照先 cpm_DomainInfo::GetVoxNum(), と m_localDomainInfo.
参照元 cpm_BaseParaManager::GetLocalVoxelSize().

6.16.3.16 const int * cpm_VoxelInfo::GetNeighborRankID ( ) const

自ランクの隣接ランク番号を取得

- LMR で隣接ランクが 4 つの場合は、1 番目のランクを返す

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 265 行で定義されています。
参照先 m_neighborRankID.
参照元 cpm_ParaManager::GetNeighborRankID().

6.16.3.17 const int * cpm_VoxelInfo::GetNodeHeadIndex ( ) const

自ランクの始点頂点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 203 行で定義されています。
参照先 m_nodeHeadIndex.
参照元 cpm_ParaManagerLMR::GetNodeHeadIndex(), と cpm_ParaManager::GetNodeHeadIndex().

6.16.3.18 const int * cpm_VoxelInfo::GetNodeTailIndex ( ) const

自ランクの終点頂点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 239 行で定義されています。
参照先 m_nodeTailIndex.
参照元 cpm_ParaManagerLMR::GetNodeTailIndex(), と cpm_ParaManager::GetNodeTailIndex().

6.16.3.19 const int * cpm_VoxelInfo::GetPeriodicRankID ( ) const

自ランクの周期境界の隣接ランク番号を取得

- LMR で隣接ランクが 4 つの場合は、1 番目のランクを返す

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 273 行で定義されています。
参照先 m_periodicRankID.
参照元 cpm_ParaManager::GetPeriodicRankID().

6.16.3.20 const double * cpm_VoxelInfo::GetPitch ( ) const

ローカルピッチを取得

戻り値

ピッチ実数配列のポインタ

cpm_VoxelInfo.cpp の 73 行で定義されています。
参照先 cpm_DomainInfo::GetPitch(), と m_localDomainInfo.
参照元 cpm_ParaManagerLMR::GetPitch(), と cpm_ParaManager::GetPitch().

6.16.3.21 const int * cpm_VoxelInfo::GetVoxelHeadIndex ( ) const

自ランクの始点VOXEL の全体空間でのインデクスを取得

- LMR のときは自身のレベルにおける始点インデクスを返す

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 193 行で定義されています。

参照先 m_voxelHeadIndex.

参照元 cpm_ParaManagerLMR::GetVoxelHeadIndex(), と cpm_ParaManager::GetVoxelHeadIndex().

6.16.3.22 const int * cpm_VoxelInfo::GetVoxelTailIndex ( ) const

自ランクの終点VOXEL の全体空間でのインデクスを取得

- LMR のときは自身のレベルにおける終点インデクスを返す

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 229 行で定義されています。

参照先 m_voxelTailIndex.

参照元 cpm_ParaManagerLMR::GetVoxelTailIndex(), と cpm_ParaManager::GetVoxelTailIndex().

6.16.3.23 bool cpm_VoxelInfo::IsInnerBoundary (cpm_FaceFlag *face*) const [virtual]

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	*face*	面方向

戻り値

true	内部境界
false	内部境界でない

cpm_VoxelInfoLMRで再定義されています。

cpm_VoxelInfo.cpp の 308 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().

6.16.3.24 bool cpm_VoxelInfo::IsOuterBoundary (cpm_FaceFlag *face*) const [virtual]

自ランクの境界が外部境界かどうかを判定

引数

in	*face*	面方向

戻り値

true	外部境界
false	外部境界でない

cpm_VoxelInfoLMRで再定義されています。

cpm_VoxelInfo.cpp の 281 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().

6.16.4 フレンドと関連する関数

6.16.4.1 friend class cpm_BaseParaManager [friend]

cpm_VoxelInfo.h の 28 行で定義されています。

6.16.4.2 friend class cpm_ParaManagerCART [friend]

cpm_VoxelInfo.h の 29 行で定義されています。

6.16.5 変数

6.16.5.1 MPI_Comm cpm_VoxelInfo::m_comm [protected]

MPI コミュニケータ

cpm_VoxelInfo.h の 237 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::Init(), と cpm_VoxelInfoLMR::Init().

6.16.5.2 cpm_GlobalDomainInfo cpm_VoxelInfo::m_globalDomainInfo [protected]

空間全体の領域情報

cpm_VoxelInfo.h の 224 行で定義されています。

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), cpm_VoxelInfoCART::CreateRankMap(), cpm_VoxelInfoLMR::debugPrint(), GetDivNum(), GetGlobalArraySize(), GetGlobalNodeSize(), GetGlobalOrigin(), GetGlobalPitch(), GetGlobalRegion(), GetGlobalVoxelSize(), cpm_VoxelInfoCART::Init(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.16.5.3 cpm_LocalDomainInfo cpm_VoxelInfo::m_localDomainInfo [protected]

自ランクの領域情報

cpm_VoxelInfo.h の 227 行で定義されています。

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), cpm_VoxelInfoLMR::debugPrint(), GetDivPos(), GetLocalArraySize(), GetLocalNodeSize(), GetLocalOrigin(), GetLocalRegion(), GetLocalVoxelSize(), GetPitch(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.16.5.4 int cpm_VoxelInfo::m_neighborRankID[6] [protected]

隣接ランク番号 (外部境界は負の値)

cpm_VoxelInfo.h の 240 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), GetNeighborRankID(), IsInnerBoundary(), IsOuterBoundary(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.16.5.5 int cpm_VoxelInfo::m_nodeHeadIndex[3] [protected]

自ランクの始点頂点インデックス

cpm_VoxelInfo.h の 232 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), GetArrayHeadIndex(), と GetNodeHeadIndex().

6.16.5.6 int cpm_VoxelInfo::m_nodeTailIndex[3] [protected]

自ランクの終点頂点インデックス

cpm_VoxelInfo.h の 233 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), GetArrayTailIndex(), と GetNodeTailIndex().

6.16.5.7 int cpm_VoxelInfo::m_nRank [protected]

コミュニケータ内のランク数 (=プロセス並列数)

cpm_VoxelInfo.h の 238 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::Init(), cpm_VoxelInfoLMR::Init(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.16.5.8 int cpm_VoxelInfo::m_periodicRankID[6] [protected]

周期境界の隣接ランク番号

cpm_VoxelInfo.h の 241 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), GetPeriodicRankID(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.16.5.9 int cpm_VoxelInfo::m_rankNo [protected]

コミュニケータ内でのランク番号

cpm_VoxelInfo.h の 239 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), cpm_VoxelInfoLMR::debugPrint(), cpm_VoxelInfoCART::Init(), と cpm_VoxelInfoLMR::Init().

6.16.5.10 int cpm_VoxelInfo::m_voxelHeadIndex[3] [protected]

自ランクの始点ボクセルインデックス

cpm_VoxelInfo.h の 228 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoLMR::debugPrint(), GetArrayHeadIndex(), GetVoxelHeadIndex(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.16.5.11 int cpm_VoxelInfo::m_voxelTailIndex[3] [protected]

自ランクの終点ボクセルインデックス

cpm_VoxelInfo.h の 229 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoLMR::debugPrint(), GetArrayTailIndex(), GetVoxelTailIndex(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

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

6.17 クラス cpm_VoxelInfoCART

`#include <cpm_VoxelInfoCART.h>`

cpm_VoxelInfoCART に対する継承グラフ

cpm_VoxelInfoCART のコラボレーション図

Protected メソッド

- cpm_VoxelInfoCART ()
- virtual ~cpm_VoxelInfoCART ()
- cpm_ErrorCode Init (MPI_Comm comm, cpm_GlobalDomainInfo *dInfo)
- bool CreateRankMap ()
- bool CreateNeighborRankInfo ()
- bool CreateLocalDomainInfo ()

Protected 変数

- int * m_rankMap
  ランクマップ

フレンド

- class cpm_ParaManager
- class cpm_ParaManagerCART

Additional Inherited Members

6.17.1 説明

カーテシアン用のVOXEL 空間情報管理クラス

cpm_VoxelInfoCART.h の 25 行で定義されています。

6.17.2 コンストラクタとデストラクタ

6.17.2.1 cpm_VoxelInfoCART::cpm_VoxelInfoCART ( ) [protected]

コンストラクタ

cpm_VoxelInfoCART.cpp の 21 行で定義されています。

参照先 m_rankMap.

6.17.2.2 cpm_VoxelInfoCART::~cpm_VoxelInfoCART ( ) [protected], [virtual]

デストラクタ

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

参照先 m_rankMap.

6.17.3 関数

6.17.3.1 bool cpm_VoxelInfoCART::CreateLocalDomainInfo ( ) [protected]

ローカル領域情報を生成

戻り値

true	正常終了
false	エラー

cpm_VoxelInfoCART.cpp の 250 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(), cpm_VoxelInfo::m_globalDomainInfo, cpm_VoxelInfo::m_localDomainInfo, cpm_VoxelInfo::m_nodeHeadIndex, cpm_VoxelInfo::m_nodeTailIndex, m_rankMap, cpm_VoxelInfo::m_rankNo, cpm_VoxelInfo::m_voxelHeadIndex, cpm_VoxelInfo::m_voxelTailIndex, cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_ActiveSubdomainInfo::SetPos(), cpm_DomainInfo::SetRegion(), と cpm_DomainInfo::SetVoxNum().

参照元 Init().

6.17.3.2 bool cpm_VoxelInfoCART::CreateNeighborRankInfo ( ) [protected]

隣接ランク情報を生成

戻り値

true	正常終了
false	エラー

cpm_VoxelInfoCART.cpp の 144 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_VoxelInfo::m_globalDomainInfo, cpm_VoxelInfo::m_localDomainInfo, cpm_VoxelInfo::m_neighborRankID, cpm_VoxelInfo::m_periodicRankID, m_rankMap, cpm_VoxelInfo::m_rankNo, X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 Init().

6.17.3.3 bool cpm_VoxelInfoCART::CreateRankMap ( ) [protected]

ランクマップを生成

戻り値

true	正常終了
false	エラー

cpm_VoxelInfoCART.cpp の 81 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(), cpm_VoxelInfo::m_globalDomainInfo, と m_rankMap.

参照元 Init().

6.17.3.4 cpm_ErrorCode cpm_VoxelInfoCART::Init (MPI_Comm *comm,* cpm_GlobalDomainInfo * *dInfo*) [protected]

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	*comm*	MPI コミュニケータ
in	*dInfo*	領域分割情報

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_VoxelInfoCART.cpp の 37 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(), cpm_VoxelInfo::m_comm, cpm_VoxelInfo::m_globalDomainInfo, cpm_VoxelInfo::m_nRank, と cpm_VoxelInfo::m_rankNo.

参照元 cpm_ParaManager::VoxelInit().

6.17.4 フレンドと関連する関数

6.17.4.1 friend class cpm_ParaManager [friend]

cpm_VoxelInfoCART.h の 27 行で定義されています。

6.17.4.2 friend class cpm_ParaManagerCART [friend]

cpm_VoxelInfoCART.h の 28 行で定義されています。

6.17.5 変数

6.17.5.1 int* cpm_VoxelInfoCART::m_rankMap [protected]

ランクマップ

cpm_VoxelInfoCART.h の 83 行で定義されています。

参照元 cpm_VoxelInfoCART(), CreateLocalDomainInfo(), CreateNeighborRankInfo(), CreateRankMap(), と ~cpm_VoxelInfoCART().

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_VoxelInfoCART.h
- cpm_VoxelInfoCART.cpp

6.18 クラス cpm_VoxelInfoLMR

`#include <cpm_VoxelInfoLMR.h>`

cpm_VoxelInfoLMR に対する継承グラフ

cpm_VoxelInfoLMR のコラボレーション図

Public メソッド

- void debugPrint ()

Protected メソッド

- cpm_VoxelInfoLMR ()
- virtual ~cpm_VoxelInfoLMR ()
- void SetGlobaliDomainInfo (S_OCT_DOMAIN_INFO &dInfo)
- void SetLocalDomainInfo (S_OCT_DOMAIN_INFO &dInfo)
- void SetNeighborInfo (const std::map< int, int > &leafIDmap)
- virtual const int * GetNeighborLeafList (cpm_FaceFlag face, int &num) const
- virtual const int * GetPeriodicLeafList (cpm_FaceFlag face, int &num) const
- virtual const int * GetNeighborRankList (cpm_FaceFlag face, int &num) const
- virtual const int * GetPeriodicRankList (cpm_FaceFlag face, int &num) const
- virtual int GetNeighborLevelDiff (cpm_FaceFlag face) const
- virtual bool IsOuterBoundary (cpm_FaceFlag face) const
- virtual bool IsInnerBoundary (cpm_FaceFlag face) const

Static Protected メソッド

- static cpm_ErrorCode Init (MPI_Comm comm, std::string treeFile, LeafMap &leafMap)
- static cpm_ErrorCode LoadOctreeFile (std::string octFile, BCMFileIO::OctHeader &header, std::vector< Pedigree > &pedigrees)
- static cpm_ErrorCode LoadOctreeHeader (std::string octFile, BCMFileIO::OctHeader &header)
- static cpm_ErrorCode LoadOctreeHeader (FILE *fp, BCMFileIO::OctHeader &header, bool &isNeedSwap)
- static std::map< int, int > GetLeafIDMap (int nRank, int numLeaf)
- static int GetNumLeaf (std::string treeFile)

Protected 変数

- BCMFileIO::OctHeader m_octHeader
  木情報ファイルのヘッダー情報
- BCMOctree * m_octree
  生成された木情報
- Node * m_node
  自ランクが担当するリーフノード
- int m_leafID
  リーフ *ID*
- const NeighborInfo * m_neighborInfo
  BCMOctree **から生成した隣接情報**

- int m_neighborLeafID_LMR [6][4]
  隣接リーフ番号 (外部境界は負の値)
- int m_periodicLeafID_LMR [6][4]
  周期境界の隣接リーフ番号
- int m_neighborRankID_LMR [6][4]
  隣接ランク番号 (外部境界は負の値)
- int m_periodicRankID_LMR [6][4]
  周期境界の隣接ランク番号
- int m_neighborLevelDiff [6]
  隣接リーフとのレベル差 *(-1/0/1)*

フレンド

- class cpm_BaseParaManager
- class cpm_ParaManagerLMR

Additional Inherited Members

6.18.1 説明

LMR 用のVOXEL 空間情報管理クラス

cpm_VoxelInfoLMR.h の 35 行で定義されています。

6.18.2 コンストラクタとデストラクタ

6.18.2.1 cpm_VoxelInfoLMR::cpm_VoxelInfoLMR ( ) [protected]

コンストラクタ

cpm_VoxelInfoLMR.cpp の 22 行で定義されています。

参照先 cpm_Base::getRankNull(), m_neighborInfo, m_neighborLeafID_LMR, m_neighborLevelDiff, m_neighborRankID_LMR, m_octree, m_periodicLeafID_LMR, と m_periodicRankID_LMR.

参照元 Init().

6.18.2.2 cpm_VoxelInfoLMR::~cpm_VoxelInfoLMR ( ) [protected], [virtual]

デストラクタ

cpm_VoxelInfoLMR.cpp の 42 行で定義されています。

参照先 m_octree.

6.18.3 関数

6.18.3.1 void cpm_VoxelInfoLMR::debugPrint ( ) [inline]

cpm_VoxelInfoLMR.h の 43 行で定義されています。

参照先 cpm_VoxelInfo::GetDivNum(), cpm_VoxelInfo::GetDivPos(), Node::getLevel(), cpm_DomainInfo::GetVoxNum(), cpm_VoxelInfo::m_globalDomainInfo, m_leafID, cpm_VoxelInfo::m_localDomainInfo, m_neighborLeafID_LMR, m_neighborRankID_LMR, m_node, m_periodicLeafID_LMR, m_periodicRankID_LMR, cpm_VoxelInfo::m_rankNo, cpm_VoxelInfo::m_voxelHeadIndex, cpm_VoxelInfo::m_voxelTailIndex, X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

6.18.3.2 std::map< int, int > cpm_VoxelInfoLMR::GetLeafIDMap (int *nRank,* int *numLeaf*) [static], [protected]

各ランクの担当リーフマップを取得

引数

in	*nRank*	並列数
in	*numLeaf*	全リーフ数

戻り値

リーフマップ (map<leafID,rankNo>)

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

参照元 Init().

6.18.3.3 const int * cpm_VoxelInfoLMR::GetNeighborLeafList (cpm_FaceFlag *face,* int & *num*) const [protected], [virtual]

指定面における自リーフの隣接リーフ番号を取得

引数

in	*face*	面方向
out	*num*	面の数 (CART のとき 1)

戻り値

指定面における自リーフの隣接リーフ番号整数配列のポインタ

cpm_VoxelInfoLMR.cpp の 545 行で定義されています。

参照先 m_neighborLeafID_LMR.

参照元 cpm_ParaManagerLMR::GetNeighborLeafList(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.18.3.4 int cpm_VoxelInfoLMR::GetNeighborLevelDiff (cpm_FaceFlag *face*) const [protected], [virtual]

指定面におけるレベル差を取得

引数

in	*face*	面方向

戻り値

レベル差 (0:同じレベル, 1:fine, -1:coarse)

cpm_VoxelInfoLMR.cpp の 597 行で定義されています。

参照先 m_neighborLevelDiff.

参照元 cpm_ParaManagerLMR::GetNeighborLevelDiff(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.18.3.5 const int * cpm_VoxelInfoLMR::GetNeighborRankList (cpm_FaceFlag *face,* int & *num*) const [protected], [virtual]

指定面における自リーフの隣接ランク番号を取得

引数

in	*face*	面方向
out	*num*	面の数 (CART のとき 1)

戻り値

指定面における自リーフの隣接ランク番号整数配列のポインタ

cpm_VoxelInfoLMR.cpp の 571 行で定義されています。

参照先 cpm_Base::IsRankNull(), と m_neighborRankID_LMR.

参照元 cpm_ParaManagerLMR::GetNeighborRankList(), IsInnerBoundary(), IsOuterBoundary(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.18.3.6 int cpm_VoxelInfoLMR::GetNumLeaf (std::string *treeFile*) [static], [protected]

木情報ファイルからリーフ数を取得する

引数

in	*treefile*	木情報ファイル

戻り値

リーフ数

cpm_VoxelInfoLMR.cpp の 522 行で定義されています。

参照先 CPM_SUCCESS, LoadOctreeHeader(), BCMFileIO::OctHeader::numLeaf, S_OCT_DOMAIN_INFO::octFile, と cpm_TextParserDomainLMR::Read().

参照元 cpm_ParaManagerLMR::GetNumLeaf().

6.18.3.7 const int * cpm_VoxelInfoLMR::GetPeriodicLeafList (cpm_FaceFlag *face,* int & *num*) const [protected], [virtual]

指定面における自リーフの周期境界の隣接リーフ番号を取得

引数

in	*face*	面方向
out	*num*	面の数 (CART のとき 1)

戻り値

指定面における自リーフの周期境界の隣接リーフ番号整数配列のポインタ

cpm_VoxelInfoLMR.cpp の 558 行で定義されています。

参照先 m_periodicLeafID_LMR.

参照元 cpm_ParaManagerLMR::GetPeriodicLeafList(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.18.3.8 const int * cpm_VoxelInfoLMR::GetPeriodicRankList (cpm_FaceFlag *face,* int & *num*) const [protected], [virtual]

指定面における自リーフの周期境界の隣接ランク番号を取得

引数

in	*face*	面方向
out	*num*	面の数 (CART のとき 1)

戻り値

指定面における自リーフの周期境界の隣接ランク番号整数配列のポインタ

cpm_VoxelInfoLMR.cpp の 584 行で定義されています。

参照先 cpm_Base::IsRankNull(), と m_periodicRankID_LMR.

参照元 cpm_ParaManagerLMR::GetPeriodicRankList(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.18.3.9 cpm_ErrorCode cpm_VoxelInfoLMR::Init (MPI_Comm *comm,* std::string *treeFile,* LeafMap & *leafMap*) [static], [protected]

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	*comm*	MPI コミュニケータ
in	*treeFile*	領域情報ファイル
out	*leafMap*	リーフごとのVoxel 空間情報マップ

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_VoxelInfoLMR.cpp の 50 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_LMR_INVALID_OCTFILE, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, cpm_VoxelInfoLMR(), Node::getBlockID(), GetLeafIDMap(), BCMOctree::getLeafNodeArray(), BCMOctree::getOrigin(), Node::getPedigree(), cpm_Base::IsCommNull(), LoadOctreeFile(), cpm_VoxelInfo::m_comm, m_leafID, m_node, cpm_VoxelInfo::m_nRank, m_octHeader, m_octree, cpm_VoxelInfo::m_rankNo, BCMFileIO::OctHeader::maxLevel, BCMFileIO::OctHeader::numLeaf, S_OCT_DOMAIN_INFO::octFile, BCMFileIO::OctHeader::org, BCMFileIO::OctHeader::padding, S_OCT_DOMAIN_INFO::print(), cpm_TextParserDomainLMR::Read(), BCMFileIO::OctHeader::rgn, BCMFileIO::OctHeader::rootDims, SetGlobaliDomainInfo(), SetLocalDomainInfo(), と SetNeighborInfo().

参照元 cpm_ParaManagerLMR::VoxelInit_LMR().

6.18.3.10 bool cpm_VoxelInfoLMR::IsInnerBoundary (cpm_FaceFlag *face*) const [protected], [virtual]

自リーフの境界が内部境界 (隣が不活性ドメイン) かどうかを判定

引数

in	*face*	面方向

戻り値

true	内部境界
false	内部境界でない

cpm_VoxelInfoを再定義しています。

cpm_VoxelInfoLMR.cpp の 634 行で定義されています。

参照先 cpm_VoxelInfo::GetDivNum(), cpm_VoxelInfo::GetDivPos(), GetNeighborRankList(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManagerLMR::IsInnerBoundary().

6.18.3.11 bool cpm_VoxelInfoLMR::IsOuterBoundary (cpm_FaceFlag *face*) const [protected], [virtual]

自リーフの境界が外部境界かどうかを判定

引数

in	*face*	面方向

戻り値

true	外部境界
false	外部境界でない

cpm_VoxelInfoを再定義しています。

cpm_VoxelInfoLMR.cpp の 605 行で定義されています。

参照先 cpm_VoxelInfo::GetDivNum(), cpm_VoxelInfo::GetDivPos(), GetNeighborRankList(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManagerLMR::IsOuterBoundary().

6.18.3.12 cpm_ErrorCode cpm_VoxelInfoLMR::LoadOctreeFile (std::string *octFile*, BCMFileIO::OctHeader & *header*, std::vector< Pedigree > & *pedigrees*) [static], [protected]

木情報ファイルの読み込み

引数

in	*octFile*	木情報ファイル
out	*header*	ヘッダー情報
out	*pedigrees*	ペディグリー情報

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 BCMFileIO::BSwap64(), CPM_ERROR_LMR_OPEN_OCTFILE, CPM_SUCCESS, LoadOctreeHeader(), と BCMFileIO::OctHeader::numLeaf.

参照元 Init().

6.18.3.13 cpm_ErrorCode cpm_VoxelInfoLMR::LoadOctreeHeader (std::string *octFile*, BCMFileIO::OctHeader & *header*) [static], [protected]

木情報ファイルのヘッダー読み込み

- ヘッダーのみを読み込み、ファイルをクローズする

引数

in	*octFile*	木情報ファイル
out	*header*	ヘッダー情報

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ERROR_LMR_OPEN_OCTFILE, と CPM_SUCCESS.

参照元 GetNumLeaf(), と LoadOctreeFile().

6.18.3.14 cpm_ErrorCode cpm_VoxelInfoLMR::LoadOctreeHeader (FILE * *fp,* BCMFileIO::OctHeader & *header,* bool & *isNeedSwap*) [static], [protected]

木情報ファイルのヘッダー読み込み

引数

in	*fp*	木情報ファイルポインタ
out	*header*	ヘッダー情報
out	*isNeedSwap*	エンディアン変換フラグ (true:要変換)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_VoxelInfoLMR.cpp の 267 行で定義されています。

参照先 BCMFileIO::BSwap32(), BCMFileIO::BSwap64(), CPM_ERROR_LMR_INVALID_OCTFILE, CPM_ERROR_LMR_OPEN_OCTFILE, CPM_SUCCESS, BCMFileIO::OctHeader::identifier, BCMFileIO::OctHeader::maxLevel, BCMFileIO::OctHeader::numLeaf, OCTREE_FILE_IDENTIFIER, BCMFileIO::OctHeader::org, BCMFileIO::OctHeader::rgn, と BCMFileIO::OctHeader::rootDims.

6.18.3.15 void cpm_VoxelInfoLMR::SetGlobaliDomainInfo (S_OCT_DOMAIN_INFO & *dInfo*) [protected]

グローバルの領域情報をセット

引数

in	*dInfo*	領域情報

cpm_VoxelInfoLMR.cpp の 348 行で定義されています。

参照先 Node::getLevel(), cpm_VoxelInfo::m_globalDomainInfo, m_node, m_octHeader, S_OCT_DOMAIN_INFO::origin, S_OCT_DOMAIN_INFO::region, BCMFileIO::OctHeader::rootDims, cpm_GlobalDomainInfo::SetDivNum(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), と S_OCT_DOMAIN_INFO::size.

参照元 Init().

6.18.3.16 void cpm_VoxelInfoLMR::SetLocalDomainInfo (S_OCT_DOMAIN_INFO & *dInfo*) [protected]

ローカルの領域情報をセット

引数

in	*dInfo*	領域情報

cpm_VoxelInfoLMR.cpp の 377 行で定義されています。

参照先 Node::getLevel(), cpm_DomainInfo::GetOrigin(), BCMOctree::getOrigin(), Node::getPedigree(), cpm_DomainInfo::GetRegion(), BCMOctree::getRootGrid(), Pedigree::getRootID(), Pedigree::getUpperBound(), cpm_DomainInfo::GetVoxNum(), Pedigree::getX(), Pedigree::getY(), Pedigree::getZ(), cpm_VoxelInfo::m_globalDomainInfo, cpm_VoxelInfo::m_localDomainInfo, m_node, m_octHeader, m_octree, cpm_VoxelInfo::m_voxelHeadIndex, cpm_VoxelInfo::m_voxelTailIndex, BCMFileIO::OctHeader::rootDims, RootGrid::rootID2indexX(), RootGrid::rootID2indexY(), RootGrid::rootID2indexZ(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_ActiveSubdomainInfo::SetPos(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), と S_OCT_DOMAIN_INFO::size.

参照元 Init().

6.18.3.17 void cpm_VoxelInfoLMR::SetNeighborInfo (const std::map< int, int > & *leafIDmap*) [protected]

隣接情報の取得

cpm_VoxelInfoLMR.cpp の 447 行で定義されています。

参照先 RootGrid::clearPeriodicX(), RootGrid::clearPeriodicY(), RootGrid::clearPeriodicZ(), NeighborInfo::getID(), NeighborInfo::getLevelDifference(), BCMOctree::getRootGrid(), cpm_Base::IsRankNull(), m_neighborLeafID_LMR, m_neighborLevelDiff, cpm_VoxelInfo::m_neighborRankID, m_neighborRankID_LMR, m_node, cpm_VoxelInfo::m_nRank, m_octHeader, m_octree, m_periodicLeafID_LMR, cpm_VoxelInfo::m_periodicRankID, m_periodicRankID_LMR, BCMOctree::makeNeighborInfo(), BCMFileIO::OctHeader::numLeaf, RootGrid::setPeriodicX(), RootGrid::setPeriodicY(), RootGrid::setPeriodicZ(), X_M, X_MINUS, X_P, X_PLUS, Y_M, Y_MINUS, Y_P, Y_PLUS, Z_M, Z_MINUS, Z_P, と Z_PLUS.

参照元 Init().

6.18.4 フレンドと関連する関数

6.18.4.1 friend class cpm_BaseParaManager [friend]

cpm_VoxelInfoLMR.h の 37 行で定義されています。

6.18.4.2 friend class cpm_ParaManagerLMR [friend]

cpm_VoxelInfoLMR.h の 38 行で定義されています。

6.18.5 変数

6.18.5.1 int cpm_VoxelInfoLMR::m_leafID [protected]

リーフID

cpm_VoxelInfoLMR.h の 242 行で定義されています。

参照元 debugPrint(), と Init().

6.18.5.2 const NeighborInfo* cpm_VoxelInfoLMR::m_neighborInfo [protected]

BCMOctree から生成した隣接情報

cpm_VoxelInfoLMR.h の 245 行で定義されています。

参照元 cpm_VoxelInfoLMR().

6.18.5.3 int cpm_VoxelInfoLMR::m_neighborLeafID_LMR[6][4] [protected]

隣接リーフ番号 (外部境界は負の値)

cpm_VoxelInfoLMR.h の 246 行で定義されています。

参照元 cpm_VoxelInfoLMR(), debugPrint(), GetNeighborLeafList(), と SetNeighborInfo().

6.18.5.4 int cpm_VoxelInfoLMR::m_neighborLevelDiff[6] [protected]

隣接リーフとのレベル差 (-1/0/1)

cpm_VoxelInfoLMR.h の 250 行で定義されています。

参照元 cpm_VoxelInfoLMR(), GetNeighborLevelDiff(), と SetNeighborInfo().

6.18.5.5 int cpm_VoxelInfoLMR::m_neighborRankID_LMR[6][4] [protected]

隣接ランク番号 (外部境界は負の値)

cpm_VoxelInfoLMR.h の 248 行で定義されています。

参照元 cpm_VoxelInfoLMR(), debugPrint(), GetNeighborRankList(), と SetNeighborInfo().

6.18.5.6 Node* cpm_VoxelInfoLMR::m_node [protected]

自ランクが担当するリーフノード

cpm_VoxelInfoLMR.h の 241 行で定義されています。

参照元 debugPrint(), Init(), SetGlobaliDomainInfo(), SetLocalDomainInfo(), と SetNeighborInfo().

6.18.5.7 BCMFileIO::OctHeader cpm_VoxelInfoLMR::m_octHeader [protected]

木情報ファイルのヘッダー情報

cpm_VoxelInfoLMR.h の 239 行で定義されています。

参照元 Init(), SetGlobaliDomainInfo(), SetLocalDomainInfo(), と SetNeighborInfo().

6.18.5.8 BCMOctree* cpm_VoxelInfoLMR::m_octree [protected]

生成された木情報

cpm_VoxelInfoLMR.h の 240 行で定義されています。

参照元 cpm_VoxelInfoLMR(), Init(), SetLocalDomainInfo(), SetNeighborInfo(), と ~cpm_VoxelInfoLMR().

6.18.5.9 int cpm_VoxelInfoLMR::m_periodicLeafID_LMR[6][4] [protected]

周期境界の隣接リーフ番号

cpm_VoxelInfoLMR.h の 247 行で定義されています。

参照元 cpm_VoxelInfoLMR(), debugPrint(), GetPeriodicLeafList(), と SetNeighborInfo().

6.18.5.10 int cpm_VoxelInfoLMR::m_periodicRankID_LMR[6][4] [protected]

周期境界の隣接ランク番号

cpm_VoxelInfoLMR.h の 249 行で定義されています。

参照元 cpm_VoxelInfoLMR(), debugPrint(), GetPeriodicRankList(), と SetNeighborInfo().

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_VoxelInfoLMR.h
- cpm_VoxelInfoLMR.cpp

6.19 クラス Divider

ブロック分割判定クラス (基底クラス).

`#include <Divider.h>`

Public 型

- enum NodeType { BRANCH, LEAF_ACTIVE, LEAF_NO_ACTIVE }
  ブロック (ノード) タイプ型

Public メソッド

- Divider ()
  コンストラクタ.
- virtual ~Divider ()
  デストラクタ.
- virtual NodeType operator() (const Pedigree &pedigree)=0

6.19.1 説明

ブロック分割判定クラス (基底クラス).

Divider.h の 24 行で定義されています。

6.19.2 列挙型

6.19.2.1 enum Divider::NodeType

ブロック (ノード) タイプ型

列挙型の値

BRANCH 枝 (分割を続ける)

LEAF_ACTIVE アクティブなリーフノード (分割を終了)

LEAF_NO_ACTIVE 非アクティブなリーフノード (分割を終了)

Divider.h の 29 行で定義されています。

6.19.3 コンストラクタとデストラクタ

6.19.3.1 Divider::Divider ( ) [inline]

コンストラクタ.

Divider.h の 38 行で定義されています。

6.19.3.2 virtual Divider::~Divider ( ) [inline], [virtual]

デストラクタ.

Divider.h の 41 行で定義されています。

6.19.4 関数

6.19.4.1 virtual NodeType Divider::operator() (const Pedigree & *pedigree*) [pure virtual]

ブロックを分割するかどうかを判定.

引数

in	*pedigree*	ブロックのPedigree

戻り値

ブロックタイプ

このクラスの説明は次のファイルから生成されました:

- Divider.h

6.20 構造体 BCMFileIO::GridRleCode

RLE 圧縮符号の走査用構造体

`#include <BCMFileCommon.h>`

Public 変数

- bitVoxelCell c
  データ
- unsigned char len
  ラン長

6.20.1 説明

RLE 圧縮符号の走査用構造体

BCMFileCommon.h の 87 行で定義されています。

6.20.2 変数

6.20.2.1 bitVoxelCell BCMFileIO::GridRleCode::c

データ

BCMFileCommon.h の 89 行で定義されています。

6.20.2.2 unsigned char BCMFileIO::GridRleCode::len

ラン長

BCMFileCommon.h の 90 行で定義されています。

この構造体の説明は次のファイルから生成されました:

- BCMFileCommon.h

6.21 構造体 BCMFileIO::IdxProc

インデックスファイル用プロセス情報

`#include <BCMFileCommon.h>`

Public 変数

- std::string hostname
  ホスト名
- unsigned int rank
  ランク番号
- unsigned int rangeMin
  ブロック*ID* のレンジ最小値
- unsigned int rangeMax
  ブロック*ID* のレンジ最大値

6.21.1 説明

インデックスファイル用プロセス情報

BCMFileCommon.h の 137 行で定義されています。

6.21.2 変数

6.21.2.1 std::string BCMFileIO::IdxProc::hostname

ホスト名

BCMFileCommon.h の 139 行で定義されています。

6.21.2.2 unsigned int BCMFileIO::IdxProc::rangeMax

ブロックID のレンジ最大値

BCMFileCommon.h の 142 行で定義されています。

6.21.2.3 unsigned int BCMFileIO::IdxProc::rangeMin

ブロックID のレンジ最小値

BCMFileCommon.h の 141 行で定義されています。

6.21.2.4 unsigned int BCMFileIO::IdxProc::rank

ランク番号

BCMFileCommon.h の 140 行で定義されています。

この構造体の説明は次のファイルから生成されました:

- BCMFileCommon.h

6.22 構造体 BCMFileIO::IdxUnit

インデックスファイル用単位系情報

`#include <BCMFileCommon.h>`

Public 変数

- std::string length
  長さ単位 *(NonDimensional, m, cm, mm)*
- double L0_scale
  規格化に用いたスケール *(***単位**:**指定単位***)*
- std::string velocity
  時間単位 *(NonDimensional, Dimensional)*
- double V0_scale
  規格化に用いた時間スケール *(***単位***:Dimensional* **の場合** *m/s)*

6.22.1 説明

インデックスファイル用単位系情報

BCMFileCommon.h の 128 行で定義されています。

6.22.2 変数

6.22.2.1 double BCMFileIO::IdxUnit::L0_scale

規格化に用いたスケール (単位:指定単位)

BCMFileCommon.h の 131 行で定義されています。

6.22.2.2 std::string BCMFileIO::IdxUnit::length

長さ単位 (NonDimensional, m, cm, mm)

BCMFileCommon.h の 130 行で定義されています。

6.22.2.3 double BCMFileIO::IdxUnit::V0_scale

規格化に用いた時間スケール (単位:Dimensional の場合 m/s)

BCMFileCommon.h の 133 行で定義されています。

6.22.2.4 std::string BCMFileIO::IdxUnit::velocity

時間単位 (NonDimensional, Dimensional)

BCMFileCommon.h の 132 行で定義されています。

この構造体の説明は次のファイルから生成されました:

- BCMFileCommon.h

6.23 構造体 BCMFileIO::LBCellIDHeader

LeafBlock のCellID ヘッダ構造体

`#include <BCMFileCommon.h>`

Public 変数

- uint64_t numBlock

  ブロック数
- uint64_t compSize

  圧縮符号サイズ (バイト単位)

6.23.1 説明

LeafBlock のCellID ヘッダ構造体

BCMFileCommon.h の 77 行で定義されています。

6.23.2 変数

6.23.2.1 uint64_t BCMFileIO::LBCellIDHeader::compSize

圧縮符号サイズ (バイト単位)

BCMFileCommon.h の 80 行で定義されています。

6.23.2.2 uint64_t BCMFileIO::LBCellIDHeader::numBlock

ブロック数

BCMFileCommon.h の 79 行で定義されています。

この構造体の説明は次のファイルから生成されました:

- BCMFileCommon.h

6.24 構造体 BCMFileIO::LBHeader

LeafBlock ファイルヘッダ構造体

`#include <BCMFileCommon.h>`

Public 変数

- unsigned int identifier
  エンディアン識別子
- unsigned char kind
  ブロックファイル種類
- unsigned char dataType
  1 **セルあたりのサイズ**
- unsigned short bitWidth
  1 **セルあたりのビット幅**
- unsigned int vc
  仮想セルサイズ
- unsigned int size [3]
  ブロックサイズ
- uint64_t numBlock
  ファイルに記載されている総ブロック数

6.24.1 説明

LeafBlock ファイルヘッダ構造体

BCMFileCommon.h の 64 行で定義されています。

6.24.2 変数

6.24.2.1 unsigned short BCMFileIO::LBHeader::bitWidth

1 セルあたりのビット幅

BCMFileCommon.h の 69 行で定義されています。

6.24.2.2 unsigned char BCMFileIO::LBHeader::dataType

1 セルあたりのサイズ

BCMFileCommon.h の 68 行で定義されています。

6.24.2.3 unsigned int BCMFileIO::LBHeader::identifier

エンディアン識別子

BCMFileCommon.h の 66 行で定義されています。

6.24.2.4 unsigned char BCMFileIO::LBHeader::kind

ブロックファイル種類

BCMFileCommon.h の 67 行で定義されています。

6.24.2.5 uint64_t BCMFileIO::LBHeader::numBlock

ファイルに記載されている総ブロック数

BCMFileCommon.h の 72 行で定義されています。

6.24.2.6 unsigned int BCMFileIO::LBHeader::size[3]

ブロックサイズ

BCMFileCommon.h の 71 行で定義されています。

6.24.2.7 unsigned int BCMFileIO::LBHeader::vc

仮想セルサイズ

BCMFileCommon.h の 70 行で定義されています。

この構造体の説明は次のファイルから生成されました:

- BCMFileCommon.h

6.25 クラス NeighborInfo

隣接情報クラス.

`#include <NeighborInfo.h>`

Public メソッド

- NeighborInfo ()
- ~NeighborInfo ()
  デストラクタ.
- void setLevelDifference (int dLevel)
  レベル差を設定.
- int getLevelDifference () const
  レベル差を取得.
- void setID (int id)
  隣接ブロック*ID* を設定.
- void setID (Subface subface, int id)
  隣接ブロック*ID* を設定.
- int getID () const
  隣接ブロック*ID* を取得.
- int getID (Subface subface) const
  隣接ブロック*ID* を取得.
- void setRank (int rank)
  隣接ブロックランクを設定.
- void setRank (Subface subface, int rank)
  隣接ブロックランクを設定.
- int getRank () const
  隣接ブロックランクを取得.
- int getRank (Subface subface) const
  隣接ブロックランクを取得.

- void setNeighborSubface (Subface subface)
  隣接ブロックのサブフェイス番号を設定.
- Subface getNeighborSubface () const
  隣接ブロックのサブフェイス番号を取得.
- void setOuterBoundary (bool flag=true)
  外部境界フラグを *(***オンに***)* **設定**.
- bool isOuterBoundary () const
  外部境界であるか確認.
- bool exists () const
  隣接ブロックが存在するか *(***内部境界** *or* **周期境界***)* **確認**.
- void print () const
  デバッグ情報出力.

Static Public メソッド

- static int getNeighborChildId (Face face, Subface subface)
  指定した隣接面を含む隣接ブロックにおける子ブロック番号を取得.
- static Subface childIdToSubface (Face face, int childId)
  指定した接触面における子ブロックの*Subface* **番号を取得**.
- static Face reverseFace (Face face)
  Subface **番号を対面ブロックのものに変換**.

Private 変数

- int neighborID [NUM_SUBFACE]
- int neighborRank [NUM_SUBFACE]
- bool outerBoundary
- int levelDiffarence
- int neighborSubface

6.25.1 説明

隣接情報クラス.

NeighborInfo.h の 30 行で定義されています。

6.25.2 コンストラクタとデストラクタ

6.25.2.1 NeighborInfo::NeighborInfo ( ) [inline]

コンストラクタ.

覚え書き

初期値は, neighborID[i]=-1(隣接ブロックなし), neighborRank[i]=MPI::PROC_NULL(隣接ブロックなし), outBoundary=false(内部境界), levelDiffarence=0(レベル差なし), neighborSubface=0(サブブロック 0 と隣接)

NeighborInfo.h の 62 行で定義されています。

参照先 NUM_SUBFACE.

6.25.2.2 NeighborInfo::~NeighborInfo ( ) [inline]

デストラクタ.

NeighborInfo.h の 73 行で定義されています。

6.25.3 関数

6.25.3.1 static Subface NeighborInfo::childIdToSubface (Face *face,* int *childId*) [inline], [static]

指定した接触面における子ブロックのSubface 番号を取得.

NeighborInfo.h の 190 行で定義されています。

参照先 EX_FAILURE, Exit, X_M, X_P, Y_M, Y_P, Z_M, と Z_P.

参照元 BCMOctree::makeNeighborInfo().

6.25.3.2 bool NeighborInfo::exists ( ) const [inline]

隣接ブロックが存在するか (内部境界 or 周期境界) 確認.

NeighborInfo.h の 153 行で定義されています。

6.25.3.3 int NeighborInfo::getID ( ) const [inline]

隣接ブロックID を取得.

NeighborInfo.h の 97 行で定義されています。

参照元 cpm_VoxelInfoLMR::SetNeighborInfo().

6.25.3.4 int NeighborInfo::getID (Subface *subface*) const [inline]

隣接ブロックID を取得.

NeighborInfo.h の 103 行で定義されています。

6.25.3.5 int NeighborInfo::getLevelDifference ( ) const [inline]

レベル差を取得.

NeighborInfo.h の 82 行で定義されています。

参照元 cpm_VoxelInfoLMR::SetNeighborInfo().

6.25.3.6 static int NeighborInfo::getNeighborChildId (Face *face,* Subface *subface*) [inline], [static]

指定した隣接面を含む隣接ブロックにおける子ブロック番号を取得.

NeighborInfo.h の 170 行で定義されています。

参照先 EX_FAILURE, Exit, X_M, X_P, Y_M, Y_P, Z_M, と Z_P.

参照元 BCMOctree::makeNeighborInfo().

6.25.3.7 Subface NeighborInfo::getNeighborSubface ( ) const [inline]

隣接ブロックのサブフェイス番号を取得.

NeighborInfo.h の 137 行で定義されています。

6.25.3.8 int NeighborInfo::getRank ( ) const [inline]

隣接ブロックランクを取得.

NeighborInfo.h の 120 行で定義されています。

6.25.3.9 int NeighborInfo::getRank (Subface *subface*) const [inline]

隣接ブロックランクを取得.

NeighborInfo.h の 126 行で定義されています。

6.25.3.10 bool NeighborInfo::isOuterBoundary ( ) const [inline]

外部境界であるか確認.

NeighborInfo.h の 148 行で定義されています。

6.25.3.11 void NeighborInfo::print ( ) const [inline]

デバッグ情報出力.

NeighborInfo.h の 158 行で定義されています。

参照先 NUM_SUBFACE.

6.25.3.12 static Face NeighborInfo::reverseFace (Face *face*) [inline], [static]

Subface 番号を対面ブロックのものに変換.

NeighborInfo.h の 208 行で定義されています。

参照先 EX_FAILURE, Exit, X_M, X_P, Y_M, Y_P, Z_M, と Z_P.

6.25.3.13 void NeighborInfo::setID (int *id*) [inline]

隣接ブロックID を設定.

NeighborInfo.h の 85 行で定義されています。

参照元 BCMOctree::makeNeighborInfo().

6.25.3.14 void NeighborInfo::setID (Subface *subface,* int *id*) [inline]

隣接ブロックID を設定.

NeighborInfo.h の 91 行で定義されています。

6.25.3.15 void NeighborInfo::setLevelDifference (int *dLevel*) [inline]

レベル差を設定.

NeighborInfo.h の 76 行で定義されています。

参照元 BCMOctree::makeNeighborInfo().

6.25.3.16 void NeighborInfo::setNeighborSubface (Subface *subface*) [inline]

隣接ブロックのサブフェイス番号を設定.
NeighborInfo.h の 131 行で定義されています。
参照元 BCMOctree::makeNeighborInfo().

6.25.3.17 void NeighborInfo::setOuterBoundary (bool *flag* = true) [inline]

外部境界フラグを (オンに) 設定.
NeighborInfo.h の 143 行で定義されています。

6.25.3.18 void NeighborInfo::setRank (int *rank*) [inline]

隣接ブロックランクを設定.
NeighborInfo.h の 108 行で定義されています。
参照元 BCMOctree::makeNeighborInfo().

6.25.3.19 void NeighborInfo::setRank (Subface *subface,* int *rank*) [inline]

隣接ブロックランクを設定.
NeighborInfo.h の 114 行で定義されています。

6.25.4 変数

6.25.4.1 int NeighborInfo::levelDiffarence [private]

隣接ブロックとのレベル差 (-1, 0, +1). (隣のレベル - 自分のレベル)
NeighborInfo.h の 46 行で定義されています。

6.25.4.2 int NeighborInfo::neighborID[NUM_SUBFACE] [private]

隣接ブロックID. (隣接ブロックが存在しない場合は-1 を入れる)
NeighborInfo.h の 34 行で定義されています。

6.25.4.3 int NeighborInfo::neighborRank[NUM_SUBFACE] [private]

隣接ブロック所属ランク. (周期境界以外の外部境界面の場合はMPI::PROC_NULL を入れる)
NeighborInfo.h の 38 行で定義されています。

6.25.4.4 int NeighborInfo::neighborSubface [private]

隣接する相手のサブブロック番号. (levelDiffarence = -1 以外の場合は 0 を入れる)
NeighborInfo.h の 50 行で定義されています。

6.25.4.5 bool NeighborInfo::outerBoundary [private]

外部境界フラグ. (周期境界の場合も true)

NeighborInfo.h の 42 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:

- NeighborInfo.h

6.26 クラス Node

Octree ノードクラス.

`#include <Node.h>`

Node のコラボレーション図

Public メソッド

- Node (int rootID=0)
  コンストラクタ *(***ルートノードとして生成***).*
- Node (Node *parent, int i)
- ~Node ()
  デストラクタ.
- bool isRootNode () const
- bool isLeafNode () const
- bool isActive () const
- void setActive (bool OnOff=true)
- int getBlockID () const
- void setBlockID (int id)
- const Pedigree & getPedigree () const
- int getLevel () const
- void makeChildNodes ()
  8 **つの子ノードを生成**.
- Vec3d getBlockSize () const
- Node * getParent ()
- Node * getChild (int i)

Private 変数

- Node * parent
  親ノードへのポインタ
- Node ** childList
  子ノードリスト
- bool active
  アクティブノードフラグ
- int id
  ブロック*ID(***アクティブなリーフノード以外には***-1* **を入れる***)*
- Pedigree pedigree
  Pedigree.

6.26.1 説明

Octree ノードクラス.

Node.h の 26 行で定義されています。

6.26.2 コンストラクタとデストラクタ

6.26.2.1 Node::Node (int *rootID* = 0) [inline]

コンストラクタ (ルートノードとして生成).

Node.h の 41 行で定義されています。

6.26.2.2 Node::Node (Node * *parent*, int *i*) [inline]

コンストラクタ (子ノードとして生成).

引数

in	*parent*	親ノード
in	*i*	子ノード番号 (0〜7)

Node.h の 49 行で定義されています。

6.26.2.3 Node::~Node ( ) [inline]

デストラクタ.

Node.h の 53 行で定義されています。

6.26.3 関数

6.26.3.1 int Node::getBlockID ( ) const [inline]

ブロックID を取得.

戻り値

ブロックID

Node.h の 88 行で定義されています。

参照元 cpm_VoxelInfoLMR::Init(), と BCMOctree::makeNeighborInfo().

6.26.3.2 Vec3d Node::getBlockSize ( ) const [inline]

規格化されたブロックサイズを計算.

戻り値

ブロックサイズ

Node.h の 119 行で定義されています。

6.26.3.3 Node* Node::getChild (int *i*) [inline]

子ノードを所得.

引数

in	*i*	子ノード番号 (0〜7)

戻り値

子ノードへのポインタ

Node.h の 135 行で定義されています。

参照元 BCMOctree::BCMOctree(), BCMOctree::buildTreeFromPedigreeList(), BCMOctree::deleteNode(), BCMOctree::findNeighborNode(), BCMOctree::makeNeighborInfo(), BCMOctree::makeNode(), BCMOctree::packPedigrees(), BCMOctree::pickupLeafNodeHilbertOrdering(), と BCMOctree::pickupLeafNodeZOrdering().

6.26.3.4 int Node::getLevel ( ) const [inline]

ツリーレベルを取得.

戻り値

ツリーレベル

Node.h の 106 行で定義されています。

参照元 cpm_VoxelInfoLMR::debugPrint(), BCMOctree::makeNeighborInfo(), cpm_VoxelInfoLMR::SetGlobalDomainInfo(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.26.3.5 Node* Node::getParent ( ) [inline]

親ノードを取得.

戻り値

親ノードへのポインタ

Node.h の 128 行で定義されています。

6.26.3.6 const Pedigree& Node::getPedigree ( ) const [inline]

Pedigree を取得.

戻り値

Pedigree

Node.h の 100 行で定義されています。

参照元 BCMOctree::checkOnOuterBoundary(), BCMOctree::findNeighborNode(), BCMOctree::getOrigin(), cpm_VoxelInfoLMR::Init(), BCMOctree::makeNeighborInfo(), BCMOctree::makeNode(), BCMOctree::packPedigrees(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.26.3.7 bool Node::isActive ( ) const [inline]

アクティブノード判定.

戻り値

アクティブノードの場合 true

Node.h の 76 行で定義されています。

参照元 BCMOctree::makeNeighborInfo(), BCMOctree::packPedigrees(), BCMOctree::pickupLeafNodeHilbertOrdering(), と BCMOctree::pickupLeafNodeZOrdering().

6.26.3.8 bool Node::isLeafNode ( ) const [inline]

リーフノード判定.

戻り値

リーフノードの場合 true

Node.h の 70 行で定義されています。

参照元 BCMOctree::BCMOctree(), BCMOctree::buildTreeFromPedigreeList(), BCMOctree::deleteNode(), BCMOctree::findNeighborNode(), BCMOctree::makeNeighborInfo(), BCMOctree::packPedigrees(), BCMOctree::pickupLeafNodeHilbertOrdering(), と BCMOctree::pickupLeafNodeZOrdering().

6.26.3.9 bool Node::isRootNode ( ) const [inline]

ルートノード判定.

戻り値

ルートノードの場合 true

Node.h の 64 行で定義されています。

6.26.3.10 void Node::makeChildNodes ( ) [inline]

8 つの子ノードを生成.

Node.h の 109 行で定義されています。

参照元 BCMOctree::BCMOctree(), BCMOctree::buildTreeFromPedigreeList(), と BCMOctree::makeNode().

6.26.3.11 void Node::setActive (bool *OnOff* = true) [inline]

アクティブノードフラグの設定.

引数

in	*OnOff*	アクティブノードフラグ値

Node.h の 82 行で定義されています。

参照元 BCMOctree::BCMOctree(), BCMOctree::buildTreeFromPedigreeList(), と BCMOctree::makeNode().

6.26.3.12 void Node::setBlockID (int *id*) [inline]

ブロックID を設定.

引数

in	*id*	ブロックID

Node.h の 94 行で定義されています。

参照元 BCMOctree::BCMOctree(), BCMOctree::pickupLeafNodeHilbertOrdering(), と BCMOctree::pickupLeafNodeZOrdering().

6.26.4 変数

6.26.4.1 bool Node::active [private]

アクティブノードフラグ

Node.h の 32 行で定義されています。

6.26.4.2 Node** Node::childList [private]

子ノードリスト

Node.h の 30 行で定義されています。

6.26.4.3 int Node::id [private]

ブロックID(アクティブなリーフノード以外には-1 を入れる)

Node.h の 34 行で定義されています。

6.26.4.4 Node* Node::parent [private]

親ノードへのポインタ

Node.h の 28 行で定義されています。

6.26.4.5 Pedigree Node::pedigree [private]

Pedigree.

Node.h の 36 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:

- Node.h

6.27 構造体 BCMFileIO::OctHeader

Octree ファイルヘッダ構造体

`#include <BCMFileCommon.h>`

Public メソッド

- OctHeader ()

Public 変数

- unsigned int identifier
  エンディアン識別子
- double org [3]
  原点座標
- double rgn [3]
  領域サイズ
- unsigned int rootDims [3]
  ルート分割数
- unsigned int maxLevel
  Octree **最大分割レベル**
- uint64_t numLeaf
  リーフノード数
- uint64_t padding
  16 **バイトアライメント用パディング**

6.27.1 説明

Octree ファイルヘッダ構造体

BCMFileCommon.h の 49 行で定義されています。

6.27.2 コンストラクタとデストラクタ

6.27.2.1 BCMFileIO::OctHeader::OctHeader ( ) [inline]

BCMFileCommon.h の 59 行で定義されています。

6.27.3 変数

6.27.3.1 unsigned int BCMFileIO::OctHeader::identifier

エンディアン識別子

BCMFileCommon.h の 51 行で定義されています。

参照元 cpm_VoxelInfoLMR::LoadOctreeHeader().

6.27.3.2 unsigned int BCMFileIO::OctHeader::maxLevel

Octree 最大分割レベル

BCMFileCommon.h の 55 行で定義されています。

参照元 cpm_VoxelInfoLMR::Init(), と cpm_VoxelInfoLMR::LoadOctreeHeader().

6.27.3.3 uint64_t BCMFileIO::OctHeader::numLeaf

リーフノード数

BCMFileCommon.h の 56 行で定義されています。

参照元 cpm_VoxelInfoLMR::GetNumLeaf(), cpm_VoxelInfoLMR::Init(), cpm_VoxelInfoLMR::LoadOctreeFile(), cpm_VoxelInfoLMR::LoadOctreeHeader(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.27.3.4 double BCMFileIO::OctHeader::org[3]

原点座標

BCMFileCommon.h の 52 行で定義されています。

参照元 cpm_VoxelInfoLMR::Init(), と cpm_VoxelInfoLMR::LoadOctreeHeader().

6.27.3.5 uint64_t BCMFileIO::OctHeader::padding

16 バイトアライメント用パディング

BCMFileCommon.h の 57 行で定義されています。

参照元 cpm_VoxelInfoLMR::Init().

6.27.3.6 double BCMFileIO::OctHeader::rgn[3]

領域サイズ

BCMFileCommon.h の 53 行で定義されています。

参照元 cpm_VoxelInfoLMR::Init(), と cpm_VoxelInfoLMR::LoadOctreeHeader().

6.27.3.7 unsigned int BCMFileIO::OctHeader::rootDims[3]

ルート分割数

BCMFileCommon.h の 54 行で定義されています。

参照元 cpm_VoxelInfoLMR::Init(), cpm_VoxelInfoLMR::LoadOctreeHeader(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

この構造体の説明は次のファイルから生成されました:

- BCMFileCommon.h

6.28 クラス Partition

1 次元ブロック領域分割用ユーティリティクラス.

`#include <Partition.h>`

Public メソッド

- Partition (int nProcs, int nItems)
- ~Partition ()
  デストラクタ.
- int getStart (int rank) const
- int getEnd (int rank) const
- int getNum (int rank) const
- int getRank (int i) const
- void print () const
  分割内容を出力.

Private 変数

- int nProcs
  プロセス数
- int nItems
  全要素数
- std::vector< int > end
  各プロセスの末尾要素番号を納めたリスト

6.28.1 説明

1 次元ブロック領域分割用ユーティリティクラス.

Partition.h の 27 行で定義されています。

6.28.2 コンストラクタとデストラクタ

6.28.2.1 Partition::Partition (int *nProcs*, int *nItems*) [inline]

コンストラクタ.

引数

in	*nProcs*	プロセス数
in	*nItems*	全要素数

Partition.h の 40 行で定義されています。

参照先 end, と nProcs.

6.28.2.2 Partition::~Partition ( ) [inline]

デストラクタ.

Partition.h の 56 行で定義されています。

6.28.3 関数

6.28.3.1 int Partition::getEnd (int *rank*) const [inline]

末尾要素番号の取得.

引数

in	*rank*	プロセス番号

戻り値

末尾要素番号+1

Partition.h の 74 行で定義されています。

参照先 end, と nProcs.

6.28.3.2 int Partition::getNum (int *rank*) const [inline]

担当要素数を取得.

引数

in	*rank*	プロセス番号

戻り値

担当要素数

Partition.h の 84 行で定義されています。

参照先 end, と nProcs.

6.28.3.3 int Partition::getRank (int *i*) const [inline]

担当プロセスを取得

引数

in	*i*	要素番号

戻り値

担当プロセス番号

覚え書き

範囲外の要素番号が指定された場合MPI::PROC_NULL を返す.

Partition.h の 97 行で定義されています。

参照先 end, と nItems.

参照元 BCMOctree::makeNeighborInfo().

6.28.3.4 int Partition::getStart (int *rank*) const [inline]

先頭要素番号の取得.

引数

in	*rank*	プロセス番号

戻り値

先頭要素番号

Partition.h の 63 行で定義されています。

参照先 end, と nProcs.

6.28.3.5 void Partition::print ( ) const [inline]

分割内容を出力.

Partition.h の 105 行で定義されています。

参照先 end, と nProcs.

6.28.4 変数

6.28.4.1 std::vector<int> Partition::end [private]

各プロセスの末尾要素番号を納めたリスト

Partition.h の 32 行で定義されています。

参照元 getEnd(), getNum(), getRank(), getStart(), Partition(), と print().

6.28.4.2 int Partition::nItems [private]

全要素数

Partition.h の 30 行で定義されています。

参照元 getRank().

6.28.4.3 int Partition::nProcs [private]

プロセス数

Partition.h の 29 行で定義されています。

参照元 getEnd(), getNum(), getStart(), Partition(), と print().

このクラスの説明は次のファイルから生成されました:

- Partition.h

6.29 クラス Pedigree

`#include <Pedigree.h>`

Public メソッド

- Pedigree (unsigned rootID=0)
  コンストラクタ *(**ルートノード**).*
- Pedigree (unsigned level, unsigned x, unsigned y, unsigned z, unsigned rootID=0)
- Pedigree (const Pedigree &parent, unsigned ijk)
- ~Pedigree ()
  デストラクタ.
- unsigned getLevel () const
- unsigned getX () const
- unsigned getY () const
- unsigned getZ () const
- unsigned getRootID () const
- unsigned getUpperBound () const
- unsigned getX (unsigned level) const
- unsigned getY (unsigned level) const
- unsigned getZ (unsigned level) const
- unsigned getChildId (unsigned level) const
- void serialize (void *buf) const
- void deserialize (const void *buf)

Static Public メソッド

- static size_t GetSerializeSize ()

Static Public 変数

- static const unsigned MaxLevel = 0xf
  最大レベル *(4* **ビット***)*
- static const unsigned MaxRootID = 0xfff
  最大ルート数 *(12* **ビット***)*
- static const unsigned MaxCoord = 0xffff
  最大座標値 *(16* **ビット***)*

Private メソッド

- void setPedigree (unsigned level, unsigned x, unsigned y, unsigned z, unsigned rootID)

Private 変数

- uint64_t p
  Pedigre **格納用内部** *64* **ビットデータ**

6.29.1 説明

Pedigree クラス.

```
*    64 ビットによる実装.
*    x:      [63:48] (16bit)
*    y:      [47:32] (16bit)
*    z:      [31:16] (16bit)
*    rootID: [15:4]  (12bit)
*    level:  [3:0]   ( 4bit)
*
```

Pedigree.h の 36 行で定義されています。

6.29.2 コンストラクタとデストラクタ

6.29.2.1 Pedigree::Pedigree (unsigned *rootID* = 0) [inline]

コンストラクタ (ルートノード).

Pedigree.h の 73 行で定義されています。

参照先 setPedigree().

6.29.2.2 Pedigree::Pedigree (unsigned *level*, unsigned *x*, unsigned *y*, unsigned *z*, unsigned *rootID* = 0) [inline]

コンストラクタ.

引数

in	*level*	ツリーレベル
in	*x*	x 位置
in	*y*	y 位置
in	*z*	z 位置
in	*rootID*	ルートID

Pedigree.h の 85 行で定義されています。

参照先 setPedigree().

6.29.2.3 Pedigree::Pedigree (const Pedigree & *parent*, unsigned *ijk*) [inline]

コンストラクタ (子ノード).

引数

in	*parent*	親ノードのPedigree
in	*ijk*	子ノード番号 (0〜7)

Pedigree.h の 100 行で定義されています。

参照先 getLevel(), getRootID(), getX(), getY(), getZ(), と setPedigree().

6.29.2.4 Pedigree::~Pedigree ( ) [inline]

デストラクタ.

Pedigree.h の 115 行で定義されています。

6.29.3 関数

6.29.3.1 void Pedigree::deserialize (const void * *buf*) [inline]

デシリアライズ.

引数

in	*buf*	シリアライズデータ

Pedigree.h の 211 行で定義されています。

参照先 p.

参照元 BCMOctree::buildTreeFromPedigreeList().

6.29.3.2 unsigned Pedigree::getChildId (unsigned *level*) const [inline]

指定されたレベルでの子ノード番号を取得.

引数

in	*level*	レベル

戻り値

子ノード番号 (0〜7)

Pedigree.h の 188 行で定義されています。

参照先 getX(), getY(), と getZ().

参照元 BCMOctree::buildTreeFromPedigreeList(), と BCMOctree::makeNeighborInfo().

6.29.3.3 unsigned Pedigree::getLevel ( ) const [inline]

ツリーレベルを取得.

戻り値

ツリーレベル

Pedigree.h の 121 行で定義されています。

参照先 p.

参照元 BCMOctree::buildTreeFromPedigreeList(), BCMOctree::findNeighborNode(), getUpperBound(), getX(), getY(), getZ(), operator<<(), と Pedigree().

6.29.3.4 unsigned Pedigree::getRootID ( ) const [inline]

ルートID を取得.

戻り値

ルートID

Pedigree.h の 145 行で定義されています。

参照先 p.

参照元 BCMOctree::buildTreeFromPedigreeList(), BCMOctree::checkOnOuterBoundary(), BCMOctree::findNeighborNode(), BCMOctree::getOrigin(), operator<<(), Pedigree(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.29.3.5 static size_t Pedigree::GetSerializeSize ( ) [inline], [static]

シリアライズに必要なバイト数を取得.

戻り値

バイト数

Pedigree.h の 197 行で定義されています。

参照元 BCMOctree::broadcast(), BCMOctree::buildTreeFromPedigreeList(), BCMOctree::packPedigrees(), と BCMOctree::ReceiveFromMaster().

6.29.3.6 unsigned Pedigree::getUpperBound ( ) const [inline]

そのレベルでの最大座標値を取得

戻り値

最大座標値 (= 2 のレベル値乗)

Pedigree.h の 151 行で定義されています。

参照先 getLevel().

参照元 BCMOctree::checkOnOuterBoundary(), BCMOctree::findNeighborNode(), BCMOctree::getOrigin(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.29.3.7 unsigned Pedigree::getX ( ) const [inline]

X 方向位置を取得.

戻り値

X 方向位置

Pedigree.h の 127 行で定義されています。

参照先 p.

参照元 BCMOctree::checkOnOuterBoundary(), BCMOctree::findNeighborNode(), getChildId(), BCMOctree::getOrigin(), operator<<(), Pedigree(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.29.3.8 unsigned Pedigree::getX (unsigned *level*) const [inline]

指定されたレベルのX 座標値を取得.

引数

in	*level*	レベル

戻り値

X 座標値 (0 or 1)

Pedigree.h の 158 行で定義されています。

参照先 getLevel(), MaxLevel, と p.

6.29.3.9 unsigned Pedigree::getY ( ) const [inline]

Y 方向位置を取得.

戻り値

Y 方向位置

Pedigree.h の 133 行で定義されています。

参照先 p.

参照元 BCMOctree::checkOnOuterBoundary(), BCMOctree::findNeighborNode(), getChildId(), BCMOctree::getOrigin(), operator<<(), Pedigree(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.29.3.10 unsigned Pedigree::getY (unsigned *level*) const [inline]

指定されたレベルのY 座標値を取得.

引数

in	*level*	レベル

戻り値

Y 座標値 (0 or 1)

Pedigree.h の 168 行で定義されています。

参照先 getLevel(), MaxLevel, と p.

6.29.3.11 unsigned Pedigree::getZ ( ) const [inline]

Z 方向位置を取得.

戻り値

Z 方向位置

Pedigree.h の 139 行で定義されています。

参照先 p.

参照元 BCMOctree::checkOnOuterBoundary(), BCMOctree::findNeighborNode(), getChildId(), BCMOctree::getOrigin(), operator<<(), Pedigree(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.29.3.12 unsigned Pedigree::getZ (unsigned *level*) const [inline]

指定されたレベルのZ 座標値を取得.

引数

in	*level*	レベル

戻り値

Z 座標値 (0 or 1)

Pedigree.h の 178 行で定義されています。

参照先 getLevel(), MaxLevel, と p.

6.29.3.13 void Pedigree::serialize (void * *buf*) const [inline]

シリアライズ.

引数

out	*buf*	シリアライズデータの出力先領域

Pedigree.h の 203 行で定義されています。

参照先 p.

6.29.3.14 void Pedigree::setPedigree (unsigned *level*, unsigned *x*, unsigned *y*, unsigned *z*, unsigned *rootID*) [inline], [private]

内部データにPedigree 値を設定.

引数

in	*level*	ツリーレベル
in	*x*	x 位置
in	*y*	y 位置
in	*z*	z 位置
in	*rootID*	ルートID

Pedigree.h の 56 行で定義されています。

参照先 MaxCoord, MaxLevel, MaxRootID, と p.

参照元 Pedigree().

6.29.4 変数

6.29.4.1 const unsigned Pedigree::MaxCoord = 0xffff [static]

最大座標値 (16 ビット)

Pedigree.h の 42 行で定義されています。

参照元 setPedigree().

6.29.4.2 const unsigned Pedigree::MaxLevel = 0xf [static]

最大レベル (4 ビット)

Pedigree.h の 40 行で定義されています。

参照元 getX(), getY(), getZ(), と setPedigree().

6.29.4.3 const unsigned Pedigree::MaxRootID = 0xfff [static]

最大ルート数 (12 ビット)

Pedigree.h の 41 行で定義されています。

参照元 setPedigree().

6.29.4.4 uint64_t Pedigree::p [private]

Pedigre 格納用内部 64 ビットデータ

Pedigree.h の 46 行で定義されています。

参照元 deserialize(), getLevel(), getRootID(), getX(), getY(), getZ(), serialize(), と setPedigree().

このクラスの説明は次のファイルから生成されました:

- Pedigree.h

6.30 クラス RootGrid

`#include <RootGrid.h>`

Public メソッド

- RootGrid (int nx, int ny, int nz)
- RootGrid (const Vec3i &n)
- ~RootGrid ()
  デストラクタ.
- int getSize () const
- int getSizeX () const
- int getSizeY () const
- int getSizeZ () const
- void setPeriodicX ()
  X **方向に周期境界条件を設定**.
- void setPeriodicY ()
  Y **方向に周期境界条件を設定**.
- void setPeriodicZ ()

Z 方向に周期境界条件を設定.
- void clearPeriodicX ()
  X 方向の周期境界条件を解除.
- void clearPeriodicY ()
  Y 方向の周期境界条件を解除.
- void clearPeriodicZ ()
  Z 方向の周期境界条件を解除.
- int rootID2indexX (int rootID) const
- int rootID2indexY (int rootID) const
- int rootID2indexZ (int rootID) const
- int index2rootID (int ix, int iy, int iz) const
- int getNeighborRoot (int rootID, Face face) const
- bool isOuterBoundary (int rootID, Face face) const
- void broadcast (MPI::Intracomm &comm=MPI::COMM_WORLD)

Static Public メソッド

- static RootGrid * ReceiveFromMaster (MPI::Intracomm &comm=MPI::COMM_WORLD)

Private 変数

- int nx
  X 方向ルート数
- int ny
  Y 方向ルート数
- int nz
  Z 方向ルート数
- bool periodicX
  X 方向周期境界条件フラグ
- bool periodicY
  Y 方向周期境界条件フラグ
- bool periodicZ
  Z 方向周期境界条件フラグ

6.30.1 説明

マルチルートOctree 用のルートブロック配置管理クラス.

覚え書き

位置 (i,j,k) のルートブロックのID は，i + nx*j + nx*ny*k

RootGrid.h の 30 行で定義されています。

6.30.2 コンストラクタとデストラクタ

6.30.2.1 RootGrid::RootGrid (int *nx*, int *ny*, int *nz*) [inline]

コンストラクタ.

引数

in	*nx*	X 方向ルート数
in	*ny*	Y 方向ルート数
in	*nz*	Z 方向ルート数

RootGrid.h の 48 行で定義されています。

6.30.2.2 RootGrid::RootGrid (const Vec3i & *n*) [inline]

コンストラクタ.

引数

in	*n*	ルート数ベクトル

RootGrid.h の 55 行で定義されています。

6.30.2.3 RootGrid::~RootGrid ( ) [inline]

デストラクタ.

RootGrid.h の 59 行で定義されています。

6.30.3 関数

6.30.3.1 void RootGrid::broadcast (MPI::Intracomm & *comm* = MPI::COMM_WORLD) [inline]

ルート配置情報を他プロセスにブロードキャスト.

引数

in	*comm*	MPI コミュニケータ

RootGrid.h の 228 行で定義されています。

参照元 BCMOctree::broadcast().

6.30.3.2 void RootGrid::clearPeriodicX ( ) [inline]

X 方向の周期境界条件を解除.

RootGrid.h の 95 行で定義されています。

参照元 cpm_VoxelInfoLMR::SetNeighborInfo().

6.30.3.3 void RootGrid::clearPeriodicY ( ) [inline]

Y 方向の周期境界条件を解除.

RootGrid.h の 98 行で定義されています。

参照元 cpm_VoxelInfoLMR::SetNeighborInfo().

6.30.3.4 void RootGrid::clearPeriodicZ ( ) [inline]

Z 方向の周期境界条件を解除.

RootGrid.h の 101 行で定義されています。

参照元 cpm_VoxelInfoLMR::SetNeighborInfo().

6.30.3.5 int RootGrid::getNeighborRoot (int *rootID,* Face *face*) const [inline]

隣接するルートのID を返す.

引数

in	*rootID*	ルートID
in	*face*	隣接面

戻り値

隣接するルートのルートID

覚え書き

隣接ルートが存在しない場合は-1 を返す.

RootGrid.h の 141 行で定義されています。

参照先 EX_FAILURE, Exit, X_M, X_P, Y_M, Y_P, Z_M, と Z_P.

参照元 BCMOctree::findNeighborNode().

6.30.3.6 int RootGrid::getSize ( ) const [inline]

ルート総数を取得.

戻り値

ルート総数

RootGrid.h の 65 行で定義されています。

参照元 BCMOctree::BCMOctree(), BCMOctree::broadcast(), BCMOctree::buildTreeFromPedigreeList(), と BCMOctree::∼BCMOctree().

6.30.3.7 int RootGrid::getSizeX ( ) const [inline]

X 方向ルート数を取得.

戻り値

X 方向ルート数

RootGrid.h の 71 行で定義されています。

6.30.3.8 int RootGrid::getSizeY ( ) const [inline]

Y 方向ルート数を取得.

戻り値

Y 方向ルート数

RootGrid.h の 77 行で定義されています。

6.30.3.9 int RootGrid::getSizeZ ( ) const [inline]

Z 方向ルート数を取得.

戻り値

Z 方向ルート数

RootGrid.h の 83 行で定義されています。

6.30.3.10 int RootGrid::index2rootID (int *ix,* int *iy,* int *iz*) const [inline]

位置インデクスをルートID に変換.

引数

in	*ix*	X 方向インデクス
in	*iy*	Y 方向インデクス
in	*iz*	Z 方向インデクス

戻り値

ルートID

RootGrid.h の 131 行で定義されています。

6.30.3.11 bool RootGrid::isOuterBoundary (int *rootID,* Face *face*) const [inline]

指定した面が外部境界かどうかチェック.

引数

in	*rootID*	ルートID
in	*face*	隣接面

戻り値

指定した面が外部境界なら true

RootGrid.h の 203 行で定義されています。

参照先 EX_FAILURE, Exit, X_M, X_P, Y_M, Y_P, Z_M, と Z_P.

参照元 BCMOctree::checkOnOuterBoundary().

6.30.3.12 static RootGrid* RootGrid::ReceiveFromMaster (MPI::Intracomm & *comm* = MPI::COMM_WORLD) [inline], [static]

ランク 0 からルート配置情報を受信.

引数

in	*comm*	MPI コミュニケータ

RootGrid.h の 244 行で定義されています。

参照先 setPeriodicX(), setPeriodicY(), と setPeriodicZ().

参照元 BCMOctree::ReceiveFromMaster().

6.30.3.13 int RootGrid::rootID2indexX (int *rootID*) const [inline]

X 方向インデクスを取得.

引数

in	*rootID*	ルートID

戻り値

X 方向インデクス

RootGrid.h の 108 行で定義されています。

参照元 BCMOctree::getOrigin(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.30.3.14 int RootGrid::rootID2indexY (int *rootID*) const [inline]

Y 方向インデクスを取得.

引数

in	*rootID*	ルートID

戻り値

Y 方向インデクス

RootGrid.h の 115 行で定義されています。

参照元 BCMOctree::getOrigin(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.30.3.15 int RootGrid::rootID2indexZ (int *rootID*) const [inline]

Z 方向インデクスを取得.

引数

in	*rootID*	ルートID

戻り値

Z 方向インデクス

RootGrid.h の 122 行で定義されています。

参照元 BCMOctree::getOrigin(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.30.3.16 void RootGrid::setPeriodicX ( ) [inline]

X 方向に周期境界条件を設定.

RootGrid.h の 86 行で定義されています。

参照元 ReceiveFromMaster(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.30.3.17 void RootGrid::setPeriodicY ( ) [inline]

Y 方向に周期境界条件を設定.

RootGrid.h の 89 行で定義されています。

参照元 ReceiveFromMaster(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.30.3.18 void RootGrid::setPeriodicZ ( ) [inline]

Z 方向に周期境界条件を設定.

RootGrid.h の 92 行で定義されています。

参照元 ReceiveFromMaster(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.30.4 変数

6.30.4.1 int RootGrid::nx [private]

X 方向ルート数

RootGrid.h の 32 行で定義されています。

6.30.4.2 int RootGrid::ny [private]

Y 方向ルート数

RootGrid.h の 33 行で定義されています。

6.30.4.3 int RootGrid::nz [private]

Z 方向ルート数

RootGrid.h の 34 行で定義されています。

6.30.4.4 bool RootGrid::periodicX [private]

X 方向周期境界条件フラグ

RootGrid.h の 36 行で定義されています。

6.30.4.5 bool RootGrid::periodicY [private]

Y 方向周期境界条件フラグ

RootGrid.h の 37 行で定義されています。

6.30.4.6 bool RootGrid::periodicZ [private]

Z 方向周期境界条件フラグ

RootGrid.h の 38 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:

- RootGrid.h

6.31 構造体 S_BNDCOMM_BUFFER

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

Public メソッド

- S_BNDCOMM_BUFFER ()
- ~S_BNDCOMM_BUFFER ()
- size_t CalcBufferSize ()

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

6.31.1 説明

袖通信バッファ情報

cpm_ParaManager.h の 27 行で定義されています。

6.31.2 コンストラクタとデストラクタ

6.31.2.1 S_BNDCOMM_BUFFER::S_BNDCOMM_BUFFER ( ) [inline]

cpm_ParaManager.h の 38 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.

6.31.2.2 S_BNDCOMM_BUFFER::~S_BNDCOMM_BUFFER ( ) [inline]

cpm_ParaManager.h の 50 行で定義されています。

参照先 m_bufX, m_bufY, と m_bufZ.

6.31.3 関数

6.31.3.1 size_t S_BNDCOMM_BUFFER::CalcBufferSize ( ) [inline]

バッファサイズの計算

戻り値

バッファサイズ [Byte]

cpm_ParaManager.h の 63 行で定義されています。

参照先 m_nwX, m_nwY, m_nwZ, と REAL_BUF_TYPE.

参照元 cpm_ParaManager::GetBndCommBufferSize().

6.31.4 変数

6.31.4.1 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufX[4]

バッファ

cpm_ParaManager.h の 34 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().

6.31.4.2 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufY[4]

バッファ

cpm_ParaManager.h の 35 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().

6.31.4.3 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufZ[4]

バッファ

cpm_ParaManager.h の 36 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().

6.31.4.4 size_t S_BNDCOMM_BUFFER::m_maxN

最大成分数

cpm_ParaManager.h の 30 行で定義されています。

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

6.31.4.5 size_t S_BNDCOMM_BUFFER::m_maxVC

最大袖数

cpm_ParaManager.h の 29 行で定義されています。

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

6.31.4.6 size_t S_BNDCOMM_BUFFER::m_nwX

バッファサイズ

cpm_ParaManager.h の 31 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(), CalcBufferSize(), cpm_ParaManager::PeriodicCommS4D(), cpm_ParaManager::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManager::SetBndCommBuffer(), cpm_ParaManager::wait_BndCommS4D(), と cpm_ParaManager::wait_BndCommS4DEx().

6.31.4.7 size_t S_BNDCOMM_BUFFER::m_nwY

バッファサイズ

cpm_ParaManager.h の 32 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(), CalcBufferSize(), cpm_ParaManager::PeriodicCommS4D(), cpm_ParaManager::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManager::SetBndCommBuffer(), cpm_ParaManager::wait_BndCommS4D(), と cpm_ParaManager::wait_BndCommS4DEx().

6.31.4.8 size_t S_BNDCOMM_BUFFER::m_nwZ

バッファサイズ

cpm_ParaManager.h の 33 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(), CalcBufferSize(), 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

6.32 構造体 S_OCT_DOMAIN_INFO

`#include <cpm_TextParserDomainLMR.h>`

Public メソッド

- S_OCT_DOMAIN_INFO ()
- void print ()

Public 変数

- double origin [3]

原点座標

- double region [3]

  領域幅

- std::string octFile

  oct **ファイル名**

- int size [3]

  1 **リーフの格子数**

- std::string unitLength

  長さ単位文字列

6.32.1 説明

領域情報ファイル構造体

cpm_TextParserDomainLMR.h の 25 行で定義されています。

6.32.2 コンストラクタとデストラクタ

6.32.2.1 S_OCT_DOMAIN_INFO::S_OCT_DOMAIN_INFO ( ) [inline]

コンストラクタ

cpm_TextParserDomainLMR.h の 34 行で定義されています。

参照先 octFile, origin, region, size, と unitLength.

6.32.3 関数

6.32.3.1 void S_OCT_DOMAIN_INFO::print ( ) [inline]

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

参照先 octFile, origin, region, size, と unitLength.

参照元 cpm_VoxelInfoLMR::Init().

6.32.4 変数

6.32.4.1 std::string S_OCT_DOMAIN_INFO::octFile

oct ファイル名

cpm_TextParserDomainLMR.h の 29 行で定義されています。

参照元 cpm_VoxelInfoLMR::GetNumLeaf(), cpm_VoxelInfoLMR::Init(), print(), cpm_TextParserDomainLMR::ReadBCMTree(), と S_OCT_DOMAIN_INFO().

6.32.4.2 double S_OCT_DOMAIN_INFO::origin[3]

原点座標

cpm_TextParserDomainLMR.h の 27 行で定義されています。

参照元 print(), cpm_TextParserDomainLMR::ReadDomain(), S_OCT_DOMAIN_INFO(), と cpm_VoxelInfoLMR::SetGlobaliDomainInfo().

6.32.4.3 double S_OCT_DOMAIN_INFO::region[3]

領域幅

cpm_TextParserDomainLMR.h の 28 行で定義されています。

参照元 print(), cpm_TextParserDomainLMR::ReadDomain(), S_OCT_DOMAIN_INFO(), と cpm_VoxelInfoLMR::SetGlobaliDomainInfo().

6.32.4.4 int S_OCT_DOMAIN_INFO::size[3]

1 リーフの格子数

cpm_TextParserDomainLMR.h の 30 行で定義されています。

参照元 print(), cpm_TextParserDomainLMR::ReadLeafBlock(), S_OCT_DOMAIN_INFO(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), と cpm_VoxelInfoLMR::SetLocalDomainInfo().

6.32.4.5 std::string S_OCT_DOMAIN_INFO::unitLength

長さ単位文字列

cpm_TextParserDomainLMR.h の 31 行で定義されています。

参照元 print(), cpm_TextParserDomainLMR::ReadLeafBlock(), と S_OCT_DOMAIN_INFO().

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_TextParserDomainLMR.h

6.33 構造体 cpm_LeafCommInfo::stCommInfo

`#include <cpm_LeafCommInfo.h>`

Public メソッド

- stCommInfo ()
  コンストラクタ
- int GetLeafID (int type)
- size_t CalcSendBufferSize (size_t sz_face[2], size_t vc_comm, size_t nmax)
- size_t CalcRecvBufferSize (size_t sz_face[2], size_t vc_comm, size_t nmax)

Public 変数

- int iOwnLeafID
  自身のリーフ *ID*
- int iDistLeafID
  通信相手のリーフ *ID*
- int iLevelDiff
  通信相手とのレベル差
- int iFaceIdx
  自身の *faceIdx*
- bool bPeriodic
  周期境界フラグ

6.33.1 説明

１通信経路情報構造体

cpm_LeafCommInfo.h の 33 行で定義されています。

6.33.2 コンストラクタとデストラクタ

6.33.2.1 cpm_LeafCommInfo::stCommInfo::stCommInfo ( ) [inline]

コンストラクタ

cpm_LeafCommInfo.h の 51 行で定義されています。

参照先 bPeriodic, iDistLeafID, iFaceIdx, iLevelDiff, と iOwnLeafID.

6.33.3 関数

6.33.3.1 size_t cpm_LeafCommInfo::stCommInfo::CalcRecvBufferSize (size_t *sz_face[2],* size_t *vc_comm,* size_t *nmax*) [inline]

必要な受信バッファサイズを計算

引数

in	*sz_face*	1 リーフの格子数 (平面内２軸)
in	*vc_comm*	通信する仮想セル数
in	*nmax*	受信バッファの最大成分数

cpm_LeafCommInfo.h の 108 行で定義されています。

参照先 iLevelDiff.

参照元 cpm_ParaManagerLMR::recv_LMR(), cpm_ParaManagerLMR::recv_LMR_Ex_wait(), と cpm_ParaManagerLMR::recv_LMR_wait().

6.33.3.2 size_t cpm_LeafCommInfo::stCommInfo::CalcSendBufferSize (size_t *sz_face[2],* size_t *vc_comm,* size_t *nmax*) [inline]

必要な送信バッファサイズを計算

引数

in	*sz_face*	1 リーフの格子数 (平面内２軸)
in	*vc_comm*	通信する仮想セル数
in	*nmax*	送信バッファの最大成分数

cpm_LeafCommInfo.h の 81 行で定義されています。

参照先 iLevelDiff.

参照元 cpm_ParaManagerLMR::copy_LMR(), cpm_ParaManagerLMR::copy_LMR_Ex(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::send_LMR(), と cpm_ParaManagerLMR::send_LMR_Ex().

6.33.3.3 int cpm_LeafCommInfo::stCommInfo::GetLeafID (int *type*) [inline]

LeafID を取得

引数

in	*type*	ソートタイプ (0:iOwnLeafID, 1:iDistLeafID)

cpm_LeafCommInfo.h の 63 行で定義されています。

参照先 iDistLeafID, と iOwnLeafID.

参照元 cpm_LeafCommInfo::Qsort().

6.33.4 変数

6.33.4.1 bool cpm_LeafCommInfo::stCommInfo::bPeriodic

周期境界フラグ

cpm_LeafCommInfo.h の 48 行で定義されています。

参照元 cpm_ParaManagerLMR::copy_LMR(), cpm_ParaManagerLMR::copy_LMR_Ex(), cpm_ParaManagerLMR::recv_LMR(), cpm_ParaManagerLMR::recv_LMR_Ex_wait(), cpm_ParaManagerLMR::recv_LMR_wait(), cpm_LeafCommInfo::SearchDistCommInfo(), cpm_ParaManagerLMR::send_LMR(), cpm_ParaManagerLMR::send_LMR_Ex(), cpm_ParaManagerLMR::SetBndCommBuffer(), と stCommInfo().

6.33.4.2 int cpm_LeafCommInfo::stCommInfo::iDistLeafID

通信相手のリーフID

cpm_LeafCommInfo.h の 39 行で定義されています。

参照元 GetLeafID(), cpm_LeafCommInfo::SearchDistCommInfo(), cpm_ParaManagerLMR::SetBndCommBuffer(), と stCommInfo().

6.33.4.3 int cpm_LeafCommInfo::stCommInfo::iFaceIdx

自身の faceIdx

cpm_LeafCommInfo.h の 45 行で定義されています。

参照元 cpm_ParaManagerLMR::packMX(), cpm_ParaManagerLMR::packMXEx(), cpm_ParaManagerLMR::packMY(), cpm_ParaManagerLMR::packMYEx(), cpm_ParaManagerLMR::packMZ(), cpm_ParaManagerLMR::packMZEx(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::packPXEx(), cpm_ParaManagerLMR::packPY(), cpm_ParaManagerLMR::packPYEx(), cpm_ParaManagerLMR::packPZ(), cpm_ParaManagerLMR::packPZEx(), cpm_ParaManagerLMR::SetBndCommBuffer(), stCommInfo(), cpm_ParaManagerLMR::unpackMX(), cpm_ParaManagerLMR::unpackMXEx(), cpm_ParaManagerLMR::unpackMY(), cpm_ParaManagerLMR::unpackMYEx(), cpm_ParaManagerLMR::unpackMZ(), cpm_ParaManagerLMR::unpackMZEx(), cpm_ParaManagerLMR::unpackPX(), cpm_ParaManagerLMR::unpackPXEx(), cpm_ParaManagerLMR::unpackPY(), cpm_ParaManagerLMR::unpackPYEx(), cpm_ParaManagerLMR::unpackPZ(), と cpm_ParaManagerLMR::unpackPZEx().

6.33.4.4 int cpm_LeafCommInfo::stCommInfo::iLevelDiff

通信相手とのレベル差

cpm_LeafCommInfo.h の 42 行で定義されています。

参照元 CalcRecvBufferSize(), CalcSendBufferSize(), cpm_ParaManagerLMR::packMX(), cpm_ParaManagerLMR::packMXEx(), cpm_ParaManagerLMR::packMY(), cpm_ParaManagerLMR::packMYEx(), cpm_ParaManagerLMR::packMZ(), cpm_ParaManagerLMR::packMZEx(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::packPXEx(), cpm_ParaManagerLMR::packPY(), cpm_ParaManagerLMR::packPYEx(), cpm_ParaManagerLMR::packPZ(), cpm_ParaManagerLMR::packPZEx(), cpm_ParaManagerLMR::SetBndCommBuffer(), stCommInfo(), cpm_ParaManagerLMR::unpackMX(), cpm_ParaManagerLMR::unpackMXEx(), cpm_ParaManagerLMR::unpackMY(), cpm_ParaManagerLMR::unpackMYEx(), cpm_ParaManagerLMR::unpackMZ(), cpm_ParaManager-

LMR::unpackMZEx(), cpm_ParaManagerLMR::unpackPX(), cpm_ParaManagerLMR::unpackPXEx(), cpm_ParaManagerLMR::unpackPY(), cpm_ParaManagerLMR::unpackPYEx(), cpm_ParaManagerLMR::unpackPZ(), と cpm_ParaManagerLMR::unpackPZEx().

6.33.4.5 int cpm_LeafCommInfo::stCommInfo::iOwnLeafID

自身のリーフID

cpm_LeafCommInfo.h の 36 行で定義されています。

参照元 GetLeafID(), cpm_ParaManagerLMR::packMX(), cpm_ParaManagerLMR::packMXEx(), cpm_ParaManagerLMR::packMY(), cpm_ParaManagerLMR::packMYEx(), cpm_ParaManagerLMR::packMZ(), cpm_ParaManagerLMR::packMZEx(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::packPXEx(), cpm_ParaManagerLMR::packPY(), cpm_ParaManagerLMR::packPYEx(), cpm_ParaManagerLMR::packPZ(), cpm_ParaManagerLMR::packPZEx(), cpm_LeafCommInfo::SearchDistCommInfo(), cpm_ParaManagerLMR::SetBndCommBuffer(), stCommInfo(), cpm_ParaManagerLMR::unpackMX(), cpm_ParaManagerLMR::unpackMXEx(), cpm_ParaManagerLMR::unpackMY(), cpm_ParaManagerLMR::unpackMYEx(), cpm_ParaManagerLMR::unpackMZ(), cpm_ParaManagerLMR::unpackMZEx(), cpm_ParaManagerLMR::unpackPX(), cpm_ParaManagerLMR::unpackPXEx(), cpm_ParaManagerLMR::unpackPY(), cpm_ParaManagerLMR::unpackPYEx(), cpm_ParaManagerLMR::unpackPZ(), と cpm_ParaManagerLMR::unpackPZEx().

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_LeafCommInfo.h

6.34 クラス テンプレート Vec3class::Vec3< T >

`#include <Vec3.h>`

Public メソッド

- Vec3 (T v=0)
- Vec3 (T _x, T _y, T _z)
- Vec3 (const T v[3])
- Vec3 (const Vec3 &v)
- Vec3< T > & assign (T _x, T _y, T _z)
- operator T * ()
- operator const T * () const
- T * ptr ()
- const T * ptr () const
- T & operator[] (const AxisEnum &axis)
- const T & operator[] (const AxisEnum &axis) const
- Vec3< T > & operator+= (const Vec3< T > &v)
- Vec3< T > & operator-= (const Vec3< T > &v)
- Vec3< T > & operator*= (const Vec3< T > &v)
- Vec3< T > & operator/= (const Vec3< T > &v)
- Vec3< T > & operator*= (T s)
- Vec3< T > & operator/= (T s)
- Vec3< T > operator+ (const Vec3< T > &v) const
- Vec3< T > operator- (const Vec3< T > &v) const
- Vec3< T > operator* (const Vec3< T > &v) const
- Vec3< T > operator/ (const Vec3< T > &v) const
- Vec3< T > operator* (T s) const
- Vec3< T > operator/ (T s) const
- Vec3< T > operator- () const

- bool operator== (const Vec3< T > &v) const
- bool operator!= (const Vec3< T > &v) const
- T lengthSquared () const
- T length () const
- Vec3< T > & normalize ()
- Vec3< T > & normalize (T *len)
- T average () const

Static Public メソッド

- static Vec3< T > xaxis ()
- static Vec3< T > yaxis ()
- static Vec3< T > zaxis ()

Public 変数

- T x
- T y
- T z

6.34.1 説明

template<typename T>class Vec3class::Vec3< T >

Vec3.h の 62 行で定義されています。

6.34.2 コンストラクタとデストラクタ

6.34.2.1 template<typename T> Vec3class::Vec3< T >::Vec3 (T *v* = 0) [inline]

Vec3.h の 67 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.2.2 template<typename T> Vec3class::Vec3< T >::Vec3 (T *_x*, T *_y*, T *_z*) [inline]

Vec3.h の 68 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.2.3 template<typename T> Vec3class::Vec3< T >::Vec3 (const T *v[3]*) [inline]

Vec3.h の 69 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.2.4 template<typename T> Vec3class::Vec3< T >::Vec3 (const Vec3< T > & *v*) [inline]

Vec3.h の 70 行で定義されています。

6.34.3 関数

6.34.3.1 template<typename T> Vec3<T>& Vec3class::Vec3< T >::assign (T _x, T _y, T _z) [inline]

Vec3.h の 72 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.2 template<typename T> T Vec3class::Vec3< T >::average ( ) const [inline]

Vec3.h の 200 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.3 template<typename T> T Vec3class::Vec3< T >::length ( ) const [inline]

Vec3.h の 182 行で定義されています。

参照先 Vec3class::Vec3< T >::lengthSquared().

参照元 Vec3class::Vec3< T >::normalize().

6.34.3.4 template<typename T> T Vec3class::Vec3< T >::lengthSquared ( ) const [inline]

Vec3.h の 178 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

参照元 Vec3class::Vec3< T >::length(), と Vec3class::lessVec3f().

6.34.3.5 template<typename T> Vec3<T>& Vec3class::Vec3< T >::normalize ( ) [inline]

Vec3.h の 184 行で定義されています。

参照先 Vec3class::Vec3< T >::length().

6.34.3.6 template<typename T> Vec3<T>& Vec3class::Vec3< T >::normalize (T * len) [inline]

Vec3.h の 192 行で定義されています。

参照先 Vec3class::Vec3< T >::length().

6.34.3.7 template<typename T> Vec3class::Vec3< T >::operator const T * ( ) const [inline]

Vec3.h の 78 行で定義されています。

参照先 Vec3class::Vec3< T >::x.

6.34.3.8 template<typename T> Vec3class::Vec3< T >::operator T * ( ) [inline]

Vec3.h の 77 行で定義されています。

参照先 Vec3class::Vec3< T >::x.

6.34.3.9 template<typename T> bool Vec3class::Vec3< T >::operator!= (const Vec3< T > & v) const [inline]

Vec3.h の 170 行で定義されています。

6.34.3.10 template<typename T> Vec3<T> Vec3class::Vec3< T >::operator∗ (const Vec3< T > & *v*) const [inline]

Vec3.h の 145 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.11 template<typename T> Vec3<T> Vec3class::Vec3< T >::operator∗ (T *s*) const [inline]

Vec3.h の 153 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.12 template<typename T> Vec3<T>& Vec3class::Vec3< T >::operator∗= (const Vec3< T > & *v*) [inline]

Vec3.h の 116 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.13 template<typename T> Vec3<T>& Vec3class::Vec3< T >::operator∗= (T *s*) [inline]

Vec3.h の 126 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.14 template<typename T> Vec3<T> Vec3class::Vec3< T >::operator+ (const Vec3< T > & *v*) const [inline]

Vec3.h の 137 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.15 template<typename T> Vec3<T>& Vec3class::Vec3< T >::operator+= (const Vec3< T > & *v*) [inline]

Vec3.h の 106 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.16 template<typename T> Vec3<T> Vec3class::Vec3< T >::operator- (const Vec3< T > & *v*) const [inline]

Vec3.h の 141 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.17 template<typename T> Vec3<T> Vec3class::Vec3< T >::operator- ( ) const [inline]

Vec3.h の 162 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.18 template<typename T> Vec3<T>& Vec3class::Vec3< T >::operator-= (const Vec3< T > & *v*) [inline]

Vec3.h の 111 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.19 template<typename T> Vec3<T> Vec3class::Vec3< T >::operator/ (const Vec3< T > & *v*) const [inline]

Vec3.h の 149 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.20 template<typename T> Vec3<T> Vec3class::Vec3< T >::operator/ (T *s*) const [inline]

Vec3.h の 157 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.21 template<typename T> Vec3<T>& Vec3class::Vec3< T >::operator/= (const Vec3< T > & *v*) [inline]

Vec3.h の 121 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.22 template<typename T> Vec3<T>& Vec3class::Vec3< T >::operator/= (T *s*) [inline]

Vec3.h の 131 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.23 template<typename T> bool Vec3class::Vec3< T >::operator== (const Vec3< T > & *v*) const [inline]

Vec3.h の 166 行で定義されています。

参照先 Vec3class::Vec3< T >::x, Vec3class::Vec3< T >::y, と Vec3class::Vec3< T >::z.

6.34.3.24 template<typename T> T& Vec3class::Vec3< T >::operator[] (const AxisEnum & *axis*) [inline]

Vec3.h の 84 行で定義されています。

参照先 Vec3class::Vec3< T >::x.

6.34.3.25 template<typename T> const T& Vec3class::Vec3< T >::operator[] (const AxisEnum & *axis*) const [inline]

Vec3.h の 94 行で定義されています。

参照先 Vec3class::Vec3< T >::x.

6.34.3.26 template<typename T> T* Vec3class::Vec3< T >::ptr ( ) [inline]

Vec3.h の 79 行で定義されています。

参照先 Vec3class::Vec3< T >::x.

6.34.3.27 template<typename T> const T* Vec3class::Vec3< T >::ptr ( ) const [inline]

Vec3.h の 80 行で定義されています。

参照先 Vec3class::Vec3< T >::x.

6.34.3.28 template<typename T> static Vec3<T> Vec3class::Vec3< T >::xaxis ( ) [inline], [static]

Vec3.h の 174 行で定義されています。

6.34.3.29 template<typename T> static Vec3<T> Vec3class::Vec3< T >::yaxis ( ) [inline], [static]

Vec3.h の 175 行で定義されています。

6.34.3.30 template<typename T> static Vec3<T> Vec3class::Vec3< T >::zaxis ( ) [inline], [static]

Vec3.h の 176 行で定義されています。

6.34.4 変数

6.34.4.1 template<typename T> T Vec3class::Vec3< T >::x

Vec3.h の 65 行で定義されています。

参照元 Vec3class::Vec3< T >::assign(), Vec3class::Vec3< T >::average(), Vec3class::cross(), Vec3class::dot(), Vec3class::Vec3< T >::lengthSquared(), Vec3class::Vec3< T >::operator const T *(), Vec3class::Vec3< T >::operator T *(), Vec3class::Vec3< T >::operator*(), Vec3class::operator*(), Vec3class::Vec3< T >::operator*=(), Vec3class::Vec3< T >::operator+(), Vec3class::Vec3< T >::operator+=(), Vec3class::Vec3< T >::operator-(), Vec3class::Vec3< T >::operator-=(), Vec3class::Vec3< T >::operator/(), Vec3class::Vec3< T >::operator/=(), Vec3class::Vec3< T >::operator==(), Vec3class::Vec3< T >::operator[](), Vec3class::Vec3< T >::ptr(), と Vec3class::Vec3< T >::Vec3().

6.34.4.2 template<typename T> T Vec3class::Vec3< T >::y

Vec3.h の 65 行で定義されています。

参照元 Vec3class::Vec3< T >::assign(), Vec3class::Vec3< T >::average(), Vec3class::cross(), Vec3class::dot(), Vec3class::Vec3< T >::lengthSquared(), Vec3class::Vec3< T >::operator*(), Vec3class::operator*(), Vec3class::Vec3< T >::operator*=(), Vec3class::Vec3< T >::operator+(), Vec3class::Vec3< T >::operator+=(), Vec3class::Vec3< T >::operator-(), Vec3class::Vec3< T >::operator-=(), Vec3class::Vec3< T >::operator/(), Vec3class::Vec3< T >::operator/=(), Vec3class::Vec3< T >::operator==(), と Vec3class::Vec3< T >::Vec3().

6.34.4.3 template<typename T> T Vec3class::Vec3< T >::z

Vec3.h の 65 行で定義されています。

参照元 Vec3class::Vec3< T >::assign(), Vec3class::Vec3< T >::average(), Vec3class::cross(), Vec3class::dot(), Vec3class::Vec3< T >::lengthSquared(), Vec3class::Vec3< T >::operator*(), Vec3class::operator*(), Vec3class::Vec3< T >::operator*=(), Vec3class::Vec3< T >::operator+(), Vec3class::Vec3< T >::operator+=(), Vec3class::Vec3< T >::operator-(), Vec3class::Vec3< T >::operator-=(), Vec3class::Vec3< T >::operator/(), Vec3class::Vec3< T >::operator/=(), Vec3class::Vec3< T >::operator==(), と Vec3class::Vec3< T >::Vec3().

このクラスの説明は次のファイルから生成されました:

- Vec3.h

Chapter 7

ファイル

7.1 BCMFileCommon.h

BCM ファイルIO 用共通クラス群

```
#include <limits.h>
#include <vector>
#include <string>
#include <list>
#include "BitVoxel.h"
#include <stdint.h>
```

BCMFileCommon.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 BCMFileIO::OctHeader
  Octree **ファイルヘッダ構造体**
- struct BCMFileIO::LBHeader
  LeafBlock **ファイルヘッダ構造体**
- struct BCMFileIO::LBCellIDHeader
  LeafBlock **の***CellID* **ヘッダ構造体**
- struct BCMFileIO::GridRleCode
  RLE **圧縮符号の走査用構造体**
- struct BCMFileIO::IdxUnit
  インデックスファイル用単位系情報
- struct BCMFileIO::IdxProc
  インデックスファイル用プロセス情報

ネームスペース

- BCMFileIO

マクロ定義

- #define OCTREE_FILE_IDENTIFIER (('O' | ('C' << 8) | ('0' << 16) | ('1' << 24)))
  Octree **ファイルのエンディアン識別子** *(OC01)*
- #define LEAFBLOCK_FILE_IDENTIFIER (('L' | ('B' << 8) | ('0' << 16) | ('1' << 24)))

LeafBlock **ファイルのエンディアン識別子** *(LB01)*
- #define ALIGNMENT

型定義

- typedef BitVoxel::bitVoxelCell BCMFileIO::bitVoxelCell

列挙型

- enum BCMFileIO::LB_KIND {
  BCMFileIO::LB_CELLID = 0, BCMFileIO::LB_SCALAR = 1, BCMFileIO::LB_VECTOR3 = 3, BCMFileIO::LB_VECTOR4 = 4,
  BCMFileIO::LB_VECTOR6 = 6, BCMFileIO::LB_TENSOR = 9 }
  リーフブロックデータタイプ
- enum BCMFileIO::LB_DATA_TYPE {
  BCMFileIO::LB_INT8 = 0, BCMFileIO::LB_UINT8 = 1, BCMFileIO::LB_INT16 = 2, BCMFileIO::LB_UINT16 = 3,
  BCMFileIO::LB_INT32 = 4, BCMFileIO::LB_UINT32 = 5, BCMFileIO::LB_INT64 = 6, BCMFileIO::LB_UINT64 = 7,
  BCMFileIO::LB_FLOAT32 = 8, BCMFileIO::LB_FLOAT64 = 9 }
  リーフセルのデータ識別子

関数

- static void BCMFileIO::BSwap16 (void *a)
  2byte **用エンディアンスワップ**
- static void BCMFileIO::BSwap32 (void *a)
  4byte **用エンディアンスワップ**
- static void BCMFileIO::BSwap64 (void *a)
  8byte **用エンディアンスワップ**

変数

- struct BCMFileIO::OctHeader BCMFileIO::ALIGNMENT

7.1.1 説明

BCM ファイルIO 用共通クラス群

BCMFileCommon.h で定義されています。

7.1.2 マクロ定義

7.1.2.1 #define ALIGNMENT

BCMFileCommon.h の 45 行で定義されています。

7.1.2.2 #define LEAFBLOCK_FILE_IDENTIFIER (('L' | ('B' << 8) | ('0' << 16) | ('1' << 24)))

LeafBlock ファイルのエンディアン識別子 (LB01)

BCMFileCommon.h の 36 行で定義されています。

7.1.2.3 #define OCTREE_FILE_IDENTIFIER (('O' | ('C' << 8) | ('0' << 16) | ('1' << 24)))

Octree ファイルのエンディアン識別子 (OC01)

BCMFileCommon.h の 33 行で定義されています。

参照元 cpm_VoxelInfoLMR::LoadOctreeHeader().

7.2 BCMOctree.cpp

```
#include <algorithm>
#include "BCMOctree.h"
```

BCMOctree.cpp のインクルード依存関係図

7.3 BCMOctree.h

BCM 用マルチルートOCtree クラス

```
#include <vector>
#include "BCMTools.h"
#include "Vec3.h"
#include "RootGrid.h"
#include "Divider.h"
#include "Pedigree.h"
#include "Node.h"
#include "NeighborInfo.h"
#include "Partition.h"
```

BCMOctree.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 BCMOctree

7.3.1 説明

BCM 用マルチルートOCtree クラス

BCMOctree.h で定義されています。

7.4 BCMTools.h

BCM Tools 共通ヘッダ

```
#include "mpi.h"
#include <cstdio>
#include <cstdlib>
#include <cassert>
```

BCMTools.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 Exit(x) ((void)printf("exit at %s:%u\n", __FILE__, __LINE__), exit((x)))

呼び出し箇所が分かる *exit* **関数マクロ** *(assert* **の代わりに使用***).*

- #define NDEBUG

  DEBUG **マクロの定義時のみ** *assert* **マクロを有効に**.

列挙型

- enum Face {
  X_M, X_P, Y_M, Y_P,
  Z_M, Z_P, NUM_FACE }

  フェイス番号.
- enum Subface {
  SF_00, SF_01, SF_10, SF_11,
  NUM_SUBFACE }

  サブフェイス番号.
- enum ExitStatus {
  EX_SUCCESS = 0, EX_USAGE = 16, EX_MEMORY = 17, EX_OPEN_FILE = 18,
  EX_READ_CONFIG = 19, EX_READ_DATA = 20, EX_WRITE_DATA = 21, EX_FAILURE = 1 }

  リターンコード.

7.4.1 説明

BCM Tools 共通ヘッダ

BCMTools.h で定義されています。

7.4.2 マクロ定義

7.4.2.1 #define Exit(*x*) ((void)printf("exit at %s:%u\n", __FILE__, __LINE__), exit((x)))

呼び出し箇所が分かる exit 関数マクロ (assert の代わりに使用).

BCMTools.h の 30 行で定義されています。

参照元 NeighborInfo::childIdToSubface(), BCMOctree::findNeighborNode(), NeighborInfo::getNeighborChildId(), RootGrid::getNeighborRoot(), RootGrid::isOuterBoundary(), BCMOctree::makeNeighborInfo(), と NeighborInfo::reverseFace().

7.4.2.2 #define NDEBUG

DEBUG マクロの定義時のみ assert マクロを有効に.

BCMTools.h の 36 行で定義されています。

7.4.3 列挙型

7.4.3.1 enum ExitStatus

リターンコード.

列挙型の値

EX_SUCCESS successful termination

EX_USAGE incorrect command arguments

EX_MEMORY memory allocation failure

EX_OPEN_FILE open file error
EX_READ_CONFIG read configuration file error
EX_READ_DATA read data error
EX_WRITE_DATA write data error
EX_FAILURE other failure

BCMTools.h の 48 行で定義されています。

7.4.3.2 enum Face

フェイス番号.

列挙型の値

X_M
X_P
Y_M
Y_P
Z_M
Z_P
NUM_FACE

BCMTools.h の 42 行で定義されています。

7.4.3.3 enum Subface

サブフェイス番号.

列挙型の値

SF_00
SF_01
SF_10
SF_11
NUM_SUBFACE

BCMTools.h の 45 行で定義されています。

7.5 BitVoxel.h

ビットボクセル圧縮/展開ライブラリ

```
#include <cstdlib>
```

BitVoxel.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 BCMFileIO::BitVoxel
  ビットボクセル圧縮/展開ライブラリ

ネームスペース

- BCMFileIO

7.5.1 説明

ビットボクセル圧縮/展開ライブラリ

BitVoxel.h で定義されています。

7.6 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

7.6.1 説明

CPM のベースクラスのヘッダーファイル

日付

2012/05/31

cpm_Base.h で定義されています。

7.6.2 マクロ定義

7.6.2.1 #define CPM_INLINE inline

cpm_Base.h の 41 行で定義されています。

7.7 cpm_BaseParaManager.cpp

```
#include "cpm_BaseParaManager.h"
```

cpm_BaseParaManager.cpp のインクルード依存関係図

マクロ定義

- #define _ALL_DIM_PAD_

7.7.1 説明

パラレルマネージャ基底クラスのソースファイル

日付

2012/05/31

cpm_BaseParaManager.cpp で定義されています。

7.7.2 マクロ定義

7.7.2.1 #define _ALL_DIM_PAD_

cpm_BaseParaManager.cpp の 354 行で定義されています。

7.8 cpm_BaseParaManager.h

```
#include "cpm_Base.h"
#include <map>
#include <vector>
#include <typeinfo>
#include "cpm_DomainInfo.h"
#include "cpm_VoxelInfo.h"
#include "cpm_ObjList.h"
#include <string.h>
#include "inline/cpm_BaseParaManager_inline.h"
```

cpm_BaseParaManager.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_BaseParaManager

型定義

- typedef std::map< int, cpm_DefPointType > DefPointMap

7.8.1 説明

パラレルマネージャ基底クラスのヘッダーファイル

日付

2012/05/31

cpm_BaseParaManager.h で定義されています。

7.8.2 型定義

7.8.2.1 typedef std::map<int, cpm_DefPointType> DefPointMap

プロセスグループ毎の定義点タイプ管理マップ

cpm_BaseParaManager.h の 31 行で定義されています。

7.9 cpm_BaseParaManager_Alloc.cpp

```
#include <stdlib.h>
#include "cpm_BaseParaManager.h"
```

cpm_BaseParaManager_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

日付

2012/05/31

cpm_BaseParaManager_Alloc.cpp で定義されています。

7.10 cpm_BaseParaManager_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。

7.10.1 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

日付

2012/05/31

cpm_BaseParaManager_inline.h で定義されています。

7.11 cpm_BaseParaManager_MPI.cpp

```
#include "stdlib.h"
#include "cpm_BaseParaManager.h"
#include <unistd.h>
```

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

7.11.1 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

日付

2012/05/31

cpm_BaseParaManager_MPI.cpp で定義されています。

7.12 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_S3D_PAD(_I, _J, _K, _NI, _NJ, _NK, _VC, _IP, _JP, _KP)
- #define _IDX_S4D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC)
- #define _IDX_S4D_PAD(_I, _J, _K, _N, _NI, _NJ, _NK, _VC, _IP, _JP, _KP)
- #define _IDX_V3D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC) (_IDX_S4D(_I,_J,_K,_N,_NI,_NJ,_NK,_VC))
- #define _IDX_V3D_PAD(_I, _J, _K, _N, _NI, _NJ, _NK, _VC, _IP, _JP, _KP) (_IDX_S4D_PAD(_I,_J,_K,_N,-_NI,_NJ,_NK,_VC,_IP,_JP,_KP))
- #define _IDX_S4DEX(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC)
- #define _IDX_S4DEX_PAD(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC, _NP, _IP, _JP, _KP)
- #define _IDX_V3DEX(_N, _I, _J, _K, _NI, _NJ, _NK, _VC) (_IDX_S4DEX(_N,_I,_J,_K,3,_NI,_NJ,_NK,_VC))
- #define _IDX_V3DEX_PAD(_N, _I, _J, _K, _NI, _NJ, _NK, _VC, _NP, _IP, _JP, _KP) (_IDX_S4DEX_PAD(_-N,_I,_J,_K,3,_NI,_NJ,_NK,_VC,_NP,_IP,_JP,_KP))
- #define stmpd_printf printf("%s (%d): ",__FILE__, __LINE__); printf

列挙型

- enum cpm_DefPointType { CPM_DEFPOINTTYPE_UNKNOWN = -1, CPM_DEFPOINTTYPE_FVM = 0, CPM_DEFPOINTTYPE_FDM = 1 }
- enum cpm_DomainType { CPM_DOMAIN_UNKNOWN = -1, CPM_DOMAIN_CARTESIAN = 0, CPM_DOMAIN_LMR = 1 }
- 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_DivPolicy { DIV_COMM_SIZE = 0, DIV_VOX_CUBE = 1 }
- enum cpm_ErrorCode {
  CPM_SUCCESS = 0, CPM_ERROR = 1000, CPM_ERROR_PM_INSTANCE = 1001, CPM_ERROR_INVALID_PTR = 1002,
  CPM_ERROR_INVALID_DOMAIN_NO = 1003, CPM_ERROR_INVALID_OBJKEY = 1004, CPM_ERROR_REGIST_OBJKEY = 1005, CPM_ERROR_TEXTPARSER = 2000,
  CPM_ERROR_NO_TEXTPARSER = 2001, CPM_ERROR_TP_NOVECTOR = 2002, CPM_ERROR_TP_VECTOR_SIZE = 2003, CPM_ERROR_TP_INVALID_G_ORG = 2004,
  CPM_ERROR_TP_INVALID_G_VOXEL = 2005, CPM_ERROR_TP_INVALID_G_PITCH = 2006,
  CPM_ERROR_TP_INVALID_G_RGN = 2007, CPM_ERROR_TP_INVALID_G_DIV = 2008,
  CPM_ERROR_TP_INVALID_POS = 2009, CPM_ERROR_TP_LMR_DOMAINFILE = 2100, CPM_ERROR_TP_LMR_DOMAIN = 2101, CPM_ERROR_TP_LMR_BCMTREE = 2102,
  CPM_ERROR_TP_LMR_LEAFBLOCK = 2103, CPM_ERROR_TP_LMR_UNIT = 2104, CPM_ERROR_TP_LMR_SIZE_NOT_E

= 2105, CPM_ERROR_VOXELINIT = 3000,
CPM_ERROR_NOT_IN_PROCGROUP = 3001, CPM_ERROR_ALREADY_VOXELINIIT = 3002,
CPM_ERROR_MISMATCH_NP_SUBDOMAIN = 3003, CPM_ERROR_CREATE_RANKMAP = 3004,
CPM_ERROR_CREATE_NEIGHBOR = 3005, CPM_ERROR_CREATE_LOCALDOMAIN = 3006,
CPM_ERROR_INSERT_VOXELMAP = 3007, CPM_ERROR_CREATE_PROCGROUP = 3008,
CPM_ERROR_INVALID_VOXELSIZE = 3009, CPM_ERROR_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_NUMDOMAIN_ZERO = 3017, CPM_ERROR_MISMATCH_DIV_SUBDOMAIN =
3018, CPM_ERROR_DECIDE_DIV_PATTERN = 3019, CPM_ERROR_ALREADY_NODEINIT = 3020,
CPM_ERROR_INVALID_NODESIZE = 3021, CPM_ERROR_INSERT_DEFPOINTTYPEMAP = 3022,
CPM_ERROR_DOMAINTYPE_VOXELINIT = 3100, CPM_ERROR_DOMAINTYPE_SETBNDCOMMBUF
= 3101,
CPM_ERROR_DOMAINTYPE_NODEINIT = 3102, CPM_ERROR_VOXELINIT_LMR = 3200, CPM_ERROR_LMR_OPEN_OC
= 3201, CPM_ERROR_LMR_INVALID_OCTFILE = 3202,
CPM_ERROR_LMR_READ_OCT_HEADER = 3203, CPM_ERROR_LMR_READ_OCT_PEDIGREE =
3204, CPM_ERROR_LMR_MISMATCH_NP_NUMLEAF = 3205, CPM_ERROR_GET_INFO = 4000,
CPM_ERROR_GET_DIVNUM = 4001, CPM_ERROR_GET_PITCH = 4002, CPM_ERROR_GET_GLOBALVOXELSIZE
= 4003, CPM_ERROR_GET_GLOBALORIGIN = 4004,
CPM_ERROR_GET_GLOBALREGION = 4005, CPM_ERROR_GET_LOCALVOXELSIZE = 4006,
CPM_ERROR_GET_LOCALORIGIN = 4007, CPM_ERROR_GET_LOCALREGION = 4008,
CPM_ERROR_GET_DIVPOS = 4009, CPM_ERROR_GET_HEADINDEX = 4011, CPM_ERROR_GET_TAILINDEX
= 4012, CPM_ERROR_GET_NEIGHBOR_RANK = 4013,
CPM_ERROR_GET_PERIODIC_RANK = 4014, CPM_ERROR_GET_MYRANK = 4015, CPM_ERROR_GET_NUMRANK
= 4016, CPM_ERROR_GET_GLOBALNODESIZE = 4017,
CPM_ERROR_GET_GLOBALARRAYSIZE = 4018, CPM_ERROR_GET_LOCALNODESIZE = 4019,
CPM_ERROR_GET_LOCALARRAYSIZE = 4020, 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_E
= 9503, CPM_ERROR_BNDCOMM_ALLOC_BUFFER = 9504,
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 }
- enum CPM_ARRAY_SHAPE {
  CPM_ARRAY_UNKNOWN = -1, CPM_ARRAY_S3D = 0, CPM_ARRAY_V3D = 1, CPM_ARRAY_V3DEX =
  2,
  CPM_ARRAY_S4D = 3, CPM_ARRAY_S4DEX = 4 }
- enum CPM_PADDING { CPM_PADDING_ON = true, CPM_PADDING_OFF = false }

7.12.1 説明

CPM の定義マクロ記述ヘッダーファイル

日付

2012/05/31

cpm_Define.h で定義されています。

7.12.2 マクロ定義

7.12.2.1 #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	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_Define.h の 48 行で定義されています。

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

7.12.2.2 #define _IDX_S3D_PAD(_I, _J, _K, _NI, _NJ, _NK, _VC, _IP, _JP, _KP)

値:

```
( (long long)(_K+_VC) * (long long)(_NI+2*(_VC)+_IP) * (long long)(_NJ+2*(_VC)+_JP) \
+ (long long)(_J+_VC) * (long long)(_NI+2*(_VC)+_IP) \
+ (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 (パディング対応)

引数

in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数
in	*_IP*	i 方向パディング数
in	*_JP*	j 方向パディング数
in	*_KP*	k 方向パディング数

戻り値

1 次元インデクス

cpm_Define.h の 67 行で定義されています。

7.12.2.3 #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	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_N*	成分インデクス
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_Define.h の 84 行で定義されています。

7.12.2.4 #define _IDX_S4D_PAD(*_I, _J, _K, _N, _NI, _NJ, _NK, _VC, _IP, _JP, _KP*)

値:

```
( (long long)(_N) * (long long)(_NI+2*(_VC)+_IP) * (long long)(_NJ+2*(_VC)+_JP) * (long long)(_NK+2*(_VC)+
      _KP) \
+ _IDX_S3D_PAD(_I,_J,_K,_NI,_NJ,_NK,_VC,_IP,_JP,_KP) \
)
```

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 (パディング対応)

引数

in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_N*	成分インデクス
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数
in	*_IP*	i 方向パディング数
in	*_JP*	j 方向パディング数
in	*_KP*	k 方向パディング数

戻り値

1 次元インデクス

cpm_Define.h の 103 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().

7.12.2.5 #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	*_N*	成分インデクス
in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_NN*	成分数
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_Define.h の 147 行で定義されています。

7.12.2.6 #define _IDX_S4DEX_PAD(*_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC, _NP, _IP, _JP, _KP*)

値:

```
( (long long)(_NN+_NP) * _IDX_S3D_PAD(_I,_J,_K,_NI,_NJ,_NK,_VC,_IP,_JP,_KP) \
+ (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 (パディング対応)

引数

in	*_N*	成分インデクス
in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_NN*	成分数
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数
in	*_NP*	成分パディング数
in	*_IP*	i 方向パディング数
in	*_JP*	j 方向パディング数
in	*_KP*	k 方向パディング数

戻り値

1 次元インデクス

cpm_Define.h の 167 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().

7.12.2.7 #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	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_N*	成分インデクス
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

cpm_Define.h の 118 行で定義されています。

7.12.2.8 #define _IDX_V3D_PAD(*_I, _J, _K, _N, _NI, _NJ, _NK, _VC, _IP, _JP, _KP*) (_IDX_S4D_PAD(_I,_J,_K,_N,_NI,_NJ,_NK,_VC,_IP,_JP,_KP))

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 (パディング対応)

引数

in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_N*	成分インデクス
in	*_NI*	i 方向インデクスサイズ

in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数
in	*_IP*	i 方向パディング数
in	*_JP*	j 方向パディング数
in	*_KP*	k 方向パディング数

cpm_Define.h の 133 行で定義されています。

7.12.2.9 #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	*_N*	成分インデクス
in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_Define.h の 182 行で定義されています。

7.12.2.10 #define _IDX_V3DEX_PAD(_N, _I, _J, _K, _NI, _NJ, _NK, _VC, _NP, _IP, _JP, _KP) (_IDX_S4DEX_PAD(_N,_I,_J,_K,3,_NI,_NJ,_NK,_VC,_NP,_IP,_JP,_KP))

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 (パディング対応)

引数

in	*_N*	成分インデクス
in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数
in	*_NP*	成分パディング数
in	*_IP*	i 方向パディング数
in	*_JP*	j 方向パディング数
in	*_KP*	k 方向パディング数

戻り値

1 次元インデクス

cpm_Define.h の 199 行で定義されています。

7.12.2.11 #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 34 行で定義されています。

参照元 S_BNDCOMM_BUFFER::CalcBufferSize(), cpm_ParaManagerLMR::GetBndCommBufferSize(), cpm_LeafCommInfo::SetBndCommBuffer(), と cpm_ParaManager::SetBndCommBuffer().

7.12.2.12 #define stmpd_printf printf("%s (%d): ",__FILE__, __LINE__); printf

デバッグライト用

cpm_Define.h の 445 行で定義されています。

参照元 cpm_ParaManagerLMR::packMX(), cpm_ParaManagerLMR::packMXEx(), cpm_ParaManagerLMR::packMY(), cpm_ParaManagerLMR::packMYEx(), cpm_ParaManagerLMR::packMZ(), cpm_ParaManagerLMR::packMZEx(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::packPXEx(), cpm_ParaManagerLMR::packPY(), cpm_ParaManagerLMR::packPYEx(), cpm_ParaManagerLMR::packPZ(), と cpm_ParaManagerLMR::packPZEx().

7.12.3 列挙型

7.12.3.1 enum CPM_ARRAY_SHAPE

配列形状タイプ

列挙型の値

CPM_ARRAY_UNKNOWN 不定

CPM_ARRAY_S3D Scalar3D {imax,jmax,kmax}.

CPM_ARRAY_V3D Vector3D {imax,jmax,kmax,3}.

CPM_ARRAY_V3DEX Vector3DEx {3,imax,jmax,kmax}.

CPM_ARRAY_S4D Scalar4D {imax,jmax,kmax,n}.

CPM_ARRAY_S4DEX Scalar4DEx {n,imax,jmax,kmax}.

cpm_Define.h の 427 行で定義されています。

7.12.3.2 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 383 行で定義されています。

7.12.3.3 enum cpm_DefPointType

定義点タイプ

列挙型の値

CPM_DEFPOINTTYPE_UNKNOWN 未定義
CPM_DEFPOINTTYPE_FVM ボクセル
CPM_DEFPOINTTYPE_FDM ノード

cpm_Define.h の 203 行で定義されています。

7.12.3.4 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 231 行で定義されています。

7.12.3.5 enum cpm_DivPolicy

自動分割ポリシー

列挙型の値

DIV_COMM_SIZE 通信面総数が小さくなるように
DIV_VOX_CUBE サブドメインが立方体に近くなるように

cpm_Define.h の 247 行で定義されています。

7.12.3.6 enum cpm_DomainType

領域分割タイプ

列挙型の値

CPM_DOMAIN_UNKNOWN 未定義
CPM_DOMAIN_CARTESIAN カーテシアン
CPM_DOMAIN_LMR LMR(Local Mesh Refinement)

cpm_Define.h の 212 行で定義されています。

7.12.3.7 enum cpm_ErrorCode

CPM のエラーコード

列挙型の値

CPM_SUCCESS 正常終了
CPM_ERROR その他のエラー
CPM_ERROR_PM_INSTANCE 並列管理クラス cpm_ParaManager のインスタンス失敗
CPM_ERROR_INVALID_PTR ポインタのエラー
CPM_ERROR_INVALID_DOMAIN_NO 領域番号が不正
CPM_ERROR_INVALID_OBJKEY 指定登録番号のオブジェクトが存在しない
CPM_ERROR_REGIST_OBJKEY オブジェクト登録に失敗:
CPM_ERROR_TEXTPARSER テキストパーサーに関するエラー
CPM_ERROR_NO_TEXTPARSER テキストパーサーを組み込んでいない
CPM_ERROR_TP_NOVECTOR 領域分割情報ファイルのベクトルデータ読み込みエラー
CPM_ERROR_TP_VECTOR_SIZE 領域分割情報ファイルのベクトルデータのサイズが不正
CPM_ERROR_TP_INVALID_G_ORG 領域分割情報ファイルのドメイン原点情報が不正
CPM_ERROR_TP_INVALID_G_VOXEL 領域分割情報ファイルのドメインVOXEL 数情報が不正
CPM_ERROR_TP_INVALID_G_PITCH 領域分割情報ファイルのドメインピッチ情報が不正
CPM_ERROR_TP_INVALID_G_RGN 領域分割情報ファイルのドメイン空間サイズ情報が不正
CPM_ERROR_TP_INVALID_G_DIV 領域分割情報ファイルのドメイン領域分割数情報が不正
CPM_ERROR_TP_INVALID_POS 領域分割情報ファイルのサブドメイン位置情報が不正
CPM_ERROR_TP_LMR_DOMAINFILE LMR 領域分割情報ファイルの読み込みエラー
CPM_ERROR_TP_LMR_DOMAIN LMR 領域分割情報ファイルのDomain ブロック読み込みエラー
CPM_ERROR_TP_LMR_BCMTREE LMR 領域分割情報ファイルのBCMTree ブロック読み込みエラー
CPM_ERROR_TP_LMR_LEAFBLOCK LMR 領域分割情報ファイルのLeafBlock 読み込みエラー
CPM_ERROR_TP_LMR_UNIT LMR 領域分割情報ファイルのLeafBlock/Unit 読み込みエラー
CPM_ERROR_TP_LMR_SIZE_NOT_EVEN LMR 領域分割情報ファイルのLeafBlock/Size が偶数でない
CPM_ERROR_VOXELINIT VoxelInit 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_VOXELINIIT 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_DECIDE_DIV_PATTERN 自動領域分割が不可能なパターン

CPM_ERROR_ALREADY_NODEINIT 指定されたプロセスグループが既に領域分割済み:

CPM_ERROR_INVALID_NODESIZE 頂点数が不正

CPM_ERROR_INSERT_DEFPOINTTYPEMAP 定義点管理のマップへの登録失敗

CPM_ERROR_DOMAINTYPE_VOXELINIT 領域分割タイプと対応しないVoxelInit がコールされた

CPM_ERROR_DOMAINTYPE_SETBNDCOMMBUF 領域分割タイプと対応しないSetBndCommBuffer がコールされた

CPM_ERROR_DOMAINTYPE_NODEINIT 領域分割タイプと対応しないNodeInit がコールされた

CPM_ERROR_VOXELINIT_LMR VoxelInit_LMR でエラー

CPM_ERROR_LMR_OPEN_OCTFILE LMR 用木情報ファイルオープンエラー

CPM_ERROR_LMR_INVALID_OCTFILE LMR 用木情報ファイルのエンディアン識別子が不正

CPM_ERROR_LMR_READ_OCT_HEADER LMR 用木情報ファイルのヘッダー情報読み込みエラー

CPM_ERROR_LMR_READ_OCT_PEDIGREE LMR 用木情報ファイルのペディグリー情報読み込みエラー

CPM_ERROR_LMR_MISMATCH_NP_NUMLEAF LMR でリーフ数と並列数が一致しない

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 始点インデクスの取得エラー

CPM_ERROR_GET_TAILINDEX 終点インデクスの取得エラー

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_GET_GLOBALNODESIZE 全体頂点数の取得エラー

CPM_ERROR_GET_GLOBALARRAYSIZE 全体ボクセル数または頂点数の取得エラー

CPM_ERROR_GET_LOCALNODESIZE 自ランクの頂点数の取得エラー

CPM_ERROR_GET_LOCALARRAYSIZE 自ランクのボクセル数または頂点数の取得エラー

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_BNDCOMM_ALLOC_BUFFER 袖通信バッファ領域確保でエラー
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 254 行で定義されています。

7.12.3.8 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 220 行で定義されています。

7.12.3.9 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 410 行で定義されています。

7.12.3.10 enum CPM_PADDING

列挙型の値

CPM_PADDING_ON

CPM_PADDING_OFF

cpm_Define.h の 437 行で定義されています。

7.12.3.11 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 239 行で定義されています。

7.13 cpm_DefineLMR.h

```
#include "cpm_Define.h"
```

cpm_DefineLMR.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 _IDX_S3D_LMR(_I, _J, _K, _IL, _NI, _NJ, _NK, _VC) (_IDX_S4D(_I,_J,_K,_IL,_NI,_NJ,_NK,_VC))
- #define _IDX_S4D_LMR(_I, _J, _K, _N, _IL, _NI, _NJ, _NK, _NN, _VC)
- #define _IDX_V3D_LMR(_I, _J, _K, _N, _IL, _NI, _NJ, _NK, _VC) (_IDX_S4D_LMR(_I,_J,_K,_N,_IL,_NI,_NJ,_NK,3,_VC))
- #define _IDX_S4DEX_LMR(_N, _I, _J, _K, _IL, _NN, _NI, _NJ, _NK, _VC)
- #define _IDX_V3DEX_LMR(_N, _I, _J, _K, _IL, _NI, _NJ, _NK, _VC) (_IDX_S4DEX_LMR(_N,_I,_J,_K,_IL,3,_NI,_NJ,_NK,_VC))

7.13.1 説明

CPM-LMR 用の定義マクロ記述ヘッダーファイル

日付

2012/05/31

cpm_DefineLMR.h で定義されています。

7.13.2 マクロ定義

7.13.2.1 #define _IDX_S3D_LMR(_I, _J, _K, _IL, _NI, _NJ, _NK, _VC) (_IDX_S4D(_I,_J,_K,_IL,_NI,_NJ,_NK,_VC))

3 次元インデクス (i,j,k,iLeaf) -> 1 次元インデクス変換マクロ

引数

in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_IL*	ローカルリーフ順番号 (0〜)
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_DefineLMR.h の 34 行で定義されています。

7.13.2.2 #define _IDX_S4D_LMR(_I, _J, _K, _N, _IL, _NI, _NJ, _NK, _NN, _VC)

値:

```
( (long long)(_NN) * (long long)(_NI+2*(_VC)) * (long long)(_NJ+2*(_VC)) * (long long)(_NK+2*(_VC)) \
* (long long)(_IL) \
+ _IDX_S4D(_I,_J,_K,_N,_NI,_NJ,_NK,_VC) \
)
```

4 次元インデクス (i,j,k,n,iLeaf) -> 1 次元インデクス変換マクロ

引数

in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_N*	成分インデクス
in	*_IL*	ローカルリーフ順番号 (0〜)
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ

in	*_NK*	k 方向インデクスサイズ
in	*_NN*	成分数
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_DefineLMR.h の 49 行で定義されています。

参照元 cpm_ParaManagerLMR::packMX(), cpm_ParaManagerLMR::packMY(), cpm_ParaManagerLMR::packMZ(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::packPY(), cpm_ParaManagerLMR::packPZ(), cpm_ParaManagerLMR::unpackMX(), cpm_ParaManagerLMR::unpackMY(), cpm_ParaManagerLMR::unpackMZ(), cpm_ParaManagerLMR::unpackPX(), cpm_ParaManagerLMR::unpackPY(), と cpm_ParaManagerLMR::unpackPZ().

7.13.2.3 #define _IDX_S4DEX_LMR(*_N, _I, _J, _K, _IL, _NN, _NI, _NJ, _NK, _VC*)

値:

```
( (long long)(_NN) * (long long)(_NI+2*(_VC)) * (long long)(_NJ+2*(_VC)) * (long long)(_NK+2*(_VC)) \
* (long long)(_IL) \
+ _IDX_S4DEX(_N,_I,_J,_K,_NN,_NI,_NJ,_NK,_VC) \
)
```

4 次元インデクス (n,i,j,k,iLeaf) -> 1 次元インデクス変換マクロ

引数

in	*_N*	成分インデクス
in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_IL*	ローカルリーフ順番号 (0〜)
in	*_NN*	成分数
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_DefineLMR.h の 81 行で定義されています。

参照元 cpm_ParaManagerLMR::packMXEx(), cpm_ParaManagerLMR::packMYEx(), cpm_ParaManagerLMR::packMZEx(), cpm_ParaManagerLMR::packPXEx(), cpm_ParaManagerLMR::packPYEx(), cpm_ParaManagerLMR::packPZEx(), cpm_ParaManagerLMR::unpackMXEx(), cpm_ParaManagerLMR::unpackMYEx(), cpm_ParaManagerLMR::unpackMZEx(), cpm_ParaManagerLMR::unpackPXEx(), cpm_ParaManagerLMR::unpackPYEx(), と cpm_ParaManagerLMR::unpackPZEx().

7.13.2.4 #define _IDX_V3D_LMR(*_I, _J, _K, _N, _IL, _NI, _NJ, _NK, _VC*) (_IDX_S4D_LMR(_I,_J,_K,_N,_IL,_NI,_NJ,_NK,3,_VC))

3 次元インデクス (i,j,k,3,iLeaf) -> 1 次元インデクス変換マクロ

引数

in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_N*	成分インデクス
in	*_IL*	ローカルリーフ順番号 (0〜)
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

cpm_DefineLMR.h の 66 行で定義されています。

7.13.2.5 #define _IDX_V3DEX_LMR(*_N, _I, _J, _K, _IL, _NI, _NJ, _NK, _VC*) (_IDX_S4DEX_LMR(_N,_I,_J,_K,_IL,3,_NI,_NJ,_NK,_VC))

3 次元インデクス (3,i,j,k,iLeaf) -> 1 次元インデクス変換マクロ

引数

in	*_N*	成分インデクス
in	*_I*	i 方向インデクス
in	*_J*	j 方向インデクス
in	*_K*	k 方向インデクス
in	*_IL*	ローカルリーフ順番号 (0〜)
in	*_NI*	i 方向インデクスサイズ
in	*_NJ*	j 方向インデクスサイズ
in	*_NK*	k 方向インデクスサイズ
in	*_VC*	仮想セル数

戻り値

1 次元インデクス

cpm_DefineLMR.h の 99 行で定義されています。

7.14 cpm_DomainInfo.cpp

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

cpm_DomainInfo.cpp のインクルード依存関係図

7.14.1 説明

DomainInfo クラスのソースファイル

日付

2012/05/31

cpm_DomainInfo.cpp で定義されています。

7.15 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

7.15.1 説明

領域情報クラスのヘッダーファイル

日付

2012/05/31

cpm_DomainInfo.h で定義されています。

7.16 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

- 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)

7.16.1 説明

CPM エンディアンユーティリティヘッダーファイル

日付

2013/04/02

cpm_EndianUtil.h で定義されています。

7.17 cpm_LeafCommInfo.cpp

```
#include "cpm_LeafCommInfo.h"
```
cpm_LeafCommInfo.cpp のインクルード依存関係図

7.17.1 説明

LMR の袖通信情報管理クラスソースファイル

日付

2016/07/29

cpm_LeafCommInfo.cpp で定義されています。

7.18 cpm_LeafCommInfo.h

```
#include "cpm_Base.h"
#include "cpm_DefineLMR.h"
#include <vector>
```
cpm_LeafCommInfo.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_LeafCommInfo
- struct cpm_LeafCommInfo::stCommInfo

7.18.1 説明

LMR の袖通信情報管理クラスヘッダー

日付

2016/07/29

cpm_LeafCommInfo.h で定義されています。

7.19 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

7.19.1 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

日付

2012/05/31

cpm_ObjList.h で定義されています。

7.19.2 型定義

7.19.2.1 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 27 行で定義されています。

7.20 cpm_ParaManager.cpp

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

cpm_ParaManager.cpp のインクルード依存関係図

7.20.1 説明

カーテシアン用パラレルマネージャクラスのソースファイル

日付

2015/03/27

cpm_ParaManager.cpp で定義されています。

7.21 cpm_ParaManager.h

```
#include "cpm_BaseParaManager.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

7.21.1 説明

カーテシアン用のパラレルマネージャクラスのヘッダーファイル

日付

2015/03/27

cpm_ParaManager.h で定義されています。

7.21.2 型定義

7.21.2.1 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 24 行で定義されています。

7.22 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

7.22.1 説明

LMR パラレルマネージャクラスのソースファイル

日付

2012/05/31

cpm_ParaManager_Alloc.cpp で定義されています。

7.23 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)

7.23.1 説明

カーテシアン用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

日付

2015/03/27

cpm_ParaManager_BndComm.h で定義されています。

7.23.2 マクロ定義

7.23.2.1 #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 21 行で定義されています。

参照元 cpm_ParaManager::packX(), と cpm_ParaManager::unpackX().

7.23.2.2 #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 28 行で定義されています。

参照元 cpm_ParaManager::packY(), と cpm_ParaManager::unpackY().

7.23.2.3 #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 35 行で定義されています。

参照元 cpm_ParaManager::packZ(), と cpm_ParaManager::unpackZ().

7.24 cpm_ParaManager_BndCommEx.h

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

マクロ定義

- #define _IDXFX(_N, _I, _J, _K, _NN, _IS, _NJ, _NK, _VC)
- #define _IDXFY(_N, _I, _J, _K, _NN, _NI, _JS, _NK, _VC)
- #define _IDXFZ(_N, _I, _J, _K, _NN, _NI, _NJ, _KS, _VC)

7.24.1 説明

カーテシアン用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

日付

2012/05/31

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

7.24.2 マクロ定義

7.24.2.1 #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 21 行で定義されています。

参照元 cpm_ParaManager::packXEx(), と cpm_ParaManager::unpackXEx().

7.24.2.2 #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 30 行で定義されています。

参照元 cpm_ParaManager::packYEx(), と cpm_ParaManager::unpackYEx().

7.24.2.3 #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 39 行で定義されています。

参照元 cpm_ParaManager::packZEx(), と cpm_ParaManager::unpackZEx().

7.25 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_NodeInit_ cpm_nodeinit_
- #define cpm_NodeInit_nodiv_ cpm_nodeinit_nodiv_
- #define cpm_IsParallel_ cpm_isparallel_
- #define cpm_GetDivNum_ cpm_getdivnum_
- #define cpm_GetPitch_ cpm_getpitch_
- #define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_
- #define cpm_GetGlobalNodeSize_ cpm_getglobalnodesize_
- #define cpm_GetGlobalArraySize_ cpm_getglobalarraysize_
- #define cpm_GetGlobalOrigin_ cpm_getglobalorigin_
- #define cpm_GetGlobalRegion_ cpm_getglobalregion_
- #define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_
- #define cpm_GetLocalNodeSize_ cpm_getlocalnodesize_
- #define cpm_GetLocalArraySize_ cpm_getlocalarraysize_
- #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_GetNodeHeadIndex_ cpm_getnodeheadindex_
- #define cpm_GetNodeTailIndex_ cpm_getnodetailindex_
- #define cpm_GetArrayHeadIndex_ cpm_getarrayheadindex_
- #define cpm_GetArrayTailIndex_ cpm_getarraytailindex_
- #define cpm_GetDefPointType_ cpm_getdefpointtype_
- #define cpm_GetNeighborRankID_ cpm_getneighborrankid_
- #define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_
- #define cpm_GetMyRankID_ cpm_getmyrankid_
- #define cpm_GetNumRank_ cpm_getnumrank_
- #define cpm_Abort_ cpm_abort_
- #define cpm_Barrier_ cpm_barrier_

- #define cpm_Wait_ cpm_wait_
- #define cpm_Waitall_ cpm_waitall_
- #define cpm_Bcast_ cpm_bcast_
- #define cpm_Send_ cpm_send_
- #define cpm_Recv_ cpm_recv_
- #define cpm_Isend_ cpm_isend_
- #define cpm_Irecv_ cpm_irecv_
- #define cpm_Allreduce_ cpm_allreduce_
- #define cpm_Gather_ cpm_gather_
- #define cpm_Allgather_ cpm_allgather_
- #define cpm_Gatherv_ cpm_gatherv_
- #define cpm_Allgatherv_ cpm_allgatherv_
- #define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_
- #define cpm_BndCommS3D_ cpm_bndcomms3d_
- #define cpm_BndCommV3D_ cpm_bndcommv3d_
- #define cpm_BndCommS4D_ cpm_bndcomms4d_
- #define cpm_BndCommS3D_nowait_ cpm_bndcomms3d_nowait_
- #define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_
- #define cpm_BndCommS4D_nowait_ cpm_bndcomms4d_nowait_
- #define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_
- #define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_
- #define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_
- #define cpm_BndCommV3DEx_ cpm_bndcommv3dex_
- #define cpm_BndCommS4DEx_ cpm_bndcomms4dex_
- #define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_
- #define cpm_BndCommS4DEx_nowait_ cpm_bndcomms4dex_nowait_
- #define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_
- #define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_
- #define cpm_PeriodicCommS3D cpm_periodiccomms3d_
- #define cpm_PeriodicCommV3D cpm_periodiccommv3d_
- #define cpm_PeriodicCommS4D cpm_periodiccomms4d_
- #define cpm_PeriodicCommV3DEx cpm_periodiccommv3dex_
- #define cpm_PeriodicCommS4DEx cpm_periodiccomms4dex_

関数

- CPM_EXTERN void cpm_Initialize_ (int ∗ierr)
- CPM_EXTERN void cpm_VoxelInit_ (int ∗div, int ∗vox, double ∗origin, double ∗region, int ∗maxVC, int ∗maxN, int ∗procGrpNo, int ∗ierr)
- CPM_EXTERN void cpm_NodeInit_ (int ∗div, int ∗nod, double ∗origin, double ∗region, int ∗maxVC, int ∗maxN, int ∗procGrpNo, int ∗ierr)
- CPM_EXTERN void cpm_VoxelInit_nodiv_ (int ∗vox, double ∗origin, double ∗region, int ∗maxVC, int ∗maxN, int ∗divPolicy, int ∗procGrpNo, int ∗ierr)
- CPM_EXTERN void cpm_NodeInit_nodiv_ (int ∗nod, double ∗origin, double ∗region, int ∗maxVC, int ∗maxN, int ∗divPolicy, 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_GetGlobalNodeSize_ (int ∗wsz, int ∗procGrpNo, int ∗ierr)
- CPM_EXTERN void cpm_GetGlobalArraySize_ (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_GetLocalNodeSize_ (int *lsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalArraySize_ (int *lsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalOrigin_ (double *lorg, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalRegion_ (double *lrgn, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetDivPos_ (int *pos, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetVoxelHeadIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetVoxelTailIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNodeHeadIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNodeTailIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetArrayHeadIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetArrayTailIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetDefPointType_ (int *ideftyp, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNeighborRankID_ (int *nID, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetPeriodicRankID_ (int *nID, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetMyRankID_ (int *id, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNumRank_ (int *nrank, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Abort_ (int *errorcode)
- CPM_EXTERN void cpm_Barrier_ (int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Wait_ (int *reqNo, int *ierr)
- CPM_EXTERN void cpm_Waitall_ (int *count, int *reqlist, int *ierr)
- CPM_EXTERN void cpm_Bcast_ (void *buf, int *count, int *datatype, int *root, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Send_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Recv_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Isend_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *reqNo, int *ierr)
- CPM_EXTERN void cpm_Irecv_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *reqNo, int *ierr)
- CPM_EXTERN void cpm_Allreduce_ (void *sendbuf, void *recvbuf, int *count, int *datatype, int *op, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Gather_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *root, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Allgather_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Gatherv_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnts, int *displs, int *recvtype, int *root, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Allgatherv_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnts, int *displs, int *recvtype, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_SetBndCommBuffer_ (int *maxVC, int *maxN, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_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)

7.25.1 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

日付

2012/05/31

cpm_ParaManager_frtIF.cpp で定義されています。

7.25.2 マクロ定義

7.25.2.1 #define cpm_Abort_ cpm_abort_

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

7.25.2.2 #define cpm_Allgather_ cpm_allgather_

cpm_ParaManager_frtIF.cpp の 77 行で定義されています。

7.25.2.3 #define cpm_Allgatherv_ cpm_allgatherv_

cpm_ParaManager_frtIF.cpp の 79 行で定義されています。

7.25.2.4 #define cpm_Allreduce_ cpm_allreduce_

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

7.25.2.5 #define cpm_Barrier_ cpm_barrier_

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

7.25.2.6 #define cpm_Bcast_ cpm_bcast_

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

7.25.2.7 #define cpm_BndCommS3D_ cpm_bndcomms3d_

cpm_ParaManager_frtIF.cpp の 81 行で定義されています。

7.25.2.8 #define cpm_BndCommS3D_nowait_ cpm_bndcomms3d_nowait_

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

7.25.2.9 #define cpm_BndCommS4D_ cpm_bndcomms4d_

cpm_ParaManager_frtIF.cpp の 83 行で定義されています。
参照元 cpm_BndCommS3D_(), と cpm_BndCommV3D_().

7.25.2.10 #define cpm_BndCommS4D_nowait_ cpm_bndcomms4d_nowait_

cpm_ParaManager_frtIF.cpp の 86 行で定義されています。
参照元 cpm_BndCommS3D_nowait_(), と cpm_BndCommV3D_nowait_().

7.25.2.11 #define cpm_BndCommS4DEx_ cpm_bndcomms4dex_

cpm_ParaManager_frtIF.cpp の 91 行で定義されています。
参照元 cpm_BndCommV3DEx_().

7.25.2.12 #define cpm_BndCommS4DEx_nowait_ cpm_bndcomms4dex_nowait_

cpm_ParaManager_frtIF.cpp の 93 行で定義されています。
参照元 cpm_BndCommV3DEx_nowait_().

7.25.2.13 #define cpm_BndCommV3D_ cpm_bndcommv3d_

cpm_ParaManager_frtIF.cpp の 82 行で定義されています。

7.25.2.14 #define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_

cpm_ParaManager_frtIF.cpp の 85 行で定義されています。

7.25.2.15 #define cpm_BndCommV3DEx_ cpm_bndcommv3dex_

cpm_ParaManager_frtIF.cpp の 90 行で定義されています。

7.25.2.16 #define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_

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

7.25.2.17 #define CPM_EXTERN extern "C"

extern 宣言

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

7.25.2.18 #define cpm_Gather_ cpm_gather_

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

7.25.2.19 #define cpm_Gatherv_ cpm_gatherv_

cpm_ParaManager_frtIF.cpp の 78 行で定義されています。

7.25.2.20 #define cpm_GetArrayHeadIndex_ cpm_getarrayheadindex_

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

7.25.2.21 #define cpm_GetArrayTailIndex_ cpm_getarraytailindex_

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

7.25.2.22 #define cpm_GetDefPointType_ cpm_getdefpointtype_

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

7.25.2.23 #define cpm_GetDivNum_ cpm_getdivnum_

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

7.25.2.24 #define cpm_GetDivPos_ cpm_getdivpos_

cpm_ParaManager_frtIF.cpp の 52 行で定義されています。

7.25.2.25 #define cpm_GetGlobalArraySize_ cpm_getglobalarraysize_

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

7.25.2.26 #define cpm_GetGlobalNodeSize_ cpm_getglobalnodesize_

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

7.25.2.27 #define cpm_GetGlobalOrigin_ cpm_getglobalorigin_

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

7.25.2.28 #define cpm_GetGlobalRegion_ cpm_getglobalregion_

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

7.25.2.29 #define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_

cpm_ParaManager_frtIF.cpp の 38 行で定義されています。

7.25.2.30 #define cpm_GetLocalArraySize_ cpm_getlocalarraysize_

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

7.25.2.31 #define cpm_GetLocalNodeSize_ cpm_getlocalnodesize_

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

7.25.2.32 #define cpm_GetLocalOrigin_ cpm_getlocalorigin_

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

7.25.2.33 #define cpm_GetLocalRegion_ cpm_getlocalregion_

cpm_ParaManager_frtIF.cpp の 51 行で定義されています。

7.25.2.34 #define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_

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

7.25.2.35 #define cpm_GetMyRankID_ cpm_getmyrankid_

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

7.25.2.36 #define cpm_GetNeighborRankID_ cpm_getneighborrankid_

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

7.25.2.37 #define cpm_GetNodeHeadIndex_ cpm_getnodeheadindex_

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

7.25.2.38 #define cpm_GetNodeTailIndex_ cpm_getnodetailindex_

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

7.25.2.39 #define cpm_GetNumRank_ cpm_getnumrank_

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

7.25.2.40 #define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_

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

7.25.2.41 #define cpm_GetPitch_ cpm_getpitch_

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

7.25.2.42 #define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_

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

7.25.2.43 #define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_

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

7.25.2.44 #define cpm_Initialize_ cpm_initialize_

cpm_ParaManager_frtIF.cpp の 28 行で定義されています。

7.25.2.45 #define cpm_Irecv_ cpm_irecv_

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

7.25.2.46 #define cpm_Isend_ cpm_isend_

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

7.25.2.47 #define cpm_IsParallel_ cpm_isparallel_

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

7.25.2.48 #define cpm_NodeInit_ cpm_nodeinit_

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

7.25.2.49 #define cpm_NodeInit_nodiv_ cpm_nodeinit_nodiv_

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

7.25.2.50 #define cpm_PeriodicCommS3D cpm_periodiccomms3d_

cpm_ParaManager_frtIF.cpp の 96 行で定義されています。

7.25.2.51 #define cpm_PeriodicCommS4D cpm_periodiccomms4d_

cpm_ParaManager_frtIF.cpp の 98 行で定義されています。

7.25.2.52 #define cpm_PeriodicCommS4DEx cpm_periodiccomms4dex_

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

7.25.2.53 #define cpm_PeriodicCommV3D cpm_periodiccommv3d_

cpm_ParaManager_frtIF.cpp の 97 行で定義されています。

7.25.2.54 #define cpm_PeriodicCommV3DEx cpm_periodiccommv3dex_

cpm_ParaManager_frtIF.cpp の 99 行で定義されています。

7.25.2.55 #define cpm_Recv_ cpm_recv_

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

7.25.2.56 #define cpm_Send_ cpm_send_

cpm_ParaManager_frtIF.cpp の 71 行で定義されています。

7.25.2.57 #define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_

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

7.25.2.58 #define cpm_VoxelInit_ cpm_voxelinit_

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

7.25.2.59 #define cpm_VoxelInit_nodiv_ cpm_voxelinit_nodiv_

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

7.25.2.60 #define cpm_Wait_ cpm_wait_

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

7.25.2.61 #define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_

cpm_ParaManager_frtIF.cpp の 87 行で定義されています。

7.25.2.62 #define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_

cpm_ParaManager_frtIF.cpp の 89 行で定義されています。
参照元 cpm_wait_BndCommS3D_(), と cpm_wait_BndCommV3D_().

7.25.2.63 #define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_

cpm_ParaManager_frtIF.cpp の 95 行で定義されています。
参照元 cpm_wait_BndCommV3DEx_().

7.25.2.64 #define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_

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

7.25.2.65 #define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_

cpm_ParaManager_frtIF.cpp の 94 行で定義されています。

7.25.2.66 #define cpm_Waitall_ cpm_waitall_

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

7.25.3 関数

7.25.3.1 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 1432 行で定義されています。

参照先 cpm_BaseParaManager::Abort(), と cpm_ParaManager::get_instance().

7.25.3.2 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 1849 行で定義されています。

参照先 cpm_BaseParaManager::Allgather(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.3 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 1951 行で定義されています。

参照先 cpm_BaseParaManager::Allgatherv(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.4 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 1748 行で定義されています。

参照先 cpm_BaseParaManager::Allreduce(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_MPI_INVALID_OPERATOR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::GetMPI_Op().

7.25.3.5 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 1460 行で定義されています。

参照先 cpm_BaseParaManager::Barrier(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.25.3.6 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

in,out	*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 1549 行で定義されています。

参照先 cpm_BaseParaManager::Bcast(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.7 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

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

cpm_ParaManager_frtIF.cpp の 2079 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_PADDING_OFF, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.8 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

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

in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi を参照)
out	*reqlist*	リクエスト番号のリスト (サイズ 12)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2221 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().

7.25.3.9 CPM_EXTERN void cpm_BndCommS4D_ (void * *array,* int * *imax,* int * *jmax,* int * *kmax,* int * *nmax,* int * *vc,* int * *vc_comm,* int * *datatype,* int * *procGrpNo,* int * *ierr*)

袖通信 (Scalar4D 版) のFortran インターフェイス

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

引数

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

cpm_ParaManager_frtIF.cpp の 2034 行で定義されています。

参照先 cpm_ParaManager::BndCommS4D(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.10 CPM_EXTERN void cpm_BndCommS4D_nowait_ (void * *array,* int * *imax,* int * *jmax,* int * *kmax,* int * *nmax,* int * *vc,* int * *vc_comm,* int * *datatype,* int * *reqlist,* int * *procGrpNo,* int * *ierr*)

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

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う
- BndCommS4D_nowait のFortran インターフェイス関数

引数

in,out	*array*	袖通信をする配列の先頭ポインタ

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

cpm_ParaManager_frtIF.cpp の 2181 行で定義されています。

参照先 cpm_ParaManager::cpm_BndCommS4D_nowait(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.25.3.11 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

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

cpm_ParaManager_frtIF.cpp の 2444 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_BaseParaManager::GetMPI_Datatype().

7.25.3.12 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

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

cpm_ParaManager_frtIF.cpp の 2541 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().

7.25.3.13 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

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

cpm_ParaManager_frtIF.cpp の 2129 行で定義されています。

参照先 cpm_ParaManager::BndCommV3D(), cpm_BndCommS4D_, CPM_ERROR_INVALID_PTR, CPM_ERR-OR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, CPM_PADDING_OFF, cpm_ParaManager::get-_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.14 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

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

cpm_ParaManager_frtIF.cpp の 2267 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().

7.25.3.15 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

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

cpm_ParaManager_frtIF.cpp の 2489 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_BaseParaManager::GetMPI_Datatype().

7.25.3.16 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

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

cpm_ParaManager_frtIF.cpp の 2581 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().

7.25.3.17 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	*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	*root*	受信するランク番号 (procGrpNo 内でのランク番号)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1799 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_BaseParaManager::Gather(), cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.18 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	*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	*root*	受信するランク番号 (procGrpNo 内でのランク番号)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1900 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_BaseParaManager::Gatherv(), cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.25.3.19 CPM_EXTERN void cpm_GetArrayHeadIndex_ (int * *idx,* int * *procGrpNo,* int * *ierr*)

自ランクの始点ボクセルまたは頂点の全体空間でのインデクスを取得

- FVM のときはボクセル、FDM のときは頂点始点インデックスを取得
- GetArrayHeadIndex 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	*procGrpNo*	プロセスグループ番号
out	*idx*	自ランクの始点ボクセルまたは頂点インデクス (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 1147 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::GetArrayHeadIndex().

7.25.3.20 CPM_EXTERN void cpm_GetArrayTailIndex_ (int * *idx,* int * *procGrpNo,* int * *ierr*)

自ランクの終点ボクセルまたは頂点の全体空間でのインデクスを取得

- FVM のときはボクセル、FDM のときは頂点終点インデックスを取得
- GetArrayTailIndex 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	*procGrpNo*	プロセスグループ番号
out	*idx*	自ランクの終点ボクセルまたは頂点インデクス (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 1192 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::GetArrayTailIndex().

7.25.3.21 CPM_EXTERN void cpm_GetDefPointType_ (int * *ideftyp,* int * *procGrpNo,* int * *ierr*)

定義点タイプを取得

- GetDefPointType 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	*procGrpNo*	プロセスグループ番号
out	*ideftyp*	定義点タイプ (-1=未定義、0=FVM、1=FDM)
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 1236 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_BaseParaManager::GetDefPointType().

7.25.3.22 CPM_EXTERN void cpm_GetDivNum_ (int * *div,* int * *procGrpNo,* int * *ierr*)

領域分割数を取得

- GetDivNum のFortran インターフェイス関数

  引数

out	*div*	領域分割数 (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 418 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().

7.25.3.23 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 928 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().

7.25.3.24 CPM_EXTERN void cpm_GetGlobalArraySize_ (int * *wsz,* int * *procGrpNo,* int * *ierr*)

全体ボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数
- GetGlobalArraySize 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	*wsz*	全体ボクセル数または頂点数 (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 588 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALARRAYSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetGlobalArraySize().

7.25.3.25 CPM_EXTERN void cpm_GetGlobalNodeSize_ (int * *wsz,* int * *procGrpNo,* int * *ierr*)

全体頂点数を取得

- GetGlobalNodeSize 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	*wsz*	全体頂点数 (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 545 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALNODESIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetGlobalNodeSize().

7.25.3.26 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 631 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_BaseParaManager::GetGlobalOrigin().

7.25.3.27 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 673 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_BaseParaManager::GetGlobalRegion().

7.25.3.28 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	*procGrpNo*	プロセスグループ番号
out	*wsz*	全体ボクセル数 (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 502 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_BaseParaManager::GetGlobalVoxelSize().

7.25.3.29 CPM_EXTERN void cpm_GetLocalArraySize_ (int * *lsz,* int * *procGrpNo,* int * *ierr*)

自ランクのボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数
- GetLocalArraySize 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	*lsz*	自ランクの頂点数 (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 801 行で定義されています。

参照先 CPM_ERROR_GET_LOCALARRAYSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetLocalArraySize().

7.25.3.30 CPM_EXTERN void cpm_GetLocalNodeSize_ (int * *lsz,* int * *procGrpNo,* int * *ierr*)

自ランクの頂点数を取得

- GetLocalNodeSize 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	*lsz*	自ランクの頂点数 (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 758 行で定義されています。

参照先 CPM_ERROR_GET_LOCALNODESIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_BaseParaManager::GetLocalNodeSize().

7.25.3.31 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	*procGrpNo*	プロセスグループ番号
out	*lorg*	自ランクの空間原点 (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 844 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().

7.25.3.32 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	*procGrpNo*	プロセスグループ番号
out	*lrgn*	自ランクの空間サイズ (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 886 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::GetLocalRegion().

7.25.3.33 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	*procGrpNo*	プロセスグループ番号
out	*lsz*	自ランクのボクセル数 (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 715 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_BaseParaManager::GetLocalVoxelSize().

7.25.3.34 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	*procGrpNo*	プロセスグループ番号
out	*id*	ランク番号

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

cpm_ParaManager_frtIF.cpp の 1358 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_BaseParaManager::GetMyRankID().

7.25.3.35 CPM_EXTERN void cpm_GetNeighborRankID_ (int * *nID*, int * *procGrpNo*, int * *ierr*)

自ランクの隣接ランク番号を取得

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

引数

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

cpm_ParaManager_frtIF.cpp の 1268 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().

7.25.3.36 CPM_EXTERN void cpm_GetNodeHeadIndex_ (int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの始点頂点の全体空間でのインデクスを取得

- GetNodeHeadIndex 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	*procGrpNo*	プロセスグループ番号
out	*idx*	自ランクの始点頂点インデクス (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 1058 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::GetNodeHeadIndex().

7.25.3.37 CPM_EXTERN void cpm_GetNodeTailIndex_ (int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの終点頂点の全体空間でのインデクスを取得

- GetNodeTailIndex 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	*procGrpNo*	プロセスグループ番号
out	*idx*	自ランクの始点頂点インデクス (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 1102 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::GetNodeTailIndex().

7.25.3.38 CPM_EXTERN void cpm_GetNumRank_ (int * *nrank,* int * *procGrpNo,* int * *ierr*)

ランク数の取得

- GetNumRank のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 1396 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_BaseParaManager::GetNumRank().

7.25.3.39 CPM_EXTERN void cpm_GetPeriodicRankID_ (int * *nID,* int * *procGrpNo,* int * *ierr*)

自ランクの周期境界の隣接ランク番号を取得

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

引数

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

cpm_ParaManager_frtIF.cpp の 1313 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().

7.25.3.40 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	*procGrpNo*	プロセスグループ番号
out	*pch*	ピッチ (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 460 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().

7.25.3.41 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	*procGrpNo*	プロセスグループ番号
out	*idx*	自ランクの始点VOXEL インデクス (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 971 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().

7.25.3.42 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	*procGrpNo*	プロセスグループ番号
out	*idx*	自ランクの終点VOXEL インデクス (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 1014 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().

7.25.3.43 CPM_EXTERN void cpm_Initialize_ (int * *ierr*)

初期化処理 (MPI_Init は実行済みの場合)

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

引数

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

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

参照先 CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, と cpm_ParaManager::get_instance().

7.25.3.44 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

in,out	*buf*	受信バッファ

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

cpm_ParaManager_frtIF.cpp の 1709 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Irecv(), C-PM_SUCCESS, と cpm_ParaManager::get_instance().

7.25.3.45 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

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

cpm_ParaManager_frtIF.cpp の 1670 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Isend(), C-PM_SUCCESS, と cpm_ParaManager::get_instance().

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

cpm_ParaManager_frtIF.cpp の 384 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_BaseParaManager::IsParallel().

7.25.3.47 CPM_EXTERN void cpm_NodeInit_ (int * *div,* int * *nod,* double * *origin,* double * *region,* int * *maxVC,* int * *maxN,* int * *procGrpNo,* int * *ierr*)

領域分割 (FDM 用)

- NodeInit 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	*nod*	空間全体の頂点数 (サイズ 3)
in	*origin*	空間全体の原点 (サイズ 3)
in	*region*	空間全体のサイズ (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	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 260 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, DIV_COMM_SIZE, cpm_ParaManager::get_instance(), と cpm_ParaManager::NodeInit().

7.25.3.48 CPM_EXTERN void cpm_NodeInit_nodiv_ (int * *nod,* double * *origin,* double * *region,* int * *maxVC,* int * *maxN,* int * *divPolicy,* int * *procGrpNo,* int * *ierr*)

領域分割 (FDM)

- NodeInit 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	*nod*	空間全体の頂点数 (サイズ 3)
in	*origin*	空間全体の原点 (サイズ 3)
in	*region*	空間全体のサイズ (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	*divPolicy*	自動分割ポリシー (0:通信面,1:立方体)
in	*procGrpNo*	領域分割を行うプロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 348 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, DIV_COMM_SIZE, DIV_VOX_CUBE, cpm_ParaManager::get_instance(), と cpm_ParaManager::NodeInit().

7.25.3.49 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

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

cpm_ParaManager_frtIF.cpp の 2778 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_PADDING_OFF, cpm_PeriodicCommS4D_(), cpm_ParaManager::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS3D().

7.25.3.50 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

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

cpm_ParaManager_frtIF.cpp の 2715 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_PADDING_OFF, cpm_ParaManager::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS4D().

参照元 cpm_PeriodicCommS3D_(), と cpm_PeriodicCommV3D_().

7.25.3.51 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

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

cpm_ParaManager_frtIF.cpp の 2917 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_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS4DEx().

参照元 cpm_PeriodicCommV3DEx_().

7.25.3.52 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

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

cpm_ParaManager_frtIF.cpp の 2847 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_PADDING_OFF, cpm_PeriodicCommS4D_(), cpm_ParaManager::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3D().

7.25.3.53 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

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

cpm_ParaManager_frtIF.cpp の 2980 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_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3DEx().

7.25.3.54 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

in,out	*buf*	受信バッファ
in	*count*	受信バッファのサイズ (ワード数)
in	*datatype*	データタイプ (fparam.fi を参照)
in	*source*	送信元のランク番号 (procGrpNo 内でのランク番号)
in	*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 1629 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_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::Recv().

7.25.3.55 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

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

cpm_ParaManager_frtIF.cpp の 1589 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_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::Send().

7.25.3.56 CPM_EXTERN void cpm_SetBndCommBuffer_ (int * *maxVC,* int * *maxN,* int * *procGrpNo,* int * *ierr*)

袖通信バッファのセット (Fortran インターフェイス)

- 袖通信バッファ確保処理のFortran インターフェイス関数

引数

in	*maxVC*	送受信バッファの最大袖数
in	*maxN*	送受信バッファの最大成分数
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 1997 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().

7.25.3.57 CPM_EXTERN void cpm_VoxelInit_ (int * *div,* int * *vox,* double * *origin,* double * *region,* int * *maxVC,* int * *maxN,* int * *procGrpNo,* int * *ierr*)

領域分割 (FVM 用)

- VoxelInit 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	*region*	空間全体のサイズ (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	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

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

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, DIV_COMM_SIZE, cpm_ParaManager::get_instance(), と cpm_ParaManager::VoxelInit().

7.25.3.58 CPM_EXTERN void cpm_VoxelInit_nodiv_ (int * *vox,* double * *origin,* double * *region,* int * *maxVC,* int * *maxN,* int * *divPolicy,* int * *procGrpNo,* int * *ierr*)

領域分割 (FVM)

- VoxelInit 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	*region*	空間全体のサイズ (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	*divPolicy*	自動分割ポリシー (0:通信面,1:立方体)
in	*procGrpNo*	領域分割を行うプロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 303 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, DIV_COMM_SIZE, DIV_VOX_CUBE, cpm_ParaManager::get_instance(), と cpm_ParaManager::VoxelInit().

7.25.3.59 CPM_EXTERN void cpm_Wait_ (int ∗ *reqNo,* int ∗ *ierr*)

Wait

- Wait のFortran インターフェイス関数

  引数

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

cpm_ParaManager_frtIF.cpp の 1488 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Wait(), と cpm_ParaManager::get_instance().

7.25.3.60 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

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

cpm_ParaManager_frtIF.cpp の 2354 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().

7.25.3.61 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

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

cpm_ParaManager_frtIF.cpp の 2314 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().

7.25.3.62 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

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

cpm_ParaManager_frtIF.cpp の 2628 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().

7.25.3.63 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

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

cpm_ParaManager_frtIF.cpp の 2399 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().

7.25.3.64 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

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

cpm_ParaManager_frtIF.cpp の 2668 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().

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

cpm_ParaManager_frtIF.cpp の 1517 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Waitall(), と cpm_ParaManager::get_instance().

7.26 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

7.26.1 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

日付

2012/05/31

cpm_ParaManager_MPI.cpp で定義されています。

7.27 cpm_ParaManagerLMR.cpp

```
#include "cpm_ParaManagerLMR.h"
```

cpm_ParaManagerLMR.cpp のインクルード依存関係図

7.27.1 説明

パラレルマネージャクラスのソースファイル

日付

2012/05/31

cpm_ParaManagerLMR.cpp で定義されています。

7.28 cpm_ParaManagerLMR.h

```
#include "cpm_DefineLMR.h"
#include "cpm_BaseParaManager.h"
#include "cpm_VoxelInfoLMR.h"
#include "cpm_LeafCommInfo.h"
#include "inline/cpm_ParaManagerLMR_BndComm.h"
#include "inline/cpm_ParaManagerLMR_BndCommEx.h"
```

cpm_ParaManagerLMR.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_ParaManagerLMR

型定義

- typedef std::map< int, LeafMap > VoxelInfoMapLMR
- typedef std::map< int, cpm_LeafCommInfo * > LeafCommInfoMap
- typedef std::map< int, LeafCommInfoMap > BndCommInfoMap

7.28.1 説明

LMR 用のパラレルマネージャクラスのヘッダーファイル

日付

2015/03/27

cpm_ParaManagerLMR.h で定義されています。

7.28.2 型定義

7.28.2.1 typedef std::map<int, LeafCommInfoMap> BndCommInfoMap

全プロセスグループの袖通信情報マップ

cpm_ParaManagerLMR.h の 34 行で定義されています。

7.28.2.2 typedef std::map<int, cpm_LeafCommInfo*> LeafCommInfoMap

プロセスグループ内の袖通信情報マップ

cpm_ParaManagerLMR.h の 31 行で定義されています。

7.28.2.3 typedef std::map<int, LeafMap> VoxelInfoMapLMR

プロセスグループ毎のVOXEL 空間情報管理マップ

cpm_ParaManagerLMR.h の 28 行で定義されています。

7.29 cpm_ParaManagerLMR_Alloc.cpp

```
#include <stdlib.h>
#include "cpm_ParaManagerLMR.h"
```

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

7.29.1 説明

LMR パラレルマネージャクラスのソースファイル

日付

2012/05/31

cpm_ParaManagerLMR_Alloc.cpp で定義されています。

7.30 cpm_ParaManagerLMR_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)

7.30.1 マクロ定義

7.30.1.1 #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_ParaManagerLMR_BndComm.h の 21 行で定義されています。

参照元 cpm_ParaManagerLMR::packMX(), cpm_ParaManagerLMR::packPX(), cpm_ParaManagerLMR::unpackMX(), と cpm_ParaManagerLMR::unpackPX().

7.30.1.2 #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_ParaManagerLMR_BndComm.h の 28 行で定義されています。

参照元 cpm_ParaManagerLMR::packMY(), cpm_ParaManagerLMR::packPY(), cpm_ParaManagerLMR::unpackMY(), と cpm_ParaManagerLMR::unpackPY().

7.30.1.3 #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_ParaManagerLMR_BndComm.h の 35 行で定義されています。

参照元 cpm_ParaManagerLMR::packMZ(), cpm_ParaManagerLMR::packPZ(), cpm_ParaManagerLMR::unpackMZ(), と cpm_ParaManagerLMR::unpackPZ().

7.31 cpm_ParaManagerLMR_BndCommEx.h

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

マクロ定義

- #define _IDXFX(_N, _I, _J, _K, _NN, _IS, _NJ, _NK, _VC)
- #define _IDXFY(_N, _I, _J, _K, _NN, _NI, _JS, _NK, _VC)
- #define _IDXFZ(_N, _I, _J, _K, _NN, _NI, _NJ, _KS, _VC)

7.31.1 マクロ定義

7.31.1.1 #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_ParaManagerLMR_BndCommEx.h の 21 行で定義されています。

参照元 cpm_ParaManagerLMR::packMXEx(), cpm_ParaManagerLMR::packPXEx(), cpm_ParaManagerLMR::unpackMXEx(), と cpm_ParaManagerLMR::unpackPXEx().

7.31.1.2 #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_ParaManagerLMR_BndCommEx.h の 30 行で定義されています。

参照元 cpm_ParaManagerLMR::packMYEx(), cpm_ParaManagerLMR::packPYEx(), cpm_ParaManagerLMR::unpackMYEx(), と cpm_ParaManagerLMR::unpackPYEx().

7.31.1.3 #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_ParaManagerLMR_BndCommEx.h の 39 行で定義されています。

参照元 cpm_ParaManagerLMR::packMZEx(), cpm_ParaManagerLMR::packPZEx(), cpm_ParaManagerLMR-::unpackMZEx(), と cpm_ParaManagerLMR::unpackPZEx().

7.32 cpm_ParaManagerLMR_frtIF.cpp

```
#include "cpm_ParaManagerLMR.h"
```

cpm_ParaManagerLMR_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_LMR_ cpm_initialize_lmr_
- #define cpm_IsParallel_LMR_ cpm_isparallel_lmr_
- #define cpm_GetNumLeaf_LMR_ cpm_getnumleaf_lmr_
- #define cpm_GetLocalNumLeaf_LMR_ cpm_getlocalnumleaf_lmr_
- #define cpm_GetLeafID_LMR_ cpm_getleafid_lmr_
- #define cpm_GetDivNum_LMR_ cpm_getdivnum_lmr_
- #define cpm_GetPitch_LMR_ cpm_getpitch_lmr_
- #define cpm_GetGlobalVoxelSize_LMR_ cpm_getglobalvoxelsize_lmr_
- #define cpm_GetGlobalNodeSize_LMR_ cpm_getglobalnodesize_lmr_
- #define cpm_GetGlobalArraySize_LMR_ cpm_getglobalarraysize_lmr_
- #define cpm_GetGlobalOrigin_LMR_ cpm_getglobalorigin_lmr_
- #define cpm_GetGlobalRegion_LMR_ cpm_getglobalregion_lmr_
- #define cpm_GetLocalVoxelSize_LMR_ cpm_getlocalvoxelsize_lmr_
- #define cpm_GetLocalNodeSize_LMR_ cpm_getlocalnodesize_lmr_
- #define cpm_GetLocalArraySize_LMR_ cpm_getlocalarraysize_lmr_
- #define cpm_GetLocalOrigin_LMR_ cpm_getlocalorigin_lmr_
- #define cpm_GetLocalRegion_LMR_ cpm_getlocalregion_lmr_
- #define cpm_GetDivPos_LMR_ cpm_getdivpos_lmr_
- #define cpm_GetVoxelHeadIndex_LMR_ cpm_getvoxelheadindex_lmr_
- #define cpm_GetVoxelTailIndex_LMR_ cpm_getvoxeltailindex_lmr_
- #define cpm_GetNodeHeadIndex_LMR_ cpm_getnodeheadindex_lmr_
- #define cpm_GetNodeTailIndex_LMR_ cpm_getnodetailindex_lmr_
- #define cpm_GetArrayHeadIndex_LMR_ cpm_getarrayheadindex_lmr_
- #define cpm_GetArrayTailIndex_LMR_ cpm_getarraytailindex_lmr_
- #define cpm_GetDefPointType_LMR_ cpm_getdefpointtype_lmr_
- #define cpm_GetNeighborRankList_LMR_ cpm_getneighborranklist_lmr_
- #define cpm_GetPeriodicRankList_LMR_ cpm_getperiodicranklist_lmr_
- #define cpm_GetNeighborLeafList_LMR_ cpm_getneighborleaflist_lmr_
- #define cpm_GetPeriodicLeafList_LMR_ cpm_getperiodicleaflist_lmr_
- #define cpm_GetMyRankID_LMR_ cpm_getmyrankid_lmr_
- #define cpm_GetNumRank_LMR_ cpm_getnumrank_lmr_
- #define cpm_Abort_LMR_ cpm_abort_lmr_
- #define cpm_Barrier_LMR_ cpm_barrier_lmr_
- #define cpm_Wait_LMR_ cpm_wait_lmr_
- #define cpm_Waitall_LMR_ cpm_waitall_lmr_
- #define cpm_Bcast_LMR_ cpm_bcast_lmr_
- #define cpm_Send_LMR_ cpm_send_lmr_
- #define cpm_Recv_LMR_ cpm_recv_lmr_
- #define cpm_Isend_LMR_ cpm_isend_lmr_
- #define cpm_Irecv_LMR_ cpm_irecv_lmr_

- #define cpm_Allreduce_LMR_ cpm_allreduce_lmr_
- #define cpm_Gather_LMR_ cpm_gather_lmr_
- #define cpm_Allgather_LMR_ cpm_allgather_lmr_
- #define cpm_Gatherv_LMR_ cpm_gatherv_lmr_
- #define cpm_Allgatherv_LMR_ cpm_allgatherv_lmr_
- #define cpm_BndCommS3D_LMR_ cpm_bndcomms3d_lmr_
- #define cpm_BndCommV3D_LMR_ cpm_bndcommv3d_lmr_
- #define cpm_BndCommS4D_LMR_ cpm_bndcomms4d_lmr_
- #define cpm_BndCommV3DEx_LMR_ cpm_bndcommv3dex_lmr_
- #define cpm_BndCommS4DEx_LMR_ cpm_bndcomms4dex_lmr_
- #define cpm_PeriodicCommS3D_LMR_ cpm_periodiccomms3d_lmr_
- #define cpm_PeriodicCommV3D_LMR_ cpm_periodiccommv3d_lmr_
- #define cpm_PeriodicCommS4D_LMR_ cpm_periodiccomms4d_lmr_
- #define cpm_PeriodicCommV3DEx_LMR_ cpm_periodiccommv3dex_lmr_
- #define cpm_PeriodicCommS4DEx_LMR_ cpm_periodiccomms4dex_lmr_
- #define cpm_GetPeriodicLeafList_LMR_ CPM_GETPERIODICLEAFLIST_LMR

関数

- CPM_EXTERN void cpm_Initialize_LMR_ (int *ierr)
- CPM_EXTERN void cpm_IsParallel_LMR_ (int *ipara, int *ierr)
- CPM_EXTERN void cpm_GetNumLeaf_LMR_ (int *numLeaf, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalNumLeaf_LMR_ (int *numLeaf, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLeafID_LMR_ (int *leafIndex, int *leafID, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetDivNum_LMR_ (int *leafIndex, int *div, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetPitch_LMR_ (int *leafIndex, double *pch, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetGlobalVoxelSize_LMR_ (int *wsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetGlobalNodeSize_LMR_ (int *wsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetGlobalArraySize_LMR_ (int *wsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetGlobalOrigin_LMR_ (double *worg, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetGlobalRegion_LMR_ (double *wrgn, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalVoxelSize_LMR_ (int *lsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalNodeSize_LMR_ (int *lsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalArraySize_LMR_ (int *lsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalOrigin_LMR_ (int *leafIndex, double *lorg, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalRegion_LMR_ (int *leafIndex, double *lrgn, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetDivPos_LMR_ (int *leafIndex, int *pos, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetVoxelHeadIndex_LMR_ (int *leafIndex, int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetVoxelTailIndex_LMR (int *leafIndex, int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNodeHeadIndex_LMR_ (int *leafIndex, int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNodeTailIndex_LMR_ (int *leafIndex, int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetArrayHeadIndex_LMR_ (int *leafIndex, int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetArrayTailIndex_LMR_ (int *leafIndex, int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetDefPointType_LMR_ (int *ideftyp, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetMyRankID_LMR_ (int *id, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNumRank_LMR_ (int *nrank, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNeighborRankList_LMR_ (int *leafIndex, int *face, int *rankList, int *numRank, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetPeriodicRankList_LMR_ (int *leafIndex, int *face, int *rankList, int *numRank, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNeighborLeafList_LMR_ (int *leafIndex, int *face, int *leafList, int *numLeaf, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetPeriodicLeafList_LMR_ (int *leafIndex, int *face, int *leafList, int *numLeaf, int *procGrpNo, int *ierr)

- CPM_EXTERN void cpm_Abort_LMR_ (int *errorcode)
- CPM_EXTERN void cpm_Barrier_LMR_ (int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Wait_LMR_ (int *reqNo, int *ierr)
- CPM_EXTERN void cpm_Waitall_LMR_ (int *count, int *reqlist, int *ierr)
- CPM_EXTERN void cpm_Bcast_LMR_ (void *buf, int *count, int *datatype, int *root, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Send_LMR_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Recv_LMR_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Isend_LMR_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *reqNo, int *ierr)
- CPM_EXTERN void cpm_Irecv_LMR_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *reqNo, int *ierr)
- CPM_EXTERN void cpm_Allreduce_LMR_ (void *sendbuf, void *recvbuf, int *count, int *datatype, int *op, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Gather_LMR_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *root, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Allgather_LMR_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnt, int *recvtype, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Gatherv_LMR_ (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_LMR_ (void *sendbuf, int *sendcnt, int *sendtype, void *recvbuf, int *recvcnts, int *displs, int *recvtype, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_BndCommS4D_LMR_ (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_LMR_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_BndCommV3D_LMR_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_BndCommS4DEx_LMR_ (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_LMR_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *datatype, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_PeriodicCommS4D_LMR_ (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_LMR_ (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_LMR_ (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_LMR_ (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_LMR_ (void *array, int *imax, int *jmax, int *kmax, int *vc, int *vc_comm, int *dir, int *pm, int *datatype, int *procGrpNo, int *ierr)

7.32.1 マクロ定義

7.32.1.1 #define cpm_Abort_LMR_ cpm_abort_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 59 行で定義されています。

7.32.1.2 #define cpm_Allgather_LMR_ cpm_allgather_lmr_

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

7.32.1.3 #define cpm_Allgatherv_LMR_ cpm_allgatherv_lmr_

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

7.32.1.4 #define cpm_Allreduce_LMR_ cpm_allreduce_lmr_

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

7.32.1.5 #define cpm_Barrier_LMR_ cpm_barrier_lmr_

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

7.32.1.6 #define cpm_Bcast_LMR_ cpm_bcast_lmr_

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

7.32.1.7 #define cpm_BndCommS3D_LMR_ cpm_bndcomms3d_lmr_

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

7.32.1.8 #define cpm_BndCommS4D_LMR_ cpm_bndcomms4d_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 75 行で定義されています。
参照元 cpm_BndCommS3D_LMR_(), と cpm_BndCommV3D_LMR_().

7.32.1.9 #define cpm_BndCommS4DEx_LMR_ cpm_bndcomms4dex_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 80 行で定義されています。
参照元 cpm_BndCommV3DEx_LMR_().

7.32.1.10 #define cpm_BndCommV3D_LMR_ cpm_bndcommv3d_lmr_

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

7.32.1.11 #define cpm_BndCommV3DEx_LMR_ cpm_bndcommv3dex_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 79 行で定義されています。

7.32.1.12 #define CPM_EXTERN extern "C"

extern 宣言
cpm_ParaManagerLMR_frtIF.cpp の 20 行で定義されています。

7.32.1.13 #define cpm_Gather_LMR_ cpm_gather_lmr_

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

7.32.1.14 #define cpm_Gatherv_LMR_ cpm_gatherv_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 71 行で定義されています。

7.32.1.15 #define cpm_GetArrayHeadIndex_LMR_ cpm_getarrayheadindex_lmr_

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

7.32.1.16 #define cpm_GetArrayTailIndex_LMR_ cpm_getarraytailindex_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 51 行で定義されています。

7.32.1.17 #define cpm_GetDefPointType_LMR_ cpm_getdefpointtype_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 52 行で定義されています。

7.32.1.18 #define cpm_GetDivNum_LMR_ cpm_getdivnum_lmr_

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

7.32.1.19 #define cpm_GetDivPos_LMR_ cpm_getdivpos_lmr_

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

7.32.1.20 #define cpm_GetGlobalArraySize_LMR_ cpm_getglobalarraysize_lmr_

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

7.32.1.21 #define cpm_GetGlobalNodeSize_LMR_ cpm_getglobalnodesize_lmr_

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

7.32.1.22 #define cpm_GetGlobalOrigin_LMR_ cpm_getglobalorigin_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 38 行で定義されています。

7.32.1.23 #define cpm_GetGlobalRegion_LMR_ cpm_getglobalregion_lmr_

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

7.32.1.24 #define cpm_GetGlobalVoxelSize_LMR_ cpm_getglobalvoxelsize_lmr_

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

7.32.1.25 #define cpm_GetLeafID_LMR_ cpm_getleafid_lmr_

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

7.32.1.26 #define cpm_GetLocalArraySize_LMR_ cpm_getlocalarraysize_lmr_

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

7.32.1.27 #define cpm_GetLocalNodeSize_LMR_ cpm_getlocalnodesize_lmr_

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

7.32.1.28 #define cpm_GetLocalNumLeaf_LMR_ cpm_getlocalnumleaf_lmr_

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

7.32.1.29 #define cpm_GetLocalOrigin_LMR_ cpm_getlocalorigin_lmr_

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

7.32.1.30 #define cpm_GetLocalRegion_LMR_ cpm_getlocalregion_lmr_

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

7.32.1.31 #define cpm_GetLocalVoxelSize_LMR_ cpm_getlocalvoxelsize_lmr_

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

7.32.1.32 #define cpm_GetMyRankID_LMR_ cpm_getmyrankid_lmr_

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

7.32.1.33 #define cpm_GetNeighborLeafList_LMR_ cpm_getneighborleaflist_lmr_

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

7.32.1.34 #define cpm_GetNeighborRankList_LMR_ cpm_getneighborranklist_lmr_

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

7.32.1.35 #define cpm_GetNodeHeadIndex_LMR_ cpm_getnodeheadindex_lmr_

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

7.32.1.36 #define cpm_GetNodeTailIndex_LMR_ cpm_getnodetailindex_lmr_

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

7.32.1.37 #define cpm_GetNumLeaf_LMR_ cpm_getnumleaf_lmr_

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

7.32.1.38 #define cpm_GetNumRank_LMR_ cpm_getnumrank_lmr_

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

7.32.1.39 #define cpm_GetPeriodicLeafList_LMR_ cpm_getperiodicleaflist_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 1378 行で定義されています。

7.32.1.40 #define cpm_GetPeriodicLeafList_LMR_ CPM_GETPERIODICLEAFLIST_LMR

cpm_ParaManagerLMR_frtIF.cpp の 1378 行で定義されています。

7.32.1.41 #define cpm_GetPeriodicRankList_LMR_ cpm_getperiodicranklist_lmr_

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

7.32.1.42 #define cpm_GetPitch_LMR_ cpm_getpitch_lmr_

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

7.32.1.43 #define cpm_GetVoxelHeadIndex_LMR_ cpm_getvoxelheadindex_lmr_

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

7.32.1.44 #define cpm_GetVoxelTailIndex_LMR_ cpm_getvoxeltailindex_lmr_

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

7.32.1.45 #define cpm_Initialize_LMR_ cpm_initialize_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 28 行で定義されています。

7.32.1.46 #define cpm_Irecv_LMR_ cpm_irecv_lmr_

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

7.32.1.47 #define cpm_Isend_LMR_ cpm_isend_lmr_

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

7.32.1.48 #define cpm_IsParallel_LMR_ cpm_isparallel_lmr_

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

7.32.1.49 #define cpm_PeriodicCommS3D_LMR_ cpm_periodiccomms3d_lmr_

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

7.32.1.50 #define cpm_PeriodicCommS4D_LMR_ cpm_periodiccomms4d_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 85 行で定義されています。
参照元 cpm_PeriodicCommS3D_LMR_(), と cpm_PeriodicCommV3D_LMR_().

7.32.1.51 #define cpm_PeriodicCommS4DEx_LMR_ cpm_periodiccomms4dex_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 87 行で定義されています。
参照元 cpm_PeriodicCommV3DEx_LMR_().

7.32.1.52 #define cpm_PeriodicCommV3D_LMR_ cpm_periodiccommv3d_lmr_

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

7.32.1.53 #define cpm_PeriodicCommV3DEx_LMR_ cpm_periodiccommv3dex_lmr_

cpm_ParaManagerLMR_frtIF.cpp の 86 行で定義されています。

7.32.1.54 #define cpm_Recv_LMR_ cpm_recv_lmr_

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

7.32.1.55 #define cpm_Send_LMR_ cpm_send_lmr_

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

7.32.1.56 #define cpm_Wait_LMR_ cpm_wait_lmr_

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

7.32.1.57 #define cpm_Waitall_LMR_ cpm_waitall_lmr_

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

7.32.2 関数

7.32.2.1 CPM_EXTERN void cpm_Abort_LMR_ (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_ParaManagerLMR_frtIF.cpp の 1434 行で定義されています。
参照先 cpm_BaseParaManager::Abort(), と cpm_ParaManagerLMR::get_instance().

7.32.2.2 CPM_EXTERN void cpm_Allgather_LMR_ (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_ParaManagerLMR_frtIF.cpp の 1851 行で定義されています。

参照先 cpm_BaseParaManager::Allgather(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.3 CPM_EXTERN void cpm_Allgatherv_LMR_ (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_ParaManagerLMR_frtIF.cpp の 1953 行で定義されています。

参照先 cpm_BaseParaManager::Allgatherv(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.4 CPM_EXTERN void cpm_Allreduce_LMR_ (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_ParaManagerLMR_frtIF.cpp の 1750 行で定義されています。

参照先 cpm_BaseParaManager::Allreduce(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_MPI_INVALID_OPERATOR, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::GetMPI_Op().

7.32.2.5 CPM_EXTERN void cpm_Barrier_LMR_ (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_ParaManagerLMR_frtIF.cpp の 1462 行で定義されています。

参照先 cpm_BaseParaManager::Barrier(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManagerLMR::get_instance().

7.32.2.6 CPM_EXTERN void cpm_Bcast_LMR_ (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

in,out	*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_ParaManagerLMR_frtIF.cpp の 1551 行で定義されています。

参照先 cpm_BaseParaManager::Bcast(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.7 CPM_EXTERN void cpm_BndCommS3D_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2051 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommS3D(), cpm_BndCommS4D_LMR_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.8 CPM_EXTERN void cpm_BndCommS4D_LMR_ (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,nLeaf) の形式の配列の袖通信を行う
- BndCommS4D のFortran インターフェイス関数

引数

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

cpm_ParaManagerLMR_frtIF.cpp の 2006 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommS4D(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.9 CPM_EXTERN void cpm_BndCommS4DEx_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2152 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommS4DEx(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.10 CPM_EXTERN void cpm_BndCommV3D_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2101 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommV3D(), cpm_BndCommS4D_LMR_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.11 CPM_EXTERN void cpm_BndCommV3DEx_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2197 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommV3DEx(), cpm_BndCommS4DEx_LMR_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.12 CPM_EXTERN void cpm_Gather_LMR_ (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	*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	*root*	受信するランク番号 (procGrpNo 内でのランク番号)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 1801 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_BaseParaManager::Gather(), cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.13 CPM_EXTERN void cpm_Gatherv_LMR_ (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	*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	*root*	受信するランク番号 (procGrpNo 内でのランク番号)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 1902 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_BaseParaManager::Gatherv(), cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMPI_Datatype().

7.32.2.14 CPM_EXTERN void cpm_GetArrayHeadIndex_LMR_ (int * *leafIndex,* int * *idx,* int * *procGrpNo,* int * *ierr*)

指定リーフの始点ボクセルまたは頂点の全体空間でのインデクスを取得

- FVM のときはボクセル、FDM のときは頂点始点インデックスを取得
- GetArrayHeadIndex 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	*leafIndex*	リーフ順番号 (1～)
out	*idx*	始点ボクセルまたは頂点インデクス (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 1049 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetArrayHeadIndex().

7.32.2.15 CPM_EXTERN void cpm_GetArrayTailIndex_LMR_ (int * *leafIndex,* int * *idx,* int * *procGrpNo,* int * *ierr*)

指定リーフの終点ボクセルまたは頂点の全体空間でのインデクスを取得

- FVM のときはボクセル、FDM のときは頂点終点インデックスを取得
- GetArrayTailIndex 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	*leafIndex*	リーフ順番号 (1～)
out	*idx*	終点ボクセルまたは頂点インデクス (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 1095 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetArrayTailIndex().

7.32.2.16 CPM_EXTERN void cpm_GetDefPointType_LMR_ (int * *ideftyp,* int * *procGrpNo,* int * *ierr*)

定義点タイプを取得

- GetDefPointType 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

out	*ideftyp*	定義点タイプ (-1=未定義、0=FVM、1=FDM)
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_ParaManagerLMR_frtIF.cpp の 1139 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetDefPointType().

7.32.2.17 CPM_EXTERN void cpm_GetDivNum_LMR_ (int * *leafIndex,* 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

in	*leafIndex*	リーフ順番号 (1～)
out	*div*	領域分割数 (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 316 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetDivNum().

7.32.2.18 CPM_EXTERN void cpm_GetDivPos_LMR_ (int * *leafIndex,* int * *pos,* int * *procGrpNo,* int * *ierr*)

指定リーフ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	*leafIndex*	リーフ順番号 (1～)
out	*pos*	領域分割位置 (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 826 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetDivPos().

7.32.2.19 CPM_EXTERN void cpm_GetGlobalArraySize_LMR_ (int * *wsz,* int * *procGrpNo,* int * *ierr*)

全体ボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数
- GetGlobalArraySize 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	*wsz*	全体ボクセル数または頂点数 (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_ParaManagerLMR_frtIF.cpp の 486 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALARRAYSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetGlobalArraySize().

7.32.2.20 CPM_EXTERN void cpm_GetGlobalNodeSize_LMR_ (int ∗ *wsz,* int ∗ *procGrpNo,* int ∗ *ierr*)

全体頂点数を取得

- GetGlobalNodeSize 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	*wsz*	全体頂点数 (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_ParaManagerLMR_frtIF.cpp の 443 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALNODESIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetGlobalNodeSize().

7.32.2.21 CPM_EXTERN void cpm_GetGlobalOrigin_LMR_ (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_ParaManagerLMR_frtIF.cpp の 528 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_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetGlobalOrigin().

7.32.2.22 CPM_EXTERN void cpm_GetGlobalRegion_LMR_ (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_ParaManagerLMR_frtIF.cpp の 570 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_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetGlobalRegion().

7.32.2.23 CPM_EXTERN void cpm_GetGlobalVoxelSize_LMR_ (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	*procGrpNo*	プロセスグループ番号
out	*wsz*	全体ボクセル数 (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_ParaManagerLMR_frtIF.cpp の 401 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_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetGlobalVoxelSize().

7.32.2.24 CPM_EXTERN void cpm_GetLeafID_LMR_ (int * *leafIndex*, int * *leafID*, int * *procGrpNo*, int * *ierr*)

指定リーフのリーフID を取得する

- GetLeafID のFortran インターフェイス関数

引数

in	*leafIndex*	リーフ順番号 (1～)
out	*leafID*	リーフID
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_ParaManagerLMR_frtIF.cpp の 282 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetLeafID().

7.32.2.25 CPM_EXTERN void cpm_GetLocalArraySize_LMR_ (int * *lsz*, int * *procGrpNo*, int * *ierr*)

1 リーフのボクセル数または頂点数を取得

- FVM のときはボクセル数、FDM のときは頂点数
- GetLocalArraySize のFortran インターフェイス関数

引数

out	*lsz*	頂点数 (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号

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

cpm_ParaManagerLMR_frtIF.cpp の 697 行で定義されています。

参照先 CPM_ERROR_GET_LOCALARRAYSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetLocalArraySize().

7.32.2.26 CPM_EXTERN void cpm_GetLocalNodeSize_LMR_ (int * *lsz*, int * *procGrpNo*, int * *ierr*)

1 リーフの頂点数を取得

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

  引数

out	*lsz*	頂点数 (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 654 行で定義されています。

参照先 CPM_ERROR_GET_LOCALNODESIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetLocalNodeSize().

7.32.2.27 CPM_EXTERN void cpm_GetLocalNumLeaf_LMR_ (int * *numLeaf*, int * *procGrpNo*, int * *ierr*)

自ランクが担当するリーフ数を取得する

引数

out	*numLeaf*	リーフ数
in	*procGrpNo*	プロセスグループ番号 (省略時=0)
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

戻り値

自ランクが担当するリーフ数

cpm_ParaManagerLMR_frtIF.cpp の 248 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetLocalNumLeaf().

7.32.2.28 CPM_EXTERN void cpm_GetLocalOrigin_LMR_ (int * *leafIndex*, double * *lorg*, int * *procGrpNo*, int * *ierr*)

指定リーフの空間原点を取得

- GetLocalOrigin のFortran インターフェイス関数

  引数

in	*leafIndex*	リーフ順番号 (1~)
out	*lorg*	空間原点 (3word の実数配列)
in	*procGrpNo*	プロセスグループ番号

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

cpm_ParaManagerLMR_frtIF.cpp の 740 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetLocalOrigin().

7.32.2.29 CPM_EXTERN void cpm_GetLocalRegion_LMR_ (int * *leafIndex,* double * *lrgn,* int * *procGrpNo,* int * *ierr*)

指定リーフ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	*leafIndex*	リーフ順番号 (1～)
out	*lrgn*	空間サイズ (3word の実数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 783 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetLocalRegion().

7.32.2.30 CPM_EXTERN void cpm_GetLocalVoxelSize_LMR_ (int * *lsz,* int * *procGrpNo,* int * *ierr*)

１リーフのボクセル数を取得

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

  引数

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

cpm_ParaManagerLMR_frtIF.cpp の 612 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_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetLocalVoxelSize().

7.32.2.31 CPM_EXTERN void cpm_GetMyRankID_LMR_ (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	*procGrpNo*	プロセスグループ番号
out	*id*	ランク番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 1170 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_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetMyRankID().

7.32.2.32 CPM_EXTERN void cpm_GetNeighborLeafList_LMR_ (int * *leafIndex,* int * *face,* int * *leafList,* int * *numLeaf,* int * *procGrpNo,* int * *ierr*)

指定リーフの指定面における隣接リーフ番号を取得

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

引数

in	*leafIndex*	リーフ順番号 (1～)
in	*face*	面方向
out	*leafList*	指定リーフの指定面における隣接リーフ番号配列
out	*numLeaf*	面の数 (0 or 1 or 4)
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_ParaManagerLMR_frtIF.cpp の 1343 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetNeighborLeafList().

7.32.2.33 CPM_EXTERN void cpm_GetNeighborRankList_LMR_ (int * *leafIndex,* int * *face,* int * *rankList,* int * *numRank,* int * *procGrpNo,* int * *ierr*)

指定リーフの指定面における自リーフの隣接ランク番号を取得

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

引数

in	*leafIndex*	リーフ順番号 (1～)
in	*face*	面方向
out	*rankList*	指定リーフの指定面における自リーフの隣接ランク番号整数配列
out	*numRank*	面の数 (0 or 1 or 4)
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_ParaManagerLMR_frtIF.cpp の 1249 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetNeighborRankList().

7.32.2.34 CPM_EXTERN void cpm_GetNodeHeadIndex_LMR_ (int * *leafIndex,* int * *idx,* int * *procGrpNo,* int * *ierr*)

指定リーフの始点頂点の全体空間でのインデクスを取得

- GetNodeHeadIndex 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	*leafIndex*	リーフ順番号 (1～)
out	*idx*	始点頂点インデクス (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号

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

cpm_ParaManagerLMR_frtIF.cpp の 958 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetNodeHeadIndex().

7.32.2.35 CPM_EXTERN void cpm_GetNodeTailIndex_LMR_ (int * *leafIndex,* int * *idx,* int * *procGrpNo,* int * *ierr*)

指定リーフの終点頂点の全体空間でのインデクスを取得

- GetNodeTailIndex 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	*leafIndex*	リーフ順番号 (1～)
out	*idx*	始点頂点インデクス (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 1003 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetNodeTailIndex().

7.32.2.36 CPM_EXTERN void cpm_GetNumLeaf_LMR_ (int * *numLeaf,* int * *procGrpNo,* int * *ierr*)

全リーフ数を取得する

引数

out	*numLeaf*	全リーフ数
in	*procGrpNo*	プロセスグループ番号 (省略時=0)
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

戻り値

全リーフ数

cpm_ParaManagerLMR_frtIF.cpp の 215 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetNumLeaf().

7.32.2.37 CPM_EXTERN void cpm_GetNumRank_LMR_ (int * *nrank,* int * *procGrpNo,* int * *ierr*)

ランク数の取得

- GetNumRank のFortran インターフェイス関数

引数

in	*procGrpNo*	プロセスグループ番号

out	*nrank*	ランク数
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 1208 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_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::GetNumRank().

7.32.2.38 CPM_EXTERN void cpm_GetPeriodicLeafList_LMR_ (int * *leafIndex,* int * *face,* int * *leafList,* int * *numLeaf,* int * *procGrpNo,* int * *ierr*)

指定リーフの指定面における周期境界の隣接リーフ番号を取得

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

引数

in	*leafIndex*	リーフ順番号 (1～)
in	*face*	面方向
out	*leafList*	指定リーフの指定面における周期境界の隣接リーフ番号配列
out	*numLeaf*	面の数 (0 or 1 or 4)
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_ParaManagerLMR_frtIF.cpp の 1392 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetPeriodicLeafList().

7.32.2.39 CPM_EXTERN void cpm_GetPeriodicRankList_LMR_ (int * *leafIndex,* int * *face,* int * *rankList,* int * *numRank,* int * *procGrpNo,* int * *ierr*)

指定リーフの指定面における自リーフの周期境界の隣接ランク番号を取得

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

引数

in	*leafIndex*	リーフ順番号 (1～)
in	*face*	面方向
out	*rankList*	指定リーフの指定面における自リーフの周期境界の隣接ランク番号整数配列
out	*numRank*	面の数 (0 or 1 or 4)
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_ParaManagerLMR_frtIF.cpp の 1296 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetPeriodicRankList().

7.32.2.40 CPM_EXTERN void cpm_GetPitch_LMR_ (int * *leafIndex,* 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	*leafIndex*	リーフ順番号 (1～)
in	*procGrpNo*	プロセスグループ番号
out	*pch*	ピッチ (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_ParaManagerLMR_frtIF.cpp の 359 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetPitch().

7.32.2.41 CPM_EXTERN void cpm_GetVoxelHeadIndex_LMR_ (int ∗ *leafIndex,* int ∗ *idx,* int ∗ *procGrpNo,* int ∗ *ierr*)

指定リーフの始点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	*leafIndex*	リーフ順番号 (1～)
out	*idx*	始点VOXEL インデクス (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 870 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetVoxelHeadIndex().

7.32.2.42 CPM_EXTERN void cpm_GetVoxelTailIndex_LMR (int ∗ *leafIndex,* int ∗ *idx,* int ∗ *procGrpNo,* int ∗ *ierr*)

指定リーフの終点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	*leafIndex*	リーフ順番号 (1～)
out	*idx*	終点VOXEL インデクス (3word の整数配列)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 914 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_ParaManagerLMR::get_instance(), と cpm_ParaManagerLMR::GetVoxelTailIndex().

7.32.2.43 CPM_EXTERN void cpm_Initialize_LMR_ (int ∗ *ierr*)

初期化処理 (MPI_Init は実行済みの場合)

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

引数

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

cpm_ParaManagerLMR_frtIF.cpp の 159 行で定義されています。

参照先 CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, と cpm_ParaManagerLMR::get_instance().

7.32.2.44 CPM_EXTERN void cpm_Irecv_LMR_ (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

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

cpm_ParaManagerLMR_frtIF.cpp の 1711 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Irecv(), CPM_SUCCESS, と cpm_ParaManagerLMR::get_instance().

7.32.2.45 CPM_EXTERN void cpm_Isend_LMR_ (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

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

cpm_ParaManagerLMR_frtIF.cpp の 1672 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Isend(), CPM_SUCCESS, と cpm_ParaManagerLMR::get_instance().

7.32.2.46 CPM_EXTERN void cpm_IsParallel_LMR_ (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	*ipara*	並列実行フラグ (1=並列実行、1 以外=逐次実行)
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManagerLMR_frtIF.cpp の 181 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManagerLMR::get_instance(), と cpm_BaseParaManager::IsParallel().

7.32.2.47 CPM_EXTERN void cpm_PeriodicCommS3D_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2313 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_LMR_, cpm_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManagerLMR::PeriodicCommS3D().

7.32.2.48 CPM_EXTERN void cpm_PeriodicCommS4D_LMR_ (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,nLeaf) 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

in,out	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)

in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*dir*	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	*pm*	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi を参照)
in	*procGrpNo*	プロセスグループ番号
out	*ierr*	終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManagerLMR_frtIF.cpp の 2250 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_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManagerLMR::PeriodicCommS4D().

7.32.2.49 CPM_EXTERN void cpm_PeriodicCommS4DEx_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2452 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_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManagerLMR::PeriodicCommS4DEx().

7.32.2.50 CPM_EXTERN void cpm_PeriodicCommV3D_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2382 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_LMR_, cpm_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManagerLMR::PeriodicCommV3D().

7.32.2.51 CPM_EXTERN void cpm_PeriodicCommV3DEx_LMR_ (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,nLeaf) 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

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

cpm_ParaManagerLMR_frtIF.cpp の 2515 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_LMR_, cpm_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_ParaManagerLMR::PeriodicCommV3DEx().

7.32.2.52 CPM_EXTERN void cpm_Recv_LMR_ (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

in,out	*buf*	受信バッファ
in	*count*	受信バッファのサイズ (ワード数)
in	*datatype*	データタイプ (fparam.fi を参照)
in	*source*	送信元のランク番号 (procGrpNo 内でのランク番号)
in	*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_ParaManagerLMR_frtIF.cpp の 1631 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_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::Recv().

7.32.2.53 CPM_EXTERN void cpm_Send_LMR_ (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

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

cpm_ParaManagerLMR_frtIF.cpp の 1591 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_ParaManagerLMR::get_instance(), cpm_BaseParaManager::GetMPI_Datatype(), と cpm_BaseParaManager::Send().

7.32.2.54 CPM_EXTERN void cpm_Wait_LMR_ (int * *reqNo*, int * *ierr*)

Wait

- Wait のFortran インターフェイス関数

引数

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

cpm_ParaManagerLMR_frtIF.cpp の 1490 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Wait(), と cpm_ParaManagerLMR::get_instance().

7.32.2.55 CPM_EXTERN void cpm_Waitall_LMR_ (int * *count*, int * *reqlist*, int * *ierr*)

Waitall

- Waitall のFortran インターフェイス関数

引数

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

cpm_ParaManagerLMR_frtIF.cpp の 1519 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_BaseParaManager::cpm_Waitall(), と cpm_ParaManagerLMR::get_instance().

7.33 cpm_ParaManagerLMR_MPI.cpp

```
#include "stdlib.h"
#include "cpm_ParaManagerLMR.h"
#include <unistd.h>
```

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

7.34 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

- CES
- 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)

7.34.1 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

日付

2013/04/02

cpm_PathUtil.h で定義されています。

7.35 cpm_TextParser.cpp

```
#include "cpm_TextParser.h"
```
cpm_TextParser.cpp のインクルード依存関係図

7.35.1 説明

TextParser クラスのソースファイル

日付

2012/05/31

cpm_TextParser.cpp で定義されています。

7.36 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

7.36.1 説明

テキストパーサークラスのヘッダーファイル

日付

2012/05/31

cpm_TextParser.h で定義されています。

7.37 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

7.37.1 説明

CPM 領域情報のTextParser クラスのソースファイル

日付

2012/05/31

LMR 用領域情報のTextParser クラスのソースファイル

日付

2015/03/27

cpm_TextParserDomain.cpp で定義されています。

7.38 cpm_TextParserDomain.h

```
#include "cpm_TextParser.h"
#include "cpm_DomainInfo.h"
#include <string.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

7.38.1 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

日付

2012/05/31

cpm_TextParserDomain.h で定義されています。

7.39 cpm_TextParserDomainLMR.cpp

```
#include "cpm_TextParserDomainLMR.h"
#include "cpm_PathUtil.h"
```

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

7.40 cpm_TextParserDomainLMR.h

```
#include "cpm_TextParser.h"
#include <string.h>
```

cpm_TextParserDomainLMR.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_OCT_DOMAIN_INFO
- class cpm_TextParserDomainLMR

7.40.1 説明

LMR 用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

日付

2012/05/31

cpm_TextParserDomainLMR.h で定義されています。

7.41 cpm_Version.h

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

マクロ定義

- #define CPM_VERSION_NO "2.1.4"
- #define CPM_REVISION "20160428_0900"

7.41.1 説明

CPM バージョン情報のヘッダーファイル

cpm_Version.h で定義されています。

7.41.2 マクロ定義

7.41.2.1 #define CPM_REVISION "20160428_0900"

CPM ライブラリのリビジョン

cpm_Version.h の 24 行で定義されています。

参照元 cpm_Base::getRevisionInfo().

7.41.2.2 #define CPM_VERSION_NO "2.1.4"

CPM ライブラリのバージョン

cpm_Version.h の 21 行で定義されています。

参照元 cpm_Base::getVersionInfo().

7.42 cpm_VoxelInfo.cpp

```
#include "cpm_VoxelInfo.h"
```

cpm_VoxelInfo.cpp のインクルード依存関係図

7.42.1 説明

VOXEL 空間情報クラスのソースファイル

日付

2012/05/31

cpm_VoxelInfo.cpp で定義されています。

7.43 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

7.43.1 説明

VOXEL 空間情報クラスのヘッダーファイル

日付

2012/05/31

cpm_VoxelInfo.h で定義されています。

7.44 cpm_VoxelInfoCART.cpp

```
#include "cpm_VoxelInfoCART.h"
```
cpm_VoxelInfoCART.cpp のインクルード依存関係図

7.44.1 説明

カーテシアン用のVOXEL 空間情報クラスのソースファイル

日付

2015/03/27

cpm_VoxelInfoCART.cpp で定義されています。

7.45 cpm_VoxelInfoCART.h

```
#include "cpm_VoxelInfo.h"
```
cpm_VoxelInfoCART.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_VoxelInfoCART

7.45.1 説明

カーテシアン用のVOXEL 空間情報クラスのヘッダーファイル o

日付

2015/03/27

cpm_VoxelInfoCART.h で定義されています。

7.46 cpm_VoxelInfoLMR.cpp

```
#include "cpm_VoxelInfoLMR.h"
```
cpm_VoxelInfoLMR.cpp のインクルード依存関係図

7.46.1 説明

VOXEL 空間情報クラスのソースファイル

日付

2012/05/31

cpm_VoxelInfoLMR.cpp で定義されています。

7.47 cpm_VoxelInfoLMR.h

```
#include "cpm_VoxelInfo.h"
#include "BCMOctree.h"
#include "BCMFileCommon.h"
#include "cpm_TextParserDomainLMR.h"
#include "cpm_DefineLMR.h"
```
cpm_VoxelInfoLMR.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_VoxelInfoLMR

型定義

- typedef std::map< int, cpm_VoxelInfoLMR * > LeafMap

7.47.1 説明

LMR 用のVOXEL 空間情報クラスのヘッダーファイル

日付

2015/03/27

cpm_VoxelInfoLMR.h で定義されています。

7.47.2 型定義

7.47.2.1 typedef std::map<int, cpm_VoxelInfoLMR*> LeafMap

cpm_VoxelInfoLMR.h の 30 行で定義されています。

7.48 Divider.h

ブロック分割判定クラス (基底クラス)

```
#include "RootGrid.h"
#include "Pedigree.h"
```

Divider.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 Divider

  ブロック分割判定クラス (基底クラス).

7.48.1 説明

ブロック分割判定クラス (基底クラス)

Divider.h で定義されています。

7.49 NeighborInfo.h

隣接情報クラス

```
#include "BCMTools.h"
#include "mpi.h"
#include <iostream>
```

NeighborInfo.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 NeighborInfo

  隣接情報クラス.

7.49.1 説明

隣接情報クラス

NeighborInfo.h で定義されています。

7.50 Node.h

Octree 用ノードクラス

```
#include "Pedigree.h"
#include "Vec3.h"
```
Node.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 Node

  Octree **ノードクラス**.

7.50.1 説明

Octree 用ノードクラス

Node.h で定義されています。

7.51 Partition.h

1 次元ブロック領域分割用ユーティリティクラス

```
#include <vector>
#include <algorithm>
#include <iostream>
#include <cassert>
#include "mpi.h"
```
Partition.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 Partition

  1 **次元ブロック領域分割用ユーティリティクラス**.

7.51.1 説明

1 次元ブロック領域分割用ユーティリティクラス

Partition.h で定義されています。

7.52 Pedigree.h

Octree 用Pedigree クラス

```
#include <stdint.h>
#include <iostream>
#include "BCMTools.h"
```
Pedigree.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 Pedigree

関数

- std::ostream & operator<< (std::ostream &os, const Pedigree &p)
  Pedigree **情報のストリームへの出力**.

7.52.1 説明

Octree 用Pedigree クラス

Pedigree.h で定義されています。

7.52.2 関数

7.52.2.1 std::ostream& operator<< (std::ostream & *os,* const Pedigree & *p*) [inline]

Pedigree 情報のストリームへの出力.

Pedigree.h の 219 行で定義されています。

参照先 Pedigree::getLevel(), Pedigree::getRootID(), Pedigree::getX(), Pedigree::getY(), と Pedigree::getZ().

7.53 RootGrid.h

マルチルートOctree 用のルートブロック配置管理クラス

```
#include "BCMTools.h"
#include "Vec3.h"
#include "mpi.h"
```

RootGrid.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 RootGrid

7.53.1 説明

マルチルートOctree 用のルートブロック配置管理クラス

RootGrid.h で定義されています。

7.54 Vec3.h

version 1.1 2014-04-23

```
#include <iostream>
#include <math.h>
```

Vec3.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 Vec3class::Vec3< T >

ネームスペース

- Vec3class

マクロ定義

- #define REAL_TYPE float

型定義

- typedef Vec3< unsigned char > Vec3class::Vec3uc
- typedef Vec3< int > Vec3class::Vec3i
- typedef Vec3< float > Vec3class::Vec3f
- typedef Vec3< double > Vec3class::Vec3d
- typedef Vec3< REAL_TYPE > Vec3class::Vec3r

列挙型

- enum Vec3class::AxisEnum { Vec3class::AXIS_X = 0, Vec3class::AXIS_Y, Vec3class::AXIS_Z, Vec3class::AXIS_ERROR }

関数

- template<typename T >
  Vec3< T > Vec3class::operator* (T s, const Vec3< T > &v)
- template<typename T >
  Vec3< T > Vec3class::multi (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T Vec3class::dot (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  Vec3< T > Vec3class::cross (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T Vec3class::distanceSquared (const Vec3< T > &a, const Vec3< T > &b)
- template<typename T >
  T Vec3class::distance (const Vec3< T > &a, const Vec3< T > &b)
- bool Vec3class::lessVec3f (const Vec3f &a, const Vec3f &b)
- template<typename T >
  std::istream & Vec3class::operator>> (std::istream &is, Vec3< T > &v)
- template<typename T >
  std::ostream & Vec3class::operator<< (std::ostream &os, const Vec3< T > &v)
- std::istream & Vec3class::operator>> (std::istream &is, Vec3uc &v)
- std::ostream & Vec3class::operator<< (std::ostream &os, const Vec3uc &v)

7.54.1 説明

version 1.1 2014-04-23

Vec3.h で定義されています。

7.54.2 マクロ定義

7.54.2.1 #define REAL_TYPE float

実数型の指定

- デフォルトでは、REAL_TYPE=float
- コンパイル時オプション-D_REAL_IS_DOUBLE_を付与することで REAL_TYPE=double になる

Vec3.h の 47 行で定義されています。

Index