Cartesian Partition Manager Library

2.0.0

作成： Doxygen 1.8.5

Fri Jun 5 2015 15:35:57

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 66 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 26 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 106 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 114 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 31 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 42 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 56 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 91 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 104 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 77 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 173 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 215 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 197 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 165 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 188 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 205 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 226 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 211 行で定義されています。

5.5.1.2 typedef Vec3<float> Vec3class::Vec3f

Vec3.h の 210 行で定義されています。

5.5.1.3 typedef Vec3<int> Vec3class::Vec3i

Vec3.h の 209 行で定義されています。

5.5.1.4 typedef Vec3<REAL_TYPE> Vec3class::Vec3r

Vec3.h の 212 行で定義されています。

5.5.1.5 typedef Vec3<unsigned char> Vec3class::Vec3uc

Vec3.h の 208 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 51 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 234 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 246 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 241 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 229 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 251 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 224 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 219 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 266 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 281 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 260 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 273 行で定義されています。

Chapter 6

クラス

6.1 クラス BCMOctree

BCM 用マルチルートOCtree クラス.

`#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)
  コンストラクタ.
- BCMOctree (RootGrid *rootGrid, const std::vector< Pedigree > &pedigrees)
  コンストラクタ *(ファイルロード用)*
- ~BCMOctree ()
  デストラクタ.
- void broadcast (MPI::Intracomm &comm=MPI::COMM_WORLD)
  Octree **情報を他** *rank* **にブロードキャスト**.
- const RootGrid * getRootGrid () const
  ルートグリッドを取得
- int getNumLeafNode () const
  リーフノード総数を取得.
- std::vector< Node * > & getLeafNodeArray ()
  リーフノードリストを取得.
- const std::vector< Node * > & getLeafNodeArray () const
  リーフノードリストを取得.
- bool 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**.
- Vec3d getOrigin (const Node *node) const
  指定されたノードの原点位置を取得.
- NeighborInfo * makeNeighborInfo (const Node *node, const Partition *partition) const
  指定されたノードの隣接情報を計算.

Static Public メソッド

- static BCMOctree * ReceiveFromMaster (MPI::Intracomm &comm=MPI::COMM_WORLD)
  rank0 **から***Octree* **情報を受信**.

Private メソッド

- BCMOctree (RootGrid *rootGrid, Ordering ordering, int numLeafNode, const unsigned char *buf)
  コンストラクタ *(***リーフノードの***Pedigree* **リストから***)*.
- void buildTreeFromPedigreeList (int numLeafNode, const unsigned char *buf)
  リーフノードの*Pedigree* **リストから***Octree* **を再構築**.
- void pickupLeafNodeZOrdering (Node *node)
  Z **オーダリング順にリーフノードリストを作成**.
- void pickupLeafNodeHilbertOrdering (Node *node, int orientation)
  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.
- 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)
  再帰呼び出しによりツリー生成.
- void deleteNode (Node *node)
  再帰呼び出しによるツリー消去.
- Node * findNeighborNode (const Node *node, Face face) const
  隣接ノード探索.
- void packPedigrees (Node *node, size_t &ip, unsigned char *buf)
  Pedigree **情報を通信用バッファにパック**.

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 33 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 38 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.

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 (ファイルロード用)

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).

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(), pickupLeafNodeZOrdering(), 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.

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.

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.

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.

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.

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.

BCMOctree.h の 109 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.

BCMOctree.h の 112 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.

BCMOctree.h の 106 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.

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

BCMOctree.h の 103 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.

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.

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.

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.

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.

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.

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 49 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 57 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 58 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 55 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 51 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 47 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 53 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 C_PARAMANAGER

C_PARAMANAGER のコラボレーション図

Private メソッド

- C_PARAMANAGER ()
- ~C_PARAMANAGER ()

Static Private メソッド

- static C_PARAMANAGER * get_instance ()

Private 変数

- cpm_ParaManager * pParaManager

フレンド

- class cpm_ParaManager

6.3.1 説明

並列管理クラスの自動破棄管理クラス

cpm_ParaManager.cpp の 24 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 C_PARAMANAGER::C_PARAMANAGER ( ) [inline], [private]

cpm_ParaManager.cpp の 29 行で定義されています。

参照先 pParaManager.

6.3.2.2 C_PARAMANAGER::~C_PARAMANAGER ( ) [inline], [private]

cpm_ParaManager.cpp の 33 行で定義されています。

参照先 pParaManager.

6.3.3 関数

6.3.3.1 static C_PARAMANAGER* C_PARAMANAGER::get_instance ( ) [inline], [static], [private]

cpm_ParaManager.cpp の 38 行で定義されています。
参照元 cpm_ParaManager::get_instance().

6.3.4 フレンドと関連する関数

6.3.4.1 friend class cpm_ParaManager [friend]

cpm_ParaManager.cpp の 26 行で定義されています。

6.3.5 変数

6.3.5.1 cpm_ParaManager* C_PARAMANAGER::pParaManager [private]

cpm_ParaManager.cpp の 28 行で定義されています。
参照元 C_PARAMANAGER(), cpm_ParaManager::get_instance(), と ~C_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.cpp

6.4 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

6.4.1 説明

CPM のサブ領域情報クラス

cpm_DomainInfo.h の 107 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_ActiveSubdomainInfo::cpm_ActiveSubdomainInfo ( )

デフォルトコンストラクタ

cpm_DomainInfo.cpp の 138 行で定義されています。

参照先 clear().

6.4.2.2 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 146 行で定義されています。

参照先 SetPos().

6.4.2.3 cpm_ActiveSubdomainInfo::~cpm_ActiveSubdomainInfo ( ) [virtual]

デストラクタ

cpm_DomainInfo.cpp の 154 行で定義されています。

6.4.3 関数

6.4.3.1 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 161 行で定義されています。

参照先 m_pos.

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

6.4.3.2 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 181 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().

6.4.3.3 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 200 行で定義されています。

参照先 m_pos.

6.4.3.4 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 189 行で定義されています。

参照先 m_pos.

6.4.3.5 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 171 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().

6.4.4 変数

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

領域分割内での位置

cpm_DomainInfo.h の 162 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

6.5 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 ()

6.5.1 説明

CPM のベースクラス

cpm_Base.h の 49 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_Base::cpm_Base ( ) [inline],[protected]

コンストラクタ

cpm_Base.h の 242 行で定義されています。

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

デストラクタ

cpm_Base.h の 245 行で定義されています。

6.5.3 関数

6.5.3.1 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 191 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().

6.5.3.2 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 213 行で定義されています。

参照先 cpm_strCompare().

6.5.3.3 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_ParaManager::GetMPI_Comm().

6.5.3.4 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 152 行で定義されています。

6.5.3.5 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 58 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_ParaManager::GetMyRankID(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerCART::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), と cpm_ParaManagerCART::PeriodicCommS4DEx().

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

リビジョンを情報を返す

cpm_Base.h の 234 行で定義されています。

参照先 CPM_REVISION.

6.5.3.7 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().

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

時刻の取得 (gettimeofday 版)

時刻

cpm_Base.h の 108 行で定義されています。

参照元 GetSpanTime().

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

バージョンを情報を返す

cpm_Base.h の 226 行で定義されています。

参照先 CPM_VERSION_NO.

6.5.3.10 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 142 行で定義されています。

参照先 GetWTime().

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

時刻の取得 (MPI_Wtime 版)

時刻

cpm_Base.h の 132 行で定義されています。

参照元 GetWSpanTime().

6.5.3.12 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_ParaManager::Allgather(), cpm_ParaManager::Allgatherv(), cpm_ParaManager::Allreduce(), cpm_ParaManager::Barrier(), cpm_ParaManager::Bcast(), cpm_ParaManager::CreateProcessGroup(), cpm_ParaManager::Gather(), cpm_ParaManager::Gatherv(), cpm_ParaManager::GetMyRankID(), cpm_ParaManager::GetNumRank(), cpm_VoxelInfoCART::Init(), cpm_VoxelInfoLMR::Init(), cpm_ParaManager::Irecv(), cpm_ParaManager::Isend(), cpm_ParaManager::Recv(), cpm_ParaManager::Send(), cpm_ParaManagerCART::VoxelInit(), と cpm_ParaManagerLMR::VoxelInit_LMR().

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

NULL のランクかどうかを確認

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_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerLMR::BndCommS4DEx_nowait(), cpm_VoxelInfoLMR::GetNeighborRankList(), cpm_VoxelInfoLMR::GetPeriodicRankList(), cpm_VoxelInfo::IsInnerBoundary(), cpm_VoxelInfo::IsOuterBoundary(), 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_ParaManagerCART::packX(), cpm_ParaManagerCART::packXEx(), cpm_ParaManagerCART::packY(), cpm_ParaManagerCART::packYEx(), cpm_ParaManagerCART::packZ(), cpm_ParaManagerCART::packZEx(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), cpm_ParaManagerLMR::recv_LMR(), cpm_ParaManagerCART::sendrecv(), cpm_VoxelInfoLMR::SetNeighborInfo(), 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(), cpm_ParaManagerLMR::unpackPZEx(), cpm_ParaManagerCART::unpackX(), cpm_ParaManagerCART::unpackXEx(), cpm_ParaManagerCART::unpackY(), cpm_ParaManagerCART::unpackYEx(), cpm_ParaManagerCART::unpackZ(), cpm_ParaManagerCART::unpackZEx(), cpm_ParaManagerLMR::wait_BndCommS4D(), と cpm_ParaManagerLMR::wait_BndCommS4DEx().

6.5.3.14 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_ParaManager::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

6.6 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
- 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**

Additional Inherited Members

6.6.1 説明

CPM の領域情報クラス

cpm_DomainInfo.h の 27 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 22 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 30 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 122 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 37 行で定義されています。

参照先 m_origin, m_pitch, m_region, と m_voxNum.

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

6.6.3.3 const double * cpm_DomainInfo::GetOrigin ( ) const

原点の取得

戻り値

原点情報実数配列のポインタ

cpm_DomainInfo.cpp の 61 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().

6.6.3.4 const double * cpm_DomainInfo::GetPitch ( ) const

ピッチの取得

戻り値

ピッチ情報実数配列のポインタ

cpm_DomainInfo.cpp の 79 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().

6.6.3.5 const double * cpm_DomainInfo::GetRegion ( ) const

空間サイズの取得

戻り値

空間サイズ情報実数配列のポインタ

cpm_DomainInfo.cpp の 97 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().

6.6.3.6 const int * cpm_DomainInfo::GetVoxNum ( ) const

VOXEL 数の取得

戻り値

VOXEL 数情報実数配列のポインタ

cpm_DomainInfo.cpp の 115 行で定義されています。

参照先 m_voxNum.

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

6.6.3.7 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 51 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_ParaManagerCART::VoxelInit(), と cpm_ParaManagerCART::VoxelInit_Subdomain().

6.6.3.8 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 69 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_ParaManagerCART::VoxelInit(), と cpm_ParaManagerCART::VoxelInit_Subdomain().

6.6.3.9 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 87 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_ParaManagerCART::VoxelInit(), と cpm_ParaManagerCART::VoxelInit_Subdomain().

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

VOXEL 数のセット

引数

in	*vox*	VOXEL 数情報

cpm_DomainInfo.cpp の 105 行で定義されています。

参照先 m_voxNum.

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

6.6.4 変数

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

原点

cpm_DomainInfo.h の 98 行で定義されています。

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

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

ピッチ

cpm_DomainInfo.h の 100 行で定義されています。

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

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

空間サイズ

cpm_DomainInfo.h の 99 行で定義されています。

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

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

VOXEL 数

cpm_DomainInfo.h の 101 行で定義されています。

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

このクラスの説明は次のファイルから生成されました:

- cpm_DomainInfo.h
- cpm_DomainInfo.cpp

6.7 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 168 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 210 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 218 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 269 行で定義されています。

参照先 IsExistSubdomain(), と m_subDomainInfo.

参照元 CheckData(), と cpm_ParaManagerCART::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 314 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_ParaManagerCART::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 225 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 248 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

cpm_DomainInfo.cpp の 298 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 306 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 282 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_ParaManagerCART::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 256 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 361 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 388 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_ParaManagerCART::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 416 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 238 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_ParaManagerCART::VoxelInit(), と cpm_ParaManagerCART::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 271 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 272 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_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

6.8.1 説明

CPM のローカル領域情報クラス

cpm_DomainInfo.h の 278 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_LocalDomainInfo::cpm_LocalDomainInfo ( )

コンストラクタ

cpm_DomainInfo.cpp の 501 行で定義されています。

6.8.2.2 cpm_LocalDomainInfo::~cpm_LocalDomainInfo ( ) [virtual]

デストラクタ

cpm_DomainInfo.cpp の 508 行で定義されています。

6.8.3 関数

6.8.3.1 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 515 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

6.9 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

6.9.1 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 33 行で定義されています。

6.9.2 型定義

6.9.2.1 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 47 行で定義されています。

6.9.2.2 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 43 行で定義されています。

6.9.3 コンストラクタとデストラクタ

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

コンストラクタ

cpm_ObjList.h の 60 行で定義されています。

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

デストラクタ

cpm_ObjList.h の 68 行で定義されています。

6.9.4 関数

6.9.4.1 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_ParaManager::cpm_Irecv(), と cpm_ParaManager::cpm_Isend().

6.9.4.2 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_ParaManager::cpm_Irecv(), と cpm_ParaManager::cpm_Isend().

6.9.4.3 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_ParaManager::cpm_Wait(), cpm_ParaManager::cpm_wait_BndCommS3D(), cpm_ParaManager::cpm_wait_BndCommS4D(), cpm_ParaManager::cpm_wait_BndCommS4DEx(), cpm_ParaManager::cpm_wait_BndCommV3D(), cpm_ParaManager::cpm_wait_BndCommV3DEx(), と cpm_ParaManager::cpm_Waitall().

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

6.9.5 変数

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

cpm_ObjList.h の 48 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().

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

使用可能な登録番号

cpm_ObjList.h の 51 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().

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

cpm_ObjList.h の 44 行で定義されています。

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

6.10 クラス cpm_ParaManager

`#include <cpm_ParaManager.h>`

cpm_ParaManager に対する継承グラフ

cpm_ParaManager のコラボレーション図

Public メソッド

- cpm_ErrorCode Initialize ()
- cpm_ErrorCode Initialize (int &argc, char **&argv)
- bool IsParallel ()
- bool IsParallel () const
- 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 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 VoxelInit_LMR (std::string treeFile, size_t maxVC=1, size_t maxN=3, int procGrpNo=0)
- int CreateProcessGroup (int nproc, int *proclist, int parentProcGrpNo=0)
- cpm_DomainType GetDomainType ()
- const cpm_VoxelInfo * FindVoxelInfo (int procGrpNo=0)
- const int * GetDivNum (int procGrpNo=0)
- const double * GetPitch (int procGrpNo=0)
- const int * GetGlobalVoxelSize (int procGrpNo=0)
- const double * GetGlobalOrigin (int procGrpNo=0)
- const double * GetGlobalRegion (int procGrpNo=0)
- const int * GetLocalVoxelSize (int procGrpNo=0)
- const double * GetLocalOrigin (int procGrpNo=0)
- const double * GetLocalRegion (int procGrpNo=0)
- const int * GetDivPos (int procGrpNo=0)
- const int * GetVoxelHeadIndex (int procGrpNo=0)
- const int * GetVoxelTailIndex (int procGrpNo=0)
- const int * GetNeighborRankID (int procGrpNo=0)
- const int * GetPeriodicRankID (int procGrpNo=0)
- const int * GetNeighborRankList (cpm_FaceFlag face, int &num, int procGrpNo=0)
- const int * GetPeriodicRankList (cpm_FaceFlag face, int &num, int procGrpNo=0)
- int GetNeighborLevelDiff (cpm_FaceFlag face, int procGrpNo=0)
- virtual bool GetBndIndexExtGc (int id, int *array, int vc, int &ista, int &jsta, int &ksta, int &ilen, int &jlen, int &klen, int procGrpNo=0)
- virtual bool GetBndIndexExtGc (int id, int *array, int imax, int jmax, int kmax, int vc, int &ista, int &jsta, int &ksta, int &ilen, int &jlen, int &klen, int procGrpNo=0)
- bool IsOuterBoundary (cpm_FaceFlag face, int procGrpNo=0)
- bool IsInnerBoundary (cpm_FaceFlag face, int procGrpNo=0)
- int GetMyRankID (int procGrpNo=0)
- int GetNumRank (int procGrpNo=0)
- std::string GetHostName ()
- MPI_Comm GetMPI_Comm (int procGrpNo=0)

- void Abort (int errorcode)
- cpm_ErrorCode Barrier (int procGrpNo=0)
- cpm_ErrorCode Wait (MPI_Request *request)
- cpm_ErrorCode Waitall (int count, MPI_Request requests[])
- template<class T >
  CPM_INLINE cpm_ErrorCode Bcast (T *buf, int count, int root, int procGrpNo=0)
- cpm_ErrorCode Bcast (MPI_Datatype dtype, void *buf, int count, int root, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode Send (T *buf, int count, int dest, int procGrpNo=0)
- cpm_ErrorCode Send (MPI_Datatype dtype, void *buf, int count, int dest, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode Recv (T *buf, int count, int source, int procGrpNo=0)
- cpm_ErrorCode Recv (MPI_Datatype dtype, void *buf, int count, int source, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode Isend (T *buf, int count, int dest, MPI_Request *request, int procGrpNo=0)
- cpm_ErrorCode Isend (MPI_Datatype dtype, void *buf, int count, int dest, MPI_Request *request, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode Irecv (T *buf, int count, int source, MPI_Request *request, int procGrpNo=0)
- cpm_ErrorCode Irecv (MPI_Datatype dtype, void *buf, int count, int source, MPI_Request *request, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode Allreduce (T *sendbuf, T *recvbuf, int count, MPI_Op op, int procGrpNo=0)
- cpm_ErrorCode Allreduce (MPI_Datatype dtype, void *sendbuf, void *recvbuf, int count, MPI_Op op, int procGrpNo=0)
- template<class Ts , class Tr >
  CPM_INLINE cpm_ErrorCode Gather (Ts *sendbuf, int sendcnt, Tr *recvbuf, int recvcnt, int root, int procGrpNo=0)
- cpm_ErrorCode Gather (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int recvcnt, int root, int procGrpNo=0)
- template<class Ts , class Tr >
  CPM_INLINE cpm_ErrorCode Allgather (Ts *sendbuf, int sendcnt, Tr *recvbuf, int recvcnt, int procGrpNo=0)
- cpm_ErrorCode Allgather (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int recvcnt, int procGrpNo=0)
- template<class Ts , class Tr >
  CPM_INLINE cpm_ErrorCode Gatherv (Ts *sendbuf, int sendcnt, Tr *recvbuf, int *recvcnts, int *displs, int root, int procGrpNo=0)
- cpm_ErrorCode Gatherv (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int *recvcnts, int *displs, int root, int procGrpNo=0)
- template<class Ts , class Tr >
  CPM_INLINE cpm_ErrorCode Allgatherv (Ts *sendbuf, int sendcnt, Tr *recvbuf, int *recvcnts, int *displs, int procGrpNo=0)
- cpm_ErrorCode Allgatherv (MPI_Datatype stype, void *sendbuf, int sendcnt, MPI_Datatype rtype, void *recvbuf, int *recvcnts, int *displs, int procGrpNo=0)
- cpm_ErrorCode cpm_Wait (int reqNo)
- cpm_ErrorCode cpm_Waitall (int count, int reqNoList[])
- cpm_ErrorCode cpm_Isend (void *buf, int count, int datatype, int dest, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_Irecv (void *buf, int count, int datatype, int source, int *reqNo, int procGrpNo=0)
- cpm_ErrorCode cpm_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)
- 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_ErrorCode BndCommS3D (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommV3D (T ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- cpm_ErrorCode BndCommV3D (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommS4D (T ∗array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo=0)
- cpm_ErrorCode BndCommS4D (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommS3D_nowait (T ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- cpm_ErrorCode BndCommS3D_nowait (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommV3D_nowait (T ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- cpm_ErrorCode BndCommV3D_nowait (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], 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[48], int procGrpNo=0)
- cpm_ErrorCode BndCommS4D_nowait (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode wait_BndCommS3D (T ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- cpm_ErrorCode wait_BndCommS3D (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode wait_BndCommV3D (T ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- cpm_ErrorCode wait_BndCommV3D (MPI_Datatype dtype, void ∗array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode wait_BndCommS4D (T ∗array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)

- cpm_ErrorCode wait_BndCommS4D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode PeriodicCommS3D (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- cpm_ErrorCode PeriodicCommS3D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode PeriodicCommV3D (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- cpm_ErrorCode PeriodicCommV3D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode PeriodicCommS4D (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- cpm_ErrorCode PeriodicCommS4D (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommV3DEx (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- cpm_ErrorCode BndCommV3DEx (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommS4DEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- cpm_ErrorCode BndCommS4DEx (MPI_Datatype dtype, void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommV3DEx_nowait (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- cpm_ErrorCode BndCommV3DEx_nowait (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommS4DEx_nowait (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[48], 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 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_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)
- 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_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)
- template<class T >
  CPM_INLINE cpm_ErrorCode PeriodicCommV3DEx (T *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- cpm_ErrorCode PeriodicCommV3DEx (MPI_Datatype dtype, void *array, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode PeriodicCommS4DEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)
- cpm_ErrorCode PeriodicCommS4DEx (MPI_Datatype dtype, void *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, cpm_DirFlag dir, cpm_PMFlag pm, int procGrpNo=0)

- template<class T >
  void InitArray (T *array, size_t size)
- template<class T >
  void CopyArray (T *source, T *dist, size_t size)
- double * AllocDoubleS3D (int vc, int procGrpNo=0)
- float * AllocFloatS3D (int vc, int procGrpNo=0)
- int * AllocIntS3D (int vc, int procGrpNo=0)
- double * AllocDoubleV3D (int vc, int procGrpNo=0)
- float * AllocFloatV3D (int vc, int procGrpNo=0)
- int * AllocIntV3D (int vc, int procGrpNo=0)
- double * AllocDoubleV3DEx (int vc, int procGrpNo=0)
- float * AllocFloatV3DEx (int vc, int procGrpNo=0)
- int * AllocIntV3DEx (int vc, int procGrpNo=0)
- double * AllocDoubleS4D (int nmax, int vc, int procGrpNo=0)
- float * AllocFloatS4D (int nmax, int vc, int procGrpNo=0)
- int * AllocIntS4D (int nmax, int vc, int procGrpNo=0)
- double * AllocDoubleS4DEx (int nmax, int vc, int procGrpNo=0)
- float * AllocFloatS4DEx (int nmax, int vc, int procGrpNo=0)
- int * AllocIntS4DEx (int nmax, int vc, int procGrpNo=0)
- void flush (std::ostream &out, int procGrpNo=0)
- void flush (FILE *fp, int procGrpNo=0)
- 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

- static cpm_ParaManager * get_instance ()
- static cpm_ParaManager * get_instance (int &argc, char **&argv, cpm_DomainType domainType=CPM_DOMAIN_CARTESIAN
- static cpm_ParaManager * get_instance (cpm_DomainType domainType)
- 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)

Protected メソッド

- cpm_ParaManager ()
- virtual ~cpm_ParaManager ()

Protected 変数

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

フレンド

- class C_PARAMANAGER

6.10.1 説明

CPM の並列管理クラス

- 現時点ではユーザがインスタンスすることを許していない
- get_instance 静的関数を用いて唯一のインスタンスを取得する

cpm_ParaManager.h の 38 行で定義されています。

6.10.2 コンストラクタとデストラクタ

6.10.2.1 cpm_ParaManager::cpm_ParaManager ( ) [protected]

コンストラクタ

cpm_ParaManager.cpp の 139 行で定義されています。

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

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

デストラクタ

cpm_ParaManager.cpp の 159 行で定義されています。

参照先 m_procGrpList, と m_voxelInfoMap.

6.10.3 関数

6.10.3.1 void cpm_ParaManager::Abort (int *errorcode*)

Abort

- MPI_Abort のインターフェイス

  引数

in	*errorcode*	MPI_Abort に渡すエラーコード

cpm_ParaManager_MPI.cpp の 165 行で定義されています。

参照元 cpm_Abort_(), cpm_ParaManagerCART::VoxelInit(), と cpm_ParaManagerLMR::VoxelInit_LMR().

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

Allgather

- MPI_Allgather のインターフェイス

  引数

in	*sendbuf*	送信データ

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 206 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Allgather_().

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

Allgather

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 451 行で定義されています。

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

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

Allgatherv

- MPI_Allgatherv のインターフェイス

引数

in	*sendbuf*	送信データ
in	*sendcnt*	送信データのサイズ
out	*recvbuf*	受信データ

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 250 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_().

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

Allgatherv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 510 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().

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

配列確保 double(imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocDoubleS4D().

6.10.3.7 double ∗ cpm_ParaManager::AllocDoubleS4D (int *nmax,* int *vc,* int *procGrpNo* = 0)

配列確保 double(imax,jmax,kmax,nmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

cpm_ParaManager_Alloc.cpp の 24 行で定義されています。

参照先 GetLocalVoxelSize().

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

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

配列確保 double(nmax,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocDoubleS4D().

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

配列確保 double(imax,jmax,kmax,3)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocDoubleS4D().

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

配列確保 double(3,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocDoubleS4D().

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

配列確保 float(imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocFloatS4D().

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

配列確保 float(imax,jmax,kmax,nmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 GetLocalVoxelSize().

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

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

配列確保 float(nmax,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocFloatS4D().

6.10.3.14 float ∗ cpm_ParaManager::AllocFloatV3D (int *vc,* int *procGrpNo* = 0)

配列確保 float(imax,jmax,kmax,3)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocFloatS4D().

6.10.3.15 float ∗ cpm_ParaManager::AllocFloatV3DEx (int *vc,* int *procGrpNo* = 0)

配列確保 float(3,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocFloatS4D().

6.10.3.16 int ∗ cpm_ParaManager::AllocIntS3D (int *vc,* int *procGrpNo* = 0)

配列確保 int(imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocIntS4D().

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

配列確保 int(imax,jmax,kmax,nmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 GetLocalVoxelSize().

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

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

配列確保 int(nmax,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocIntS4D().

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

配列確保 int(imax,jmax,kmax,3)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocIntS4D().

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

配列確保 int(3,imax,jmax,kmax)

引数

in	*vc*	仮想セル数
in	*procGrpNo*	プロセスグループ番号

戻り値

配列ポインタ

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

参照先 AllocIntS4D().

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

Allreduce

- MPI_Allreduce のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 167 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Allreduce_(), と cpm_ParaManagerCART::GetBndIndexExtGc().

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

Allreduce

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 395 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().

6.10.3.23 cpm_ErrorCode cpm_ParaManager::Barrier (int *procGrpNo* = 0)

Barrier

- MPI_Barrier のインターフェイス

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 175 行で定義されています。

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

参照元 cpm_Barrier_(), と Initialize().

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

Bcast

- MPI_Bcast のインターフェイス

引数

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

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Bcast_().

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

Bcast

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

引数

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_ParaManager_MPI.cpp の 253 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().

6.10.3.26 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS3D (T * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* int *procGrpNo* = 0)

袖通信 (Scalar3D 版)

- (imax,jmax,kmax) の形式の配列の袖通信を行う

引数

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_ParaManager_BndComm.h の 26 行で定義されています。

参照先 BndCommS4D().

参照元 cpm_BndCommS3D_().

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

袖通信 (Scalar3D 版, MPI_Datatype 指定)

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

引数

in	*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_ParaManager_MPI.cpp の 539 行で定義されています。

参照先 BndCommS4D().

6.10.3.28 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS3D_nowait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar3D 版)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (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_BndComm.h の 64 行で定義されています。

参照先 BndCommS4D_nowait().

参照元 cpm_BndCommS3D_nowait().

6.10.3.29 cpm_ErrorCode cpm_ParaManager::BndCommS3D_nowait (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar3D 版, MPI_Datatype 指定)

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

引数

in	*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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う

引数

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_ParaManager_BndComm.h の 44 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerCART::BndCommS4D(), CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_BNDCOMM, と GetDomainType().

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

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

袖通信 (Scalar4D 版, MPI_Datatype 指定)

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

引数

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_ParaManager_MPI.cpp の 557 行で定義されています。

参照先 BndCommS4D(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.10.3.32 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS4D_nowait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4D 版)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (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_BndComm.h の 84 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerCART::BndCommS4D_nowait(), CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_BNDCOMM, と GetDomainType().

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

6.10.3.33 cpm_ErrorCode cpm_ParaManager::BndCommS4D_nowait (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4D 版, MPI_Datatype 指定)

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

- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS4D をコールする

引数

in	*dtype*	袖通信データのMPI_Datatype
in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*nmax*	配列サイズ (成分数)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (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)

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

袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

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

参照先 cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerCART::BndCommS4DEx(), CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_BNDCOMM, と GetDomainType().

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

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

袖通信 (Scalar4DEx 版, MPI_Datatype 指定)

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

引数

in	*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_ParaManager_MPI.cpp の 796 行で定義されています。

参照先 BndCommS4DEx(), と CPM_ERROR_MPI_INVALID_DATATYPE.

6.10.3.36 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommS4DEx_nowait (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4DEx 版)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*nmax*	配列サイズ (成分数)
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (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_BndCommEx.h の 65 行で定義されています。

参照先 cpm_ParaManagerLMR::BndCommS4DEx_nowait(), cpm_ParaManagerCART::BndCommS4DEx_nowait(), CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_BNDCOMM, と GetDomainType().

参照元 BndCommV3DEx_nowait(), と cpm_BndCommS4DEx_nowait().

6.10.3.37 cpm_ErrorCode cpm_ParaManager::BndCommS4DEx_nowait (MPI_Datatype *dtype,* void * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

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

- (nmax,imax,jmax,kmax) の形式の配列の非同期袖通信を行う

- MPI_Datatype を指定するバージョン
- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS4DEx をコールする

引数

in	*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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

袖通信 (Vector3D 版)

- (imax,jmax,kmax,3) の形式の配列の袖通信を行う

引数

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

参照先 BndCommS4D().

参照元 cpm_BndCommV3D_().

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

袖通信 (Vector3D 版, MPI_Datatype 指定)

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

引数

in	*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_ParaManager_MPI.cpp の 548 行で定義されています。

参照先 BndCommS4D().

6.10.3.40 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommV3D_nowait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Vector3D 版)

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

引数

in	*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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 74 行で定義されています。

参照先 BndCommS4D_nowait().

参照元 cpm_BndCommV3D_nowait().

6.10.3.41 cpm_ErrorCode cpm_ParaManager::BndCommV3D_nowait (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Vector3D 版, MPI_Datatype 指定)

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

引数

in	*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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

袖通信 (Vector3DEx 版)

- (3,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

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_ParaManager_BndCommEx.h の 26 行で定義されています。

参照先 BndCommS4DEx().

参照元 cpm_BndCommV3DEx_().

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

袖通信 (Vector3DEx 版, MPI_Datatype 指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype

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_ParaManager_MPI.cpp の 787 行で定義されています。

参照先 BndCommS4DEx().

6.10.3.44 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::BndCommV3DEx_nowait (T * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Vector3DEx 版)

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
out	*req*	MPI リクエスト (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_BndCommEx.h の 55 行で定義されています。

参照先 BndCommS4DEx_nowait().

参照元 cpm_BndCommV3DEx_nowait().

6.10.3.45 cpm_ErrorCode cpm_ParaManager::BndCommV3DEx_nowait (MPI_Datatype *dtype,* void * *array,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* MPI_Request *req[48],* int *procGrpNo* = 0)

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

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

引数

in	*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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

6.10.3.47 template<class T > void cpm_ParaManager::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

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

cpm_BndCommS3D_nowait

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

引数

in	*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 2677 行で定義されています。

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

参照元 cpm_BndCommS3D_nowait_().

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

cpm_BndCommS4D_nowait

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

引数

in	*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 2761 行で定義されています。

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

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

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

cpm_BndCommS4DEx_nowait

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

引数

in	*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 2975 行で定義されています。

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

参照元 cpm_BndCommS4DEx_nowait_(), と cpm_BndCommV3DEx_nowait().

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

cpm_BndCommV3D_nowait

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

引数

in	*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 2719 行で定義されています。

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

参照元 cpm_BndCommV3D_nowait_().

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

cpm_BndCommV3DEx_nowait

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

引数

in	*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 2933 行で定義されています。

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

参照元 cpm_BndCommV3DEx_nowait_().

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

cpm_Irecv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2645 行で定義されています。

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

参照元 cpm_Irecv_().

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

cpm_Isend

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

引数

in	*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_ParaManager_frtIF.cpp の 2608 行で定義されています。

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

参照元 cpm_Isend_().

6.10.3.55 cpm_ErrorCode cpm_ParaManager::cpm_Wait (int *reqNo*)

cpm_Wait

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

引数

in	*reqNo*	リクエスト番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2544 行で定義されています。

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

参照元 cpm_Wait_().

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

cpm_wait_BndCommS3D

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

引数

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

in	*jmax*	配列サイズ (J 方向)
in	*kmax*	配列サイズ (K 方向)
in	*vc*	仮想セル数
in	*vc_comm*	通信する仮想セル数
in	*datatype*	袖通信データのデータタイプ (cpm_fparam.fi 参照)
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 2799 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_SUCCESS, cpm_-wait_BndCommS4D(), cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), GetMPI_Datatype(), m_reqList, と wait_BndCommS3D().

参照元 cpm_wait_BndCommS3D_().

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

cpm_wait_BndCommS4D

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

引数

in	*array*	袖通信をする配列の先頭ポインタ
in	*imax*	配列サイズ (I 方向)
in	*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 2891 行で定義されています。

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

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

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

cpm_wait_BndCommS4DEx

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

引数

in	*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 3059 行で定義されています。

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

参照元 cpm_wait_BndCommS4DEx_(), と cpm_wait_BndCommV3DEx().

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

cpm_wait_BndCommV3D

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

引数

in	*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 2845 行で定義されています。

参照先 CPM_ERROR_INVALID_OBJKEY, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_SUCCESS, cpm_-wait_BndCommS4D(), cpm_ObjList< T >::Delete(), cpm_ObjList< T >::Get(), GetMPI_Datatype(), m_reqList, と wait_BndCommV3D().

参照元 cpm_wait_BndCommV3D_().

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

cpm_wait_BndCommV3DEx

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

引数

in	*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 3013 行で定義されています。

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

参照元 cpm_wait_BndCommV3DEx_().

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

cpm_Waitall

- MPI_Waitall のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_frtIF.cpp の 2567 行で定義されています。

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

参照元 cpm_Waitall_().

6.10.3.62 int cpm_ParaManager::CreateProcessGroup (int *nproc*, int * *proclist*, int *parentProcGrpNo* = 0)

プロセスグループの作成

- 指定されたプロセスリストを使用してプロセスグループを生成する

引数

in	*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	*parentProcGrp-No*	親とするプロセスグループ番号 (省略時 0)

戻り値

***0* 以上**	生成されたプロセスグループ番号
-1	エラー

cpm_ParaManager.cpp の 339 行で定義されています。

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

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

VOXEL 空間マップを検索

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

VOXEL 空間情報ポインタ

cpm_ParaManager.cpp の 389 行で定義されています。

参照先 m_voxelInfoMap.

参照元 GetDivNum(), GetDivPos(), GetGlobalOrigin(), GetGlobalRegion(), GetGlobalVoxelSize(), GetLocal-Origin(), GetLocalRegion(), GetLocalVoxelSize(), GetNeighborLevelDiff(), GetNeighborRankID(), GetNeighbor-RankList(), GetPeriodicRankID(), GetPeriodicRankList(), GetPitch(), GetVoxelHeadIndex(), GetVoxelTailIndex(), IsInnerBoundary(), と IsOuterBoundary().

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

flush

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

flush

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

Gather

- MPI_Gather のインターフェイス

  引数

in	*sendbuf*	送信データ

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 184 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Gather_().

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

Gather

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

引数

in	*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_ParaManager_MPI.cpp の 422 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().

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

Gatherv

- MPI_Gatherv のインターフェイス

引数

in	*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_ParaManager_inline.h の 228 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Gatherv_().

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

Gatherv

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_MPI.cpp の 480 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().

6.10.3.70 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 48 行で定義されています。

参照先 C_PARAMANAGER::get_instance(), と C_PARAMANAGER::pParaManager.

参照元 cpm_Abort_(), cpm_Allgather_(), cpm_Allgatherv_(), cpm_Allreduce_(), cpm_Barrier_(), cpm_Bcast_(), cpm_BndCommS3D_(), cpm_BndCommS3D_nowait_(), cpm_BndCommS4D_(), cpm_BndCommS4D_nowait_(),

cpm_BndCommS4DEx_(), cpm_BndCommS4DEx_nowait_(), cpm_BndCommV3D_(), cpm_BndCommV3D_nowait_(), cpm_BndCommV3DEx_(), cpm_BndCommV3DEx_nowait_(), cpm_Gather_(), cpm_Gatherv_(), cpm_GetDivNum_(), cpm_GetDivPos_(), cpm_GetGlobalOrigin_(), cpm_GetGlobalRegion_(), cpm_GetGlobalVoxelSize_(), cpm_GetLocalOrigin_(), cpm_GetLocalRegion_(), cpm_GetLocalVoxelSize_(), cpm_GetMyRankID_(), cpm_GetNeighborRankID_(), cpm_GetNumRank_(), cpm_GetPeriodicRankID_(), cpm_GetPitch_(), cpm_GetVoxelHeadIndex_(), cpm_GetVoxelTailIndex_(), cpm_Initialize_(), cpm_Irecv_(), cpm_Isend_(), cpm_IsParallel_(), cpm_PeriodicCommS3D_(), cpm_PeriodicCommS4D_(), cpm_PeriodicCommS4DEx_(), cpm_PeriodicCommV3D_(), cpm_PeriodicCommV3DEx_(), cpm_Recv_(), cpm_Send_(), cpm_SetBndCommBuffer_(), cpm_VoxelInit_(), cpm_VoxelInit_nodiv_(), cpm_Wait_(), cpm_wait_BndCommS3D_(), cpm_wait_BndCommS4D_(), cpm_wait_BndCommS4DEx_(), cpm_wait_BndCommV3D_(), cpm_wait_BndCommV3DEx_(), と cpm_Waitall_().

6.10.3.71 cpm_ParaManager ∗ cpm_ParaManager::get_instance (int & *argc,* char ∗∗& *argv,* cpm_DomainType *domainType* = CPM_DOMAIN_CARTESIAN) [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
in	*domainType*	インスタンスする領域タイプ (デフォルト=カーテシアン)

戻り値

インスタンスのポインタ

cpm_ParaManager.cpp の 61 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_SUCCESS, C_PARAMANAGER::get_instance(), Initialize(), と C_PARAMANAGER::pParaManager.

6.10.3.72 cpm_ParaManager ∗ cpm_ParaManager::get_instance (cpm_DomainType *domainType*) [static]

唯一のインスタンスの取得 (initialize 処理も実行. fortan インターフェイス用)

引数

in	*domainType*	インスタンスする領域タイプ (デフォルト=カーテシアン)

戻り値

インスタンスのポインタ

cpm_ParaManager.cpp の 101 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_SUCCESS, C_PARAMANAGER::get_instance(), Initialize(), と C_PARAMANAGER::pParaManager.

6.10.3.73 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_ParaManagerCART, と cpm_ParaManagerLMRで再定義されています。

cpm_ParaManager.cpp の 648 行で定義されています。

6.10.3.74 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) [virtual]

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)

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_ParaManagerCARTで再定義されています。

cpm_ParaManager.cpp の 591 行で定義されています。

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

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	*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_ParaManagerCARTで再定義されています。

cpm_ParaManager.cpp の 602 行で定義されています。

6.10.3.76 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 399 行で定義されています。

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

参照元 cpm_GetDivNum_().

6.10.3.77 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 495 行で定義されています。

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

参照元 cpm_GetDivPos_().

6.10.3.78 cpm_DomainType cpm_ParaManager::GetDomainType ( )

領域分割タイプを取得

戻り値

領域分割タイプ

cpm_ParaManager.cpp の 381 行で定義されています。

参照先 m_domainType.

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

6.10.3.79 const double * cpm_ParaManager::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_ParaManager.cpp の 435 行で定義されています。

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

参照元 cpm_GetGlobalOrigin_().

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

全体空間サイズを取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

全体空間サイズ実数配列のポインタ

cpm_ParaManager.cpp の 447 行で定義されています。

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

参照元 cpm_GetGlobalRegion_().

6.10.3.81 const int * cpm_ParaManager::GetGlobalVoxelSize (int *procGrpNo* = 0)

全体ボクセル数を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

全体ボクセル数の整数配列ポインタ (3word)

cpm_ParaManager.cpp の 423 行で定義されています。

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

参照元 cpm_GetGlobalVoxelSize_(), と cpm_ParaManagerCART::GetBndIndexExtGc().

6.10.3.82 std::string cpm_ParaManager::GetHostName ( )

ホスト名の取得

- 自ランクのホスト名を取得

  戻り値

  ホスト名

cpm_ParaManager_MPI.cpp の 139 行で定義されています。

6.10.3.83 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 471 行で定義されています。

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

参照元 cpm_GetLocalOrigin_(), と cpm_GetLocalRegion_().

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

自ランクの空間サイズを取得

引数

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 483 行で定義されています。

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

6.10.3.85 const int * cpm_ParaManager::GetLocalVoxelSize (int *procGrpNo* = 0)

自ランクのボクセル数を取得

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

ローカルボクセル数の整数配列ポインタ (3word)

cpm_ParaManager.cpp の 459 行で定義されています。

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

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

6.10.3.86 MPI_Comm cpm_ParaManager::GetMPI_Comm (int *procGrpNo* = 0)

MPI コミュニケータの取得

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

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

MPI コミュニケータ

cpm_ParaManager_MPI.cpp の 150 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_ParaManagerCART::VoxelInit(), と cpm_ParaManagerLMR::VoxelInit_LMR().

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

MPI_Datatype を取得

引数

in	*ptr*	取得したいデータのポインタ

戻り値

MPI_Datatype

cpm_ParaManager_inline.h の 46 行で定義されています。

参照元 Allgather(), Allgatherv(), Allreduce(), Bcast(), cpm_Allgather_(), cpm_Allgatherv_(), cpm_Allreduce_(), cpm_Bcast_(), cpm_BndCommS3D_(), cpm_BndCommS3D_nowait(), cpm_BndCommS4D_(), cpm_BndCommS4D_nowait(), cpm_BndCommS4DEx_(), cpm_BndCommS4DEx_nowait(), cpm_BndCommV3D_(), cpm_BndCommV3D_nowait(), cpm_BndCommV3DEx_(), cpm_BndCommV3DEx_nowait(), cpm_Gather_(), cpm_Gatherv_(), cpm_Irecv(), cpm_Isend(), cpm_PeriodicCommS3D_(), cpm_PeriodicCommS4D_(), cpm_PeriodicCommS4DEx_(), cpm_PeriodicCommV3D_(), cpm_PeriodicCommV3DEx_(), cpm_Recv_(), cpm_Send_(), cpm_wait_BndCommS3D(), cpm_wait_BndCommS4D(), cpm_wait_BndCommS4DEx(), cpm_wait_BndCommV3D(), cpm_wait_BndCommV3DEx(), Gather(), Gatherv(), Irecv(), Isend(), Recv(), と Send().

6.10.3.88 MPI_Datatype cpm_ParaManager::GetMPI_Datatype (int *datatype*) [static]

MPI_Datatype を取得

- Fortran データタイプからMPI_Datatype を取得

引数

in	*datatype*	取得したいデータのポインタ

戻り値

MPI_Datatype

cpm_ParaManager_MPI.cpp の 28 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().

6.10.3.89 MPI_Op cpm_ParaManager::GetMPI_Op (int *op*) [static]

MPI_Op を取得

- Fortran オペレータタイプからMPI_Op を取得

引数

in	*op*	取得したいデータのポインタ

戻り値

MPI_Op

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

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

参照元 cpm_Allreduce_().

6.10.3.90 int cpm_ParaManager::GetMyRankID (int *procGrpNo* = 0)

ランク番号の取得

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

引数

in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

ランク番号

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

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

参照元 cpm_GetMyRankID_().

6.10.3.91 int cpm_ParaManager::GetNeighborLevelDiff (cpm_FaceFlag *face,* int *procGrpNo* = 0)

指定面におけるレベル差を取得

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

レベル差 (0:同じレベル, 1:fine, -1:coarse)

cpm_ParaManager.cpp の 579 行で定義されています。

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

参照元 cpm_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerLMR::BndCommS4DEx_nowait(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), cpm_ParaManagerLMR::SetBndCommBuffer(), cpm_ParaManagerLMR::wait_BndCommS4D(), と cpm_ParaManagerLMR::wait_BndCommS4DEx().

6.10.3.92 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 531 行で定義されています。

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

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

6.10.3.93 const int ∗ cpm_ParaManager::GetNeighborRankList (cpm_FaceFlag *face*, int & *num*, int *procGrpNo* = 0)

指定面における自ランク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)
in	*procGrpNo*	プロセスグループ番号 (省略時=0)

戻り値

指定面における自ランクの隣接ランク番号整数配列のポインタ

cpm_ParaManager.cpp の 555 行で定義されています。

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

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

6.10.3.94 int cpm_ParaManager::GetNumRank (int *procGrpNo* = 0)

ランク数の取得

- プロセスグループのランク数を取得する

  引数

in	*procGrpNo*	プロセスグループ番号 (省略時 0)

  戻り値

  ランク数

cpm_ParaManager_MPI.cpp の 111 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_ParaManagerCART::VoxelInit(), と cpm_ParaManagerCART::VoxelInit_-Subdomain().

6.10.3.95 const int * cpm_ParaManager::GetPeriodicRankID (int *procGrpNo* = 0)

自ランクの周期境界の隣接ランク番号を取得

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 543 行で定義されています。

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

参照元 cpm_GetPeriodicRankID_(), cpm_ParaManagerCART::PeriodicCommS4D(), と cpm_ParaManagerCART-::PeriodicCommS4DEx().

6.10.3.96 const int * cpm_ParaManager::GetPeriodicRankList (cpm_FaceFlag *face*, int & *num*, int *procGrpNo* = 0)

指定面における自ランク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)
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 567 行で定義されています。

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

参照元 cpm_ParaManagerLMR::PeriodicCommS4D(), と cpm_ParaManagerLMR::PeriodicCommS4DEx().

6.10.3.97 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 411 行で定義されています。

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

参照元 cpm_GetPitch_().

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

自ランクの始点VOXEL の全体空間でのインデクスを取得

- 全体空間の先頭インデクスを 0 としたC 型のインデクス

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 507 行で定義されています。

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

参照元 cpm_GetVoxelHeadIndex_(), と cpm_ParaManagerCART::GetBndIndexExtGc().

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

自ランクの終点VOXEL の全体空間でのインデクスを取得

- 全体空間の先頭インデクスを 0 としたC 型のインデクス

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 519 行で定義されています。

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

参照元 cpm_GetVoxelTailIndex_().

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

cpm_ParaManager_inline.h の 26 行で定義されています。

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

配列の初期化処理

引数

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

6.10.3.102 cpm_ErrorCode cpm_ParaManager::Initialize ( )

初期化処理 (MPI_Init は実行済みの場合)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 185 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 get_instance(), と Initialize().

6.10.3.103 cpm_ErrorCode cpm_ParaManager::Initialize (int & *argc,* char ∗∗& *argv*)

初期化処理 (MPI_Init も実行する)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager.cpp の 235 行で定義されています。

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

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

Irecv

- MPI_Irecv のインターフェイス

引数

out	*buf*	受信データ
in	*count*	受信データのサイズ

in	*source*	送信元のランク番号 (procGrpNo 内でのランク番号)
out	*request*	リクエストハンドル
in	*procGrpNo*	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_inline.h の 150 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_ParaManagerCART::sendrecv().

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

Irecv

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

引数

in	*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_ParaManager_MPI.cpp の 366 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().

6.10.3.106 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManager::Isend (T * *buf,* int *count,* int *dest,* MPI_Request * *request,* int *procGrpNo* = 0)

Isend

- MPI_Isend のインターフェイス

引数

in	*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_ParaManager_inline.h の 133 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_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerLMR::BndCommS4DEx_nowait(), cpm_Isend(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), と cpm_ParaManagerCART::sendrecv().

6.10.3.107 cpm_ErrorCode cpm_ParaManager::Isend (MPI_Datatype *dtype*, void * *buf*, int *count*, int *dest*, MPI_Request * *request*, int *procGrpNo* = 0)

Isend

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

引数

in	*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_ParaManager_MPI.cpp の 337 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().

6.10.3.108 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 627 行で定義されています。

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

6.10.3.109 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 614 行で定義されています。

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

6.10.3.110 bool cpm_ParaManager::IsParallel ( )

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

戻り値

true	並列実行
false	逐次実行

cpm_ParaManager.cpp の 259 行で定義されています。

参照先 m_nRank.

参照元 cpm_IsParallel_(), と Initialize().

6.10.3.111 bool cpm_ParaManager::IsParallel ( ) const

並列実行であるかチェックする (const)

- 並列実行であっても、並列数が 1 のときは false となる

  戻り値

true	並列実行
false	逐次実行

cpm_ParaManager.cpp の 271 行で定義されています。

参照先 m_nRank.

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

周期境界袖通信 (Scalar3D 版)

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

  引数

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

参照先 PeriodicCommS4D().

参照元 cpm_PeriodicCommS3D_().

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

周期境界袖通信 (Scalar3D 版, MPI_Datatype 指定)

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

引数

in	*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_ParaManager_MPI.cpp の 725 行で定義されています。

参照先 PeriodicCommS4D().

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

周期境界袖通信 (Scalar4D 版)

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

引数

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_ParaManager_BndComm.h の 166 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_BNDCOMM, GetDomainType(), cpm_ParaManagerLMR::PeriodicCommS4D(), と cpm_ParaManagerCART::PeriodicCommS4D().

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

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

周期境界袖通信 (Scalar4D 版, MPI_Datatype 指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
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_ParaManager_MPI.cpp の 743 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4D().

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

周期境界袖通信 (Scalar4DEx 版)

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

引数

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_ParaManager_BndCommEx.h の 127 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_PERIODIC, GetDomainType(), cpm_-ParaManagerLMR::PeriodicCommS4DEx(), と cpm_ParaManagerCART::PeriodicCommS4DEx().

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

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

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

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

引数

in	*dtype*	袖通信データのMPI_Datatype
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_ParaManager_MPI.cpp の 955 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と PeriodicCommS4DEx().

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

周期境界袖通信 (Vector3D 版)

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

引数

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_ParaManager_BndComm.h の 156 行で定義されています。

参照先 PeriodicCommS4D().

参照元 cpm_PeriodicCommV3D_().

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

周期境界袖通信 (Vector3D 版, MPI_Datatype 指定)

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

引数

in	*dtype*	袖通信データのMPI_Datatype
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_ParaManager_MPI.cpp の 734 行で定義されています。

参照先 PeriodicCommS4D().

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

周期境界袖通信 (Vector3DEx 版)

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

引数

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_ParaManager_BndCommEx.h の 117 行で定義されています。

参照先 PeriodicCommS4DEx().

参照元 cpm_PeriodicCommV3DEx_().

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

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

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

引数

in	*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_ParaManager_MPI.cpp の 946 行で定義されています。

参照先 PeriodicCommS4DEx().

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

Recv

- MPI_Recv のインターフェイス

引数

out	*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_ParaManager_inline.h の 116 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Recv_().

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

Recv

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

引数

in	*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_ParaManager_MPI.cpp の 308 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().

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

Send

- MPI_Send のインターフェイス

引数

in	*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_ParaManager_inline.h の 99 行で定義されています。

参照先 CPM_ERROR_MPI_INVALID_DATATYPE, と GetMPI_Datatype().

参照元 cpm_Send_().

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

Send

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

引数

in	*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_ParaManager_MPI.cpp の 280 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().

6.10.3.126 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_ParaManagerCART, と cpm_ParaManagerLMRで再定義されています。

cpm_ParaManager.cpp の 640 行で定義されています。

参照先 CPM_ERROR_DOMAINTYPE_SETBNDCOMMBUF.

参照元 cpm_SetBndCommBuffer_().

6.10.3.127 **cpm_ErrorCode cpm_ParaManager::VoxelInit (cpm_GlobalDomainInfo * *domainInfo*, size_t *maxVC* = 1, size_t *maxN* = 3, int *procGrpNo* = 0)** [virtual]

カーテシアン用の領域分割

- 既に作成済みの領域分割情報を用いた領域分割処理

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManagerCARTで再定義されています。

cpm_ParaManager.cpp の 283 行で定義されています。

参照先 CPM_ERROR_DOMAINTYPE_VOXELINIT.

参照元 cpm_VoxelInit_(), と cpm_VoxelInit_nodiv_().

6.10.3.128 **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]

カーテシアン用の領域分割

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

引数

in	*div*	領域分割数
in	*vox*	空間全体のボクセル数
in	*origin*	空間全体の原点
in	*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_ParaManagerCARTで再定義されています。

cpm_ParaManager.cpp の 291 行で定義されています。

参照先 CPM_ERROR_DOMAINTYPE_VOXELINIT.

6.10.3.129 **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]

カーテシアン用の領域分割

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

引数

in	*vox*	空間全体のボクセル数
in	*origin*	空間全体の原点
in	*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_ParaManagerCARTで再定義されています。

cpm_ParaManager.cpp の 300 行で定義されています。

参照先 CPM_ERROR_DOMAINTYPE_VOXELINIT.

6.10.3.130 cpm_ErrorCode cpm_ParaManager::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で再定義されています。

cpm_ParaManager.cpp の 330 行で定義されています。

参照先 CPM_ERROR_DOMAINTYPE_VOXELINIT.

6.10.3.131 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]

カーテシアン用の領域分割 (ActiveSubdomain 指定)

- 領域分割の各種情報を引数で渡して領域分割を行う
- ActiveSubdomain ファイルで指定される領域分割位置のランクが活性ドメインになる
- I,J,K 方向の領域分割数を指定するバージョン
- 指定の領域分割数とActiveSubdomain ファイルで指定されている領域分割数が一致している必要がある
- ActiveSubdomain 数と並列数が一致している必要がある

引数

in	*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_ParaManagerCARTで再定義されています。

cpm_ParaManager.cpp の 310 行で定義されています。

参照先 CPM_ERROR_DOMAINTYPE_VOXELINIT.

6.10.3.132 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]

カーテシアン用の領域分割 (ActiveSubdomain 指定)

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

引数

in	*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_ParaManagerCARTで再定義されています。

cpm_ParaManager.cpp の 320 行で定義されています。

参照先 CPM_ERROR_DOMAINTYPE_VOXELINIT.

6.10.3.133 cpm_ErrorCode cpm_ParaManager::Wait (MPI_Request * *request*)

Wait

- MPI_Wait のインターフェイス

引数

in	*request*	リクエストハンドル

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_REQUEST, CPM_ERROR_MPI_WAIT, と CPM_SUCCESS.

6.10.3.134 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 105 行で定義されています。

参照先 wait_BndCommS4D().

参照元 cpm_wait_BndCommS3D().

6.10.3.135 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.10.3.136 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 125 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_BNDCOMM, GetDomainType(), cpm_ParaManagerLMR::wait_BndCommS4D(), と cpm_ParaManagerCART::wait_BndCommS4D().

参照元 cpm_wait_BndCommS4D(), wait_BndCommS3D(), と wait_BndCommV3D().

6.10.3.137 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.10.3.138 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 96 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_ERROR_BNDCOMM, GetDomainType(), cpm_ParaManagerLMR::wait_BndCommS4DEx(), と cpm_ParaManagerCART::wait_BndCommS4DEx().

参照元 cpm_wait_BndCommS4DEx(), と wait_BndCommV3DEx().

6.10.3.139 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.10.3.140 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm.h の 115 行で定義されています。

参照先 wait_BndCommS4D().

参照元 cpm_wait_BndCommV3D().

6.10.3.141 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.10.3.142 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx.h の 86 行で定義されています。

参照先 wait_BndCommS4DEx().

参照元 cpm_wait_BndCommV3DEx().

6.10.3.143 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)

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

6.10.3.144 cpm_ErrorCode cpm_ParaManager::Waitall (int *count,* MPI_Request *requests[]*)

Waitall

- MPI_Waitall のインターフェイス

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

参照先 CPM_ERROR_MPI_WAITALL, と CPM_SUCCESS.

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

6.10.4 フレンドと関連する関数

6.10.4.1 friend class C_PARAMANAGER [friend]

cpm_ParaManager.h の 40 行で定義されています。

6.10.5 変数

6.10.5.1 cpm_DomainType cpm_ParaManager::m_domainType [protected]

領域分割タイプ

cpm_ParaManager.h の 1854 行で定義されています。

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

6.10.5.2 int cpm_ParaManager::m_nRank [protected]

プロセス並列数

cpm_ParaManager.h の 1848 行で定義されています。

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

6.10.5.3 std::vector<MPI_Comm> cpm_ParaManager::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_ParaManager.h の 1863 行で定義されています。

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

6.10.5.4 int cpm_ParaManager::m_rankNo [protected]

MPI_COMM_WORLD での自ランク番号

cpm_ParaManager.h の 1851 行で定義されています。

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

6.10.5.5 cpm_ObjList<MPI_Request> cpm_ParaManager::m_reqList [protected]

MPI_Request の管理マップ

- Fortran インターフェイス用

cpm_ParaManager.h の 1874 行で定義されています。

参照元 cpm_BndCommS3D_nowait(), cpm_BndCommS4D_nowait(), cpm_BndCommS4DEx_nowait(), cpm_BndCommV3D_nowait(), cpm_BndCommV3DEx_nowait(), cpm_Irecv(), cpm_Isend(), cpm_Wait(), cpm_wait_BndCommS3D(), cpm_wait_BndCommS4D(), cpm_wait_BndCommS4DEx(), cpm_wait_BndCommV3D(), cpm_wait_BndCommV3DEx(), と cpm_Waitall().

6.10.5.6 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 1869 行で定義されています。

参照元 cpm_ParaManager(), FindVoxelInfo(), cpm_ParaManagerCART::VoxelInit(), cpm_ParaManagerLMR::VoxelInit_LMR(), と ~cpm_ParaManager().

このクラスの説明は次のファイルから生成されました:

- cpm_ParaManager.h
- cpm_ParaManager.cpp
- cpm_ParaManager_Alloc.cpp
- cpm_ParaManager_frtIF.cpp
- cpm_ParaManager_MPI.cpp
- cpm_ParaManager_BndComm.h
- cpm_ParaManager_BndCommEx.h
- cpm_ParaManager_inline.h

6.11 クラス cpm_ParaManagerCART

`#include <cpm_ParaManagerCART.h>`

cpm_ParaManagerCART に対する継承グラフ

cpm_ParaManagerCART 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 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 bool GetBndIndexExtGc (int id, int *array, int vc, int &ista, int &jsta, int &ksta, int &ilen, int &jlen, int &klen, int procGrpNo=0)
- virtual bool GetBndIndexExtGc (int id, int *array, int imax, int jmax, int kmax, int vc, int &ista, int &jsta, int &ksta, int &ilen, int &jlen, int &klen, int procGrpNo=0)
- 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 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 BndCommS4D_nowait (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode wait_BndCommS4D (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode 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)
- 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)

- template<class T >
  CPM_INLINE cpm_ErrorCode BndCommS4DEx_nowait (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode wait_BndCommS4DEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, MPI_Request req[12], int procGrpNo=0)
- 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)
- template<class T >
  CPM_INLINE cpm_ErrorCode packX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_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 packXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode packZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)

Protected 型

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

Protected メソッド

- cpm_ParaManagerCART ()
- virtual ~cpm_ParaManagerCART ()
- 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, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode unpackX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode unpackY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode unpackZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode unpackXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode unpackYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode packZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *sendm, T *sendp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode unpackZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, T *recvm, T *recvp, int nIDm, int nIDp)
- template<class T >
  cpm_ErrorCode sendrecv (T *sendm, T *recvm, T *sendp, T *recvp, size_t nw, MPI_Request *req, int nIDsm, int nIDrm, int nIDsp, int nIDrp, int procGrpNo=0)

Protected 変数

- BndCommInfoMap m_bndCommInfoMap

フレンド

- class cpm_ParaManager

Additional Inherited Members

6.11.1 説明

カーテシアン用の並列管理クラス cpm_ParaManager 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_ParaManagerCART.h の 72 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_ParaManagerCART::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_ParaManagerCART.h の 392 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_ParaManagerCART::cpm_ParaManagerCART ( ) [protected]

コンストラクタ

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

参照先 CPM_DOMAIN_CARTESIAN, m_bndCommInfoMap, と cpm_ParaManager::m_domainType.

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

デストラクタ

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

参照先 m_bndCommInfoMap.

6.11.4 関数

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

袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の袖通信を行う

  引数

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_ParaManager_BndComm_CART.h の 47 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(), cpm_Para-Manager::GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDC-OMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_B-UFFER::m_nwZ, packX(), packY(), packZ(), sendrecv(), unpackX(), unpackY(), unpackZ(), cpm_ParaManager::-Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4D().

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

非同期版袖通信 (Scalar4D 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm_CART.h の 151 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(), cpm_Para-Manager::GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDC-OMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_B-UFFER::m_nwZ, packX(), packY(), packZ(), sendrecv(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4D_nowait().

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

袖通信 (Scalar4DEx 版)

- (nmax,imax,jmax,kmax) の形式の配列の袖通信を行う

引数

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_ParaManager_BndCommEx_CART.h の 53 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(), cpm_Para-Manager::GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BND-COMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOM-M_BUFFER::m_nwZ, packXEx(), packYEx(), packZEx(), sendrecv(), unpackXEx(), unpackYEx(), unpackZEx(), cpm_ParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4DEx().

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

非同期版袖通信 (Scalar4DEx 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx_CART.h の 157 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(), cpm_Para-Manager::GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BND-COMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOM-M_BUFFER::m_nwZ, packXEx(), packYEx(), packZEx(), sendrecv(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4DEx_nowait().

6.11.4.5 unsigned long long cpm_ParaManagerCART::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_ParaManagerCART.cpp の 368 行で定義されています。

参照元 DecideDivPattern_CommSize().

6.11.4.6 long long cpm_ParaManagerCART::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_ParaManagerCART.cpp の 473 行で定義されています。

参照元 DecideDivPattern_Cube().

6.11.4.7 cpm_ErrorCode cpm_ParaManagerCART::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_ParaManagerCART.cpp の 292 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().

6.11.4.8 cpm_ErrorCode cpm_ParaManagerCART::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_ParaManagerCART.cpp の 397 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().

6.11.4.9 CPM_INLINE S_BNDCOMM_BUFFER* cpm_ParaManagerCART::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_ParaManagerCART.h の 458 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().

6.11.4.10 size_t cpm_ParaManagerCART::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_ParaManagerを再定義しています。

cpm_ParaManagerCART.cpp の 657 行で定義されています。

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

6.11.4.11 bool cpm_ParaManagerCART::GetBndIndexExtGc (int *id*, int * *array*, int *vc*, int & *ista*, int & *jsta*, int & *ksta*, int & *ilen*, int & *jlen*, int & *klen*, int *procGrpNo* = 0) [virtual]

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)

  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を再定義しています。

cpm_ParaManagerCART.cpp の 498 行で定義されています。

参照先 cpm_ParaManager::GetLocalVoxelSize().

6.11.4.12 bool cpm_ParaManagerCART::GetBndIndexExtGc (int *id*, int * *array*, int *imax*, int *jmax*, int *kmax*, int *vc*, int & *ista*, int & *jsta*, int & *ksta*, int & *ilen*, int & *jlen*, int & *klen*, int *procGrpNo* = 0) [virtual]

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	*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を再定義しています。

cpm_ParaManagerCART.cpp の 518 行で定義されています。

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

6.11.4.13 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::packX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndComm_CART.h の 468 行で定義されています。

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

6.11.4.14 template<class T > cpm_ErrorCode cpm_ParaManagerCART::packX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*) [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
out	*sendm*	マイナス方向の送信バッファ
out	*sendp*	プラス方向の送信バッファ
in	*nIDm*	マイナス方向の隣接ランク番号
in	*nIDp*	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

6.11.4.15 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::packXEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndCommEx_CART.h の 474 行で定義されています。

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

6.11.4.16 template<class T > cpm_ErrorCode cpm_ParaManagerCART::packXEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*) [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
out	*sendm*	マイナス方向の送信バッファ
out	*sendp*	プラス方向の送信バッファ
in	*nIDm*	マイナス方向の隣接ランク番号
in	*nIDp*	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

6.11.4.17 template<class T > cpm_ErrorCode cpm_ParaManagerCART::packY (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*) [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
out	*sendm*	マイナス方向の送信バッファ
out	*sendp*	プラス方向の送信バッファ
in	*nIDm*	マイナス方向の隣接ランク番号
in	*nIDp*	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

6.11.4.18 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::packY (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndComm_CART.h の 528 行で定義されています。

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

6.11.4.19 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::packYEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndCommEx_CART.h の 535 行で定義されています。

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

6.11.4.20 template<class T > cpm_ErrorCode cpm_ParaManagerCART::packYEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*) [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
out	*sendm*	マイナス方向の送信バッファ
out	*sendp*	プラス方向の送信バッファ
in	*nIDm*	マイナス方向の隣接ランク番号
in	*nIDp*	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

6.11.4.21 template<class T > cpm_ErrorCode cpm_ParaManagerCART::packZ (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, T * *sendm*, T * *sendp*, int *nIDm*, int *nIDp*) [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

out	*sendm*	マイナス方向の送信バッファ
out	*sendp*	プラス方向の送信バッファ
in	*nIDm*	マイナス方向の隣接ランク番号
in	*nIDp*	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

6.11.4.22 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::packZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndComm_CART.h の 588 行で定義されています。

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

6.11.4.23 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::packZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndCommEx_CART.h の 595 行で定義されています。

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

6.11.4.24 template<class T > cpm_ErrorCode cpm_ParaManagerCART::packZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* T * *sendm,* T * *sendp,* int *nIDm,* int *nIDp*) [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
out	*sendm*	マイナス方向の送信バッファ
out	*sendp*	プラス方向の送信バッファ
in	*nIDm*	マイナス方向の隣接ランク番号
in	*nIDp*	プラス方向の隣接ランク番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

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

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

周期境界袖通信 (Scalar4D 版)

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

引数

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_ParaManager_BndComm_CART.h の 318 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(), cpm_ParaManager::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_ParaManager::Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::PeriodicCommS4D().

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

周期境界袖通信 (Scalar4DEx 版)

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

引数

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_ParaManager_BndCommEx_CART.h の 324 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(), cpm_ParaManager::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_ParaManager::Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::PeriodicCommS4DEx().

6.11.4.27 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::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_CART.h の 648 行で定義されています。

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

6.11.4.28 template<class T > cpm_ErrorCode cpm_ParaManagerCART::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().

6.11.4.29 cpm_ErrorCode cpm_ParaManagerCART::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_ParaManagerを再定義しています。

cpm_ParaManagerCART.cpp の 590 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_ParaManager::GetLocalVoxelSize(), m_bndCommInfoMap, S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_maxN, S_BNDCOMM_BUFFER::m_maxVC, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, と REAL_BUF_TYPE.

参照元 VoxelInit().

6.11.4.30 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::unpackX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndComm_CART.h の 498 行で定義されています。

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

6.11.4.31 template<class T > cpm_ErrorCode cpm_ParaManagerCART::unpackX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* 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		
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	*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().

6.11.4.32 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::unpackXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndCommEx_CART.h の 505 行で定義されています。

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

6.11.4.33 template<class T > cpm_ErrorCode cpm_ParaManagerCART::unpackXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* 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	*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().

6.11.4.34 template<class T > cpm_ErrorCode cpm_ParaManagerCART::unpackY (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, 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	*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().

6.11.4.35 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::unpackY (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, T * *recvm*, T * *recvp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndComm_CART.h の 558 行で定義されています。

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

6.11.4.36 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::unpackYEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, T * *recvm*, T * *recvp*, int *nIDm*, int *nIDp*)

cpm_ParaManager_BndCommEx_CART.h の 565 行で定義されています。

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

6.11.4.37 template<class T > cpm_ErrorCode cpm_ParaManagerCART::unpackYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* 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	*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().

6.11.4.38 template<class T > cpm_ErrorCode cpm_ParaManagerCART::unpackZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* 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	*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	*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().

6.11.4.39 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::unpackZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndComm_CART.h の 618 行で定義されています。

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

6.11.4.40 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::unpackZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* T * *recvm,* T * *recvp,* int *nIDm,* int *nIDp*)

cpm_ParaManager_BndCommEx_CART.h の 625 行で定義されています。

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

6.11.4.41 template<class T > cpm_ErrorCode cpm_ParaManagerCART::unpackZEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, 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	*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	*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().

6.11.4.42 cpm_ErrorCode cpm_ParaManagerCART::VoxelInit (cpm_GlobalDomainInfo * *domainInfo*, size_t *maxVC* = 1, size_t *maxN* = 3, int *procGrpNo* = 0) [virtual]

領域分割

- 既に作成済みの領域分割情報を用いた領域分割処理

  引数

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

  戻り値

  終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManagerを再定義しています。

cpm_ParaManagerCART.cpp の 52 行で定義されています。

参照先 cpm_ParaManager::Abort(), cpm_GlobalDomainInfo::CheckData(), CPM_ERROR_ALREADY_VOXEL-INIIT, CPM_ERROR_INSERT_VOXELMAP, CPM_ERROR_INVALID_PTR, CPM_ERROR_MISMATCH_NP_-SUBDOMAIN, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, cpm_ParaManager::GetMPI_Comm(), cpm_GlobalDomainInfo::GetSubdomainNum(), cpm_VoxelInfoCART::Init(), cpm_Base::IsCommNull(), cpm_Para-Manager::m_procGrpList, cpm_ParaManager::m_voxelInfoMap, と SetBndCommBuffer().

参照元 VoxelInit(), と VoxelInit_Subdomain().

6.11.4.43 cpm_ErrorCode cpm_ParaManagerCART::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

- 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を再定義しています。

cpm_ParaManagerCART.cpp の 128 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_ParaManager::GetNumRank(), cpm_GlobalDomainInfo::SetDivNum(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), と VoxelInit().

6.11.4.44 cpm_ErrorCode cpm_ParaManagerCART::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

- 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を再定義しています。

cpm_ParaManagerCART.cpp の 186 行で定義されています。

参照先 VoxelInit().

6.11.4.45 cpm_ErrorCode cpm_ParaManagerCART::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)

- 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を再定義しています。

cpm_ParaManagerCART.cpp の 198 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_ParaManager::GetNumRank(), cpm_GlobalDomainInfo::ReadActiveSubdomainFile(), cpm_GlobalDomainInfo::SetDivNum(), cpm_DomainInfo::SetOrigin(), cpm_DomainInfo::SetPitch(), cpm_DomainInfo::SetRegion(), cpm_DomainInfo::SetVoxNum(), と VoxelInit().

参照元 VoxelInit_Subdomain().

6.11.4.46 cpm_ErrorCode cpm_ParaManagerCART::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)

- 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を再定義しています。

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

参照先 VoxelInit_Subdomain().

6.11.4.47 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerCART::wait_BndCommS4D (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, MPI_Request *req[12]*, int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar4D 版)

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

引数

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
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_CART.h の 240 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(), cpm_ParaManager::GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, unpackX(), unpackY(), unpackZ(), cpm_ParaManager::WaitAll(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::wait_BndCommS4D().

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

非同期版袖通信の wait、展開 (Scalar4DEx 版)

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

引数

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
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_CART.h の 246 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(), cpm_ParaManager::GetNeighborRankID(), S_BNDCOMM_BUFFER::m_bufX, S_BNDCOMM_BUFFER::m_bufY, S_BNDCOMM_BUFFER::m_bufZ, S_BNDCOMM_BUFFER::m_nwX, S_BNDCOMM_BUFFER::m_nwY, S_BNDCOMM_BUFFER::m_nwZ, unpackXEx(), unpackYEx(), unpackZEx(), cpm_ParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::wait_BndCommS4DEx().

6.11.5 フレンドと関連する関数

6.11.5.1 friend class cpm_ParaManager [friend]

cpm_ParaManagerCART.h の 74 行で定義されています。

6.11.6 変数

6.11.6.1 BndCommInfoMap cpm_ParaManagerCART::m_bndCommInfoMap [protected]

プロセスグループ毎の袖通信バッファ情報

cpm_ParaManagerCART.h の 711 行で定義されています。

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

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_ParaManagerCART.h
- cpm_ParaManagerCART.cpp
- cpm_ParaManager_BndComm_CART.h
- cpm_ParaManager_BndCommEx_CART.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 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 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 BndCommS4D_nowait (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[48], int procGrpNo=0)
- template<class T >
  CPM_INLINE cpm_ErrorCode wait_BndCommS4D (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, MPI_Request req[48], 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)
- 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)
- 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)
- 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)
- 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)
- template<class T >
  CPM_INLINE cpm_ErrorCode recv_LMR (const int nID[2][4], int nFace[2], int levelDiff[2], size_t nw[2], T *recvm[4], MPI_Request *reqm, T *recvp[4], MPI_Request *reqp, int procGrpNo)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)

- template<class T >
  CPM_INLINE cpm_ErrorCode packMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_-comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_-comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_-comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_-comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode packMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode packPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_-comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  CPM_INLINE cpm_ErrorCode unpackPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_-comm, const int nID, int faceNo, int levelDiff, T *recvbuf)

Static Public メソッド

- static int GetNumLeaf (std::string treeFile)

Protected 型

- typedef std::map< int,
  S_BNDCOMM_BUFFER_LMR * > BndCommInfoMapLMR

Protected メソッド

- cpm_ParaManagerLMR ()
- virtual ~cpm_ParaManagerLMR ()
- CPM_INLINE S_BNDCOMM_BUFFER_LMR * GetBndCommBuffer (int procGrpNo=0)
- template<class T >
  cpm_ErrorCode recv_LMR (const int nID[2][4], int nFace[2], int levelDiff[2], size_t nw[2], T *recvm[4], MPI_-Request *reqm, T *recvp[4], MPI_Request *reqp, int procGrpNo=0)
- template<class T >
  cpm_ErrorCode packMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode packPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode unpackMX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode unpackPX (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode packMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode packPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode unpackMY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode unpackPY (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode packMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode packPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode unpackMZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode unpackPZ (T *array, int imax, int jmax, int kmax, int nmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode packMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode packPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode unpackMXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode unpackPXEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)

- template<class T >
  cpm_ErrorCode packMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode packPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode unpackMYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode unpackPYEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode packMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode packPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *sendbuf, size_t nw)
- template<class T >
  cpm_ErrorCode unpackMZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)
- template<class T >
  cpm_ErrorCode unpackPZEx (T *array, int nmax, int imax, int jmax, int kmax, int vc, int vc_comm, const int nID, int faceNo, int levelDiff, T *recvbuf)

Protected 変数

- BndCommInfoMapLMR m_bndCommInfoMap

フレンド

- class cpm_ParaManager

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 83 行で定義されています。

6.12.2 型定義

6.12.2.1 typedef std::map<int, S_BNDCOMM_BUFFER_LMR*> cpm_ParaManagerLMR::BndCommInfoMapLMR [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_ParaManagerLMR.h の 281 行で定義されています。

6.12.3 コンストラクタとデストラクタ

6.12.3.1 cpm_ParaManagerLMR::cpm_ParaManagerLMR ( ) [protected]

コンストラクタ

cpm_ParaManagerLMR.cpp の 24 行で定義されています。

参照先 CPM_DOMAIN_LMR, m_bndCommInfoMap, と cpm_ParaManager::m_domainType.

6.12.3.2 cpm_ParaManagerLMR::~cpm_ParaManagerLMR ( ) [protected], [virtual]

デストラクタ

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

参照先 m_bndCommInfoMap.

6.12.4 関数

6.12.4.1 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) 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_ParaManager_BndComm_LMR.h の 47 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetNeighborRankList(), cpm_ParaManager::Isend(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, S_BNDCOMM_BUFFER_LMR::m_bufSend, S_BNDCOMM_BUFFER_LMR::m_nrecv, S_BNDCOMM_BUFFER_LMR::m_nsend, cpm_ParaManager::m_rankNo, packMX(), packMY(), packMZ(), packPX(), packPY(), packPZ(), recv_LMR(), stmpd_printf, unpackMX(), unpackMY(), unpackMZ(), unpackPX(), unpackPY(), unpackPZ(), cpm_ParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4D().

6.12.4.2 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*, MPI_Request *req[48]*, int *procGrpNo* = 0)

非同期版袖通信 (Scalar4D 版)

- (imax,jmax,kmax,nmax) の形式の配列の非同期袖通信を行う

- wait と展開は行わず、request を返す
- wait、展開は wait_BndCommS4D をコールする

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm_LMR.h の 468 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetNeighborRankList(), cpm_ParaManager::Isend(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, S_BNDCOMM_BUFFER_LMR::m_bufSend, S_BNDCOMM_BUFFER_LMR::m_nrecv, S_BNDCOMM_BUFFER_LMR::m_nsend, cpm_ParaManager::m_rankNo, packMX(), packMY(), packMZ(), packPX(), packPY(), packPZ(), recv_LMR(), stmpd_printf, X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4D_nowait().

6.12.4.3 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) 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_ParaManager_BndCommEx_LMR.h の 53 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetNeighborRankList(), cpm_ParaManager::Isend(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, S_BNDCOMM_BUFFER_LMR::m_bufSend, S_BNDCOMM_BUFFER_LMR::m_nrecv, S_BNDCOMM_BUFFER_LMR::m_nsend, cpm_ParaManager::m_rankNo, packMXEx(), packMYEx(), packMZEx(), packPXEx(), packPYEx(), packPZEx(), recv_LMR(), stmpd_printf, unpackMXEx(), unpackMYEx(), unpackMZEx(),

unpackPXEx(), unpackPYEx(), unpackPZEx(), cpm_ParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4DEx().

6.12.4.4 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,* MPI_Request *req[48],* int *procGrpNo* = 0)

非同期版袖通信 (Scalar4DEx 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx_LMR.h の 474 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetNeighborRankList(), cpm_ParaManager::Isend(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, S_BNDCOMM_BUFFER_LMR::m_bufSend, S_BNDCOMM_BUFFER_LMR::m_nrecv, S_BNDCOMM_BUFFER_LMR::m_nsend, cpm_ParaManager::m_rankNo, packMXEx(), packMYEx(), packMZEx(), packPXEx(), packPYEx(), packPZEx(), recv_LMR(), stmpd_printf, X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::BndCommS4DEx_nowait().

6.12.4.5 CPM_INLINE S_BNDCOMM_BUFFER_LMR* cpm_ParaManagerLMR::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_ParaManagerLMR.h の 295 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().

6.12.4.6 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_ParaManagerを再定義しています。

cpm_ParaManagerLMR.cpp の 256 行で定義されています。

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

6.12.4.7 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 105 行で定義されています。

参照先 cpm_VoxelInfoLMR::GetNumLeaf().

6.12.4.8 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *sendbuf*, size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.9 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndComm_LMR.h の 1585 行で定義されています。

参照先 _IDX_S4D, _IDXFX, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.10 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.11 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndCommEx_LMR.h の 1558 行で定義されています。

参照先 _IDX_S4DEX, _IDXFX, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.12 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.13 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndComm_LMR.h の 1939 行で定義されています。

参照先 _IDX_S4D, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.14 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.15 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndCommEx_LMR.h の 1912 行で定義されています。

参照先 _IDX_S4DEX, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.16 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.17 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *sendbuf*, size_t *nw*)

cpm_ParaManager_BndComm_LMR.h の 2293 行で定義されています。

参照先 _IDX_S4D, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.18 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packMZEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *sendbuf*, size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.19 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *sendbuf*, size_t *nw*)

cpm_ParaManager_BndCommEx_LMR.h の 2266 行で定義されています。

参照先 _IDX_S4DEX, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.20 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPX (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.21 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndComm_LMR.h の 1656 行で定義されています。

参照先 _IDX_S4D, _IDXFX, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.22 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPXEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.23 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndCommEx_LMR.h の 1629 行で定義されています。

参照先 _IDX_S4DEX, _IDXFX, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.24 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.25 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndComm_LMR.h の 2010 行で定義されています。

参照先 _IDX_S4D, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.26 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.27 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndCommEx_LMR.h の 1983 行で定義されています。

参照先 _IDX_S4DEX, _IDXFY, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.28 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.29 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndComm_LMR.h の 2364 行で定義されています。

参照先 _IDX_S4D, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.30 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::packPZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
out	*sendbuf*	送信バッファ
in	*nw*	送信バッファサイズ

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

6.12.4.31 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *sendbuf,* size_t *nw*)

cpm_ParaManager_BndCommEx_LMR.h の 2337 行で定義されています。

参照先 _IDX_S4DEX, _IDXFZ, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_SUCCESS, cpm_Base::IsRankNull(), cpm_ParaManager::m_rankNo, と stmpd_printf.

6.12.4.32 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) の形式の配列の周期境界方向の袖通信を行う

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	*dir*	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	*pm*	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	*procGrpNo*	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm_LMR.h の 1007 行で定義されています。

参照先 BOTH, CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetPeriodicRankList(), cpm_Base::getRankNull(), cpm_ParaManager::Isend(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, S_BNDCOMM_BUFFER_LMR::m_bufSend, S_BNDCOMM_BUFFER_LMR::m_nrecv, S_BNDCOMM_BUFFER_LMR::m_nsend, cpm_ParaManager::m_rankNo, MINUS2PLUS, packMX(), packMY(), packMZ(), packPX(), packPY(), packPZ(), PLUS2MINUS, recv_LMR(), stmpd_printf, unpackMX(), unpackMY(), unpackMZ(), unpackPX(), unpackPY(), unpackPZ(), cpm_ParaManager::Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::PeriodicCommS4D().

6.12.4.33 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) の形式の配列の周期境界方向の袖通信を行う

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	*dir*	通信する軸方向 (X_DIR or Y_DIR or Z_DIR)
in	*pm*	通信する正負方向 (PLUS2MINUS or MINUS2PLUS or BOTH)
in	*procGrpNo*	プロセスグループ番号

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx_LMR.h の 1013 行で定義されています。

参照先 BOTH, CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_BNDCOMM_BUFFERLENGTH, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetPeriodicRankList(), cpm_Base::getRankNull(), cpm_ParaManager::Isend(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, S_BNDCOMM_BUFFER_LMR::m_bufSend, S_BNDCOMM_BUFFER_LMR::m_nrecv, S_BNDCOMM_BUFFER_LMR::m_nsend, cpm_ParaManager::m_rankNo, MINUS2PLUS, packMXEx(), packMYEx(), packMZEx(), packPXEx(), packPYEx(), packPZEx(), PLUS2MINUS, recv_LMR(), stmpd_printf, unpackMXEx(), unpackMYEx(), unpackMZEx(), unpackPXEx(), unpackPYEx(), unpackPZEx(), cpm_ParaManager::Waitall(), X_DIR, X_MINUS, X_PLUS, Y_DIR, Y_MINUS, Y_PLUS, Z_DIR, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::PeriodicCommS4DEx().

6.12.4.34 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR (const int *nID[2][4],* int *nFace[2],* int *levelDiff[2],* size_t *nw[2],* T * *recvm[4],* MPI_Request * *reqm,* T * *recvp[4],* MPI_Request * *reqp,* int *procGrpNo* = 0) [protected]

1 方向 (プラス、マイナス) の非同期受信処理

引数

in	*nID*	隣接ランクリスト ([0]:マイナス側、[1]:プラス側)
in	*nFace*	隣接ランク数 ([0]:マイナス側、[1]:プラス側)
in	*levelDiff*	隣接領域とのレベル差 ([0]:マイナス側、[1]:プラス側)
in	*nw*	1 面あたりの受信サイズ ([0]:マイナス側、[1]:プラス側)
in	*recvm*	マイナス方向の受信バッファ
out	*reqm*	マイナス方向のMPI_Request 配列のポインタ (サイズ 4)
in	*recvp*	プラス方向の受信バッファ
out	*reqp*	プラス方向のMPI_Request 配列のポインタ (サイズ 4)
in	*procGrpNo*	プロセスグループ番号

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

6.12.4.35 template<class T > CPM_INLINE cpm_ErrorCode cpm_ParaManagerLMR::recv_LMR (const int *nID[2][4],* int *nFace[2],* int *levelDiff[2],* size_t *nw[2],* T * *recvm[4],* MPI_Request * *reqm,* T * *recvp[4],* MPI_Request * *reqp,* int *procGrpNo*)

cpm_ParaManager_BndComm_LMR.h の 1552 行で定義されています。

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

6.12.4.36 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

- 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を再定義しています。

cpm_ParaManagerLMR.cpp の 141 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_ParaManager::GetLocalVoxelSize(), cpm_ParaManager::GetNeighborLevelDiff(), m_bndCommInfoMap, S_BNDCOMM_BUFFER_LMR::m_bufRecv, S_BNDCOMM_BUFFER_LMR::m_bufSend, S_BNDCOMM_BUFFER_LMR::m_maxN, S_BNDCOMM_BUFFER_LMR::m_maxVC, S_BNDCOMM_BUFFER_LMR::m_nface, S_BNDCOMM_BUFFER_LMR::m_nrecv, S_BNDCOMM_BUFFER_LMR::m_nsend, REAL_BUF_TYPE, X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 VoxelInit_LMR().

6.12.4.37 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.38 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndComm_LMR.h の 1727 行で定義されています。

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

6.12.4.39 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMXEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.40 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndCommEx_LMR.h の 1700 行で定義されています。

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

6.12.4.41 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMY (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.42 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndComm_LMR.h の 2081 行で定義されています。

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

6.12.4.43 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMYEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.44 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndCommEx_LMR.h の 2054 行で定義されています。

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

6.12.4.45 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMZ (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.46 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndComm_LMR.h の 2435 行で定義されています。

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

6.12.4.47 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackMZEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.48 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndCommEx_LMR.h の 2408 行で定義されています。

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

6.12.4.49 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPX (T * *array*, int *imax*, int *jmax*, int *kmax*, int *nmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.50 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndComm_LMR.h の 1832 行で定義されています。

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

6.12.4.51 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPXEx (T * *array*, int *nmax*, int *imax*, int *jmax*, int *kmax*, int *vc*, int *vc_comm*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.52 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*)

cpm_ParaManager_BndCommEx_LMR.h の 1805 行で定義されています。

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

6.12.4.53 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPY (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.54 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*)

cpm_ParaManager_BndComm_LMR.h の 2186 行で定義されています。

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

6.12.4.55 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPYEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.56 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*)

cpm_ParaManager_BndCommEx_LMR.h の 2159 行で定義されています。

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

6.12.4.57 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPZ (T * *array,* int *imax,* int *jmax,* int *kmax,* int *nmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.58 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,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*)

cpm_ParaManager_BndComm_LMR.h の 2540 行で定義されています。

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

6.12.4.59 template<class T > cpm_ErrorCode cpm_ParaManagerLMR::unpackPZEx (T * *array,* int *nmax,* int *imax,* int *jmax,* int *kmax,* int *vc,* int *vc_comm,* const int *nID,* int *faceNo,* int *levelDiff,* T * *recvbuf*) [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	*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	*nID*	隣接ランク番号
in	*faceNo*	面番号 (0/1/2/3)
in	*levelDiff*	隣接領域とのレベル差
in	*recvbuf*	受信バッファ

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

6.12.4.60 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*, const int *nID*, int *faceNo*, int *levelDiff*, T * *recvbuf*)

cpm_ParaManager_BndCommEx_LMR.h の 2513 行で定義されています。

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

6.12.4.61 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_ParaManagerを再定義しています。

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

参照先 cpm_ParaManager::Abort(), CPM_ERROR_ALREADY_VOXELINIIT, CPM_ERROR_INSERT_VOXELMAP, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, cpm_ParaManager::GetMPI_Comm(), cpm_VoxelInfoLMR::Init(), cpm_Base::IsCommNull(), cpm_ParaManager::m_procGrpList, cpm_ParaManager::m_voxelInfoMap, と SetBndCommBuffer().

6.12.4.62 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*, MPI_Request *req[48]*, int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar4D 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndComm_LMR.h の 790 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetNeighborRankList(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, unpackMX(), unpackMY(), unpackMZ(), unpackPX(), unpackPY(), unpackPZ(), cpm_ParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::wait_BndCommS4D().

6.12.4.63 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*, MPI_Request *req*[48], int *procGrpNo* = 0)

非同期版袖通信の wait、展開 (Scalar4DEx 版)

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

引数

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

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_ParaManager_BndCommEx_LMR.h の 796 行で定義されています。

参照先 CPM_ERROR_BNDCOMM_BUFFER, CPM_ERROR_INVALID_PTR, CPM_SUCCESS, GetBndCommBuffer(), cpm_ParaManager::GetNeighborLevelDiff(), cpm_ParaManager::GetNeighborRankList(), cpm_Base::IsRankNull(), S_BNDCOMM_BUFFER_LMR::m_bufRecv, unpackMXEx(), unpackMYEx(), unpackMZEx(), unpackPXEx(), unpackPYEx(), unpackPZEx(), cpm_ParaManager::Waitall(), X_MINUS, X_PLUS, Y_MINUS, Y_PLUS, Z_MINUS, と Z_PLUS.

参照元 cpm_ParaManager::wait_BndCommS4DEx().

6.12.5 フレンドと関連する関数

6.12.5.1 friend class cpm_ParaManager [friend]

cpm_ParaManagerLMR.h の 85 行で定義されています。

6.12.6 変数

6.12.6.1 BndCommInfoMapLMR cpm_ParaManagerLMR::m_bndCommInfoMap [protected]

プロセスグループ毎の袖通信バッファ情報

cpm_ParaManagerLMR.h の 750 行で定義されています。

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

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_ParaManager_BndComm_LMR.h
- cpm_ParaManager_BndCommEx_LMR.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 26 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 22 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 31 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 38 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 57 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 92 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 127 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 27 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 23 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 30 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 37 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 87 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 49 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	*dlnfo*	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 249 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 56 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 23 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 30 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 37 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 156 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 83 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 236 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 49 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

Protected メソッド

- cpm_VoxelInfo ()
- virtual ~cpm_VoxelInfo ()
- const int * GetDivNum () const
- const int * GetDivPos () const
- const double * GetPitch () const
- const double * GetGlobalPitch () const
- const int * GetGlobalVoxelSize () const
- const double * GetGlobalOrigin () const
- const double * GetGlobalRegion () const
- const int * GetLocalVoxelSize () const
- const double * GetLocalOrigin () const
- const double * GetLocalRegion () const
- const int * GetVoxelHeadIndex () const
- const int * GetVoxelTailIndex () const
- const int * GetNeighborRankID () const
- const int * GetPeriodicRankID () 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

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
- 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_ParaManager
- 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 27 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 ( ) [protected]

コンストラクタ

cpm_VoxelInfo.cpp の 22 行で定義されています。

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

6.16.2.2 cpm_VoxelInfo::~cpm_VoxelInfo ( ) [protected], [virtual]

デストラクタ

cpm_VoxelInfo.cpp の 43 行で定義されています。

6.16.3 関数

6.16.3.1 const int * cpm_VoxelInfo::GetDivNum ( ) const [protected]

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 50 行で定義されています。

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

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

6.16.3.2 const int * cpm_VoxelInfo::GetDivPos ( ) const [protected]

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 58 行で定義されています。

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

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

6.16.3.3 const double * cpm_VoxelInfo::GetGlobalOrigin ( ) const [protected]

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 90 行で定義されています。

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

参照元 cpm_ParaManager::GetGlobalOrigin().

6.16.3.4 const double * cpm_VoxelInfo::GetGlobalPitch ( ) const [protected]

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 74 行で定義されています。

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

6.16.3.5 const double * cpm_VoxelInfo::GetGlobalRegion ( ) const [protected]

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 98 行で定義されています。

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

参照元 cpm_ParaManager::GetGlobalRegion().

6.16.3.6 const int * cpm_VoxelInfo::GetGlobalVoxelSize ( ) const [protected]

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 82 行で定義されています。

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

参照元 cpm_ParaManager::GetGlobalVoxelSize().

6.16.3.7 const double * cpm_VoxelInfo::GetLocalOrigin ( ) const [protected]

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 114 行で定義されています。
参照先 cpm_DomainInfo::GetOrigin(), と m_localDomainInfo.
参照元 cpm_ParaManager::GetLocalOrigin().

6.16.3.8 const double * cpm_VoxelInfo::GetLocalRegion ( ) const [protected]

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 122 行で定義されています。
参照先 cpm_DomainInfo::GetRegion(), と m_localDomainInfo.
参照元 cpm_ParaManager::GetLocalRegion().

6.16.3.9 const int * cpm_VoxelInfo::GetLocalVoxelSize ( ) const [protected]

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 106 行で定義されています。
参照先 cpm_DomainInfo::GetVoxNum(), と m_localDomainInfo.
参照元 cpm_ParaManager::GetLocalVoxelSize().

6.16.3.10 int cpm_VoxelInfo::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で再定義されています。
cpm_VoxelInfo.cpp の 180 行で定義されています。
参照元 cpm_ParaManager::GetNeighborLevelDiff().

6.16.3.11 const int * cpm_VoxelInfo::GetNeighborRankID ( ) const [protected]

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 146 行で定義されています。

参照先 m_neighborRankID.

参照元 cpm_ParaManager::GetNeighborRankID().

6.16.3.12 const int * cpm_VoxelInfo::GetNeighborRankList (cpm_FaceFlag *face*, int & *num*) const [protected], [virtual]

指定面における自ランク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

指定面における自ランクの隣接ランク番号整数配列のポインタ

cpm_VoxelInfoLMRで再定義されています。

cpm_VoxelInfo.cpp の 162 行で定義されています。

参照先 m_neighborRankID.

参照元 cpm_ParaManager::GetNeighborRankList().

6.16.3.13 const int * cpm_VoxelInfo::GetPeriodicRankID ( ) const [protected]

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 154 行で定義されています。

参照先 m_periodicRankID.

参照元 cpm_ParaManager::GetPeriodicRankID().

6.16.3.14 const int * cpm_VoxelInfo::GetPeriodicRankList (cpm_FaceFlag *face*, int & *num*) const [protected], [virtual]

指定面における自ランク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_VoxelInfoLMRで再定義されています。

cpm_VoxelInfo.cpp の 171 行で定義されています。

参照先 m_periodicRankID.

参照元 cpm_ParaManager::GetPeriodicRankList().

6.16.3.15 const double * cpm_VoxelInfo::GetPitch ( ) const [protected]

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 66 行で定義されています。

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

参照元 cpm_ParaManager::GetPitch().

6.16.3.16 const int * cpm_VoxelInfo::GetVoxelHeadIndex ( ) const [protected]

自ランクの始点VOXEL の全体空間でのインデクスを取得 LMR のときは最大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 130 行で定義されています。

参照先 m_voxelHeadIndex.

参照元 cpm_ParaManager::GetVoxelHeadIndex().

6.16.3.17 const int * cpm_VoxelInfo::GetVoxelTailIndex ( ) const [protected]

自ランクの終点VOXEL の全体空間でのインデクスを取得 LMR のときは最大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 138 行で定義されています。

参照先 m_voxelTailIndex.

参照元 cpm_ParaManager::GetVoxelTailIndex().

6.16.3.18 bool cpm_VoxelInfo::IsInnerBoundary (cpm_FaceFlag *face*) const [protected], [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 215 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().

6.16.3.19 bool cpm_VoxelInfo::IsOuterBoundary (cpm_FaceFlag *face*) const [protected], [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 188 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_ParaManager [friend]

cpm_VoxelInfo.h の 29 行で定義されています。

6.16.4.2 friend class cpm_ParaManagerCART [friend]

cpm_VoxelInfo.h の 30 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 179 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 171 行で定義されています。

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), cpm_VoxelInfoCART::CreateRankMap(), GetDivNum(), 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 174 行で定義されています。

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), GetDivPos(), 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 182 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), GetNeighborRankID(), GetNeighborRankList(), IsInnerBoundary(), IsOuterBoundary(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.16.5.5 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 180 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().

6.16.5.6 int cpm_VoxelInfo::m_periodicRankID[6] [protected]

周期境界の隣接ランク番号

cpm_VoxelInfo.h の 183 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), GetPeriodicRankID(), GetPeriodicRankList(), と cpm_VoxelInfoLMR::SetNeighborInfo().

6.16.5.7 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 181 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), cpm_VoxelInfoCART::Init(), と cpm_VoxelInfoLMR::Init().

6.16.5.8 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 175 行で定義されています。

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

6.16.5.9 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 176 行で定義されています。

参照元 cpm_VoxelInfo(), cpm_VoxelInfoCART::CreateLocalDomainInfo(), 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 26 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 22 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 30 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 251 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, 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 145 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 82 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
in	*maxVC*	最大の袖数 (袖通信用)
in	*maxN*	最大の成分数 (袖通信用)

  戻り値

  終了コード (CPM_SUCCESS=正常終了)

cpm_VoxelInfoCART.cpp の 38 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_ParaManagerCART::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 28 行で定義されています。

6.17.4.2 friend class cpm_ParaManagerCART [friend]

cpm_VoxelInfoCART.h の 29 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 85 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

Protected メソッド

- cpm_VoxelInfoLMR ()
- virtual ~cpm_VoxelInfoLMR ()
- cpm_ErrorCode Init (MPI_Comm comm, std::string treeFile)
- void SetGlobaliDomainInfo (S_OCT_DOMAIN_INFO &dInfo)
- void SetLocalDomainInfo (S_OCT_DOMAIN_INFO &dInfo)
- void SetNeighborInfo ()
- 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 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 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
- const NeighborInfo * m_neighborInfo
  BCMOctree **から生成した隣接情報**
- 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]
  隣接ランクとのレベル差 *(-1/0/1)*

フレンド

- class cpm_ParaManager
- 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 29 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 23 行で定義されています。

参照先 cpm_Base::getRankNull(), m_neighborInfo, m_neighborLevelDiff, m_neighborRankID_LMR, m_octree, と m_periodicRankID_LMR.

6.18.2.2 cpm_VoxelInfoLMR::~cpm_VoxelInfoLMR ( ) [protected], [virtual]

デストラクタ

cpm_VoxelInfoLMR.cpp の 41 行で定義されています。

参照先 m_octree.

6.18.3 関数

6.18.3.1 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_VoxelInfoを再定義しています。

cpm_VoxelInfoLMR.cpp の 479 行で定義されています。

参照先 m_neighborLevelDiff.

6.18.3.2 const int * cpm_VoxelInfoLMR::GetNeighborRankList (cpm_FaceFlag *face,* int & *num*) const [protected], [virtual]

指定面における自ランク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

指定面における自ランクの隣接ランク番号整数配列のポインタ

cpm_VoxelInfoを再定義しています。

cpm_VoxelInfoLMR.cpp の 453 行で定義されています。

参照先 cpm_Base::IsRankNull(), と m_neighborRankID_LMR.

参照元 IsInnerBoundary(), と IsOuterBoundary().

6.18.3.3 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 429 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().

6.18.3.4 const int * cpm_VoxelInfoLMR::GetPeriodicRankList (cpm_FaceFlag *face*, int & *num*) const [protected], [virtual]

指定面における自ランク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を再定義しています。

cpm_VoxelInfoLMR.cpp の 466 行で定義されています。

参照先 cpm_Base::IsRankNull(), と m_periodicRankID_LMR.

6.18.3.5 cpm_ErrorCode cpm_VoxelInfoLMR::Init (MPI_Comm *comm*, std::string *treeFile*) [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	*tpFile*	領域情報ファイル

戻り値

終了コード (CPM_SUCCESS=正常終了)

cpm_VoxelInfoLMR.cpp の 49 行で定義されています。

参照先 CPM_ERROR_LMR_INVALID_OCTFILE, CPM_ERROR_LMR_MISMATCH_NP_NUMLEAF, CPM_ERROR_MPI_INVALID_COMM, CPM_SUCCESS, Node::getBlockID(), BCMOctree::getLeafNodeArray(), BCMOctree::getOrigin(), Node::getPedigree(), cpm_Base::IsCommNull(), LoadOctreeFile(), cpm_VoxelInfo::m_comm, 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().

6.18.3.6 bool cpm_VoxelInfoLMR::IsInnerBoundary (cpm_FaceFlag *face*) const [protected], [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_VoxelInfoを再定義しています。

cpm_VoxelInfoLMR.cpp の 516 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.

6.18.3.7 bool cpm_VoxelInfoLMR::IsOuterBoundary (cpm_FaceFlag *face*) const [protected], [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_VoxelInfoを再定義しています。

cpm_VoxelInfoLMR.cpp の 487 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.

6.18.3.8 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 152 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().

6.18.3.9 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 193 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().

6.18.3.10 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 216 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.

6.18.3.11 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 259 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().

6.18.3.12 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 288 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().

6.18.3.13 void cpm_VoxelInfoLMR::SetNeighborInfo ( ) [protected]

隣接情報の取得

cpm_VoxelInfoLMR.cpp の 358 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_neighborLevelDiff, cpm_VoxelInfo::m_neighborRankID, m_neighborRankID_LMR, m_node, cpm_VoxelInfo::m_nRank, m_octHeader, m_octree, 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_ParaManager [friend]

cpm_VoxelInfoLMR.h の 31 行で定義されています。

6.18.4.2 friend class cpm_ParaManagerLMR [friend]

cpm_VoxelInfoLMR.h の 32 行で定義されています。

6.18.5 変数

6.18.5.1 const NeighborInfo* cpm_VoxelInfoLMR::m_neighborInfo [protected]

BCMOctree から生成した隣接情報

cpm_VoxelInfoLMR.h の 162 行で定義されています。

参照元 cpm_VoxelInfoLMR().

6.18.5.2 int cpm_VoxelInfoLMR::m_neighborLevelDiff[6] [protected]

隣接ランクとのレベル差 (-1/0/1)

cpm_VoxelInfoLMR.h の 165 行で定義されています。

参照元 cpm_VoxelInfoLMR(), GetNeighborLevelDiff(), と SetNeighborInfo().

6.18.5.3 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 163 行で定義されています。

参照元 cpm_VoxelInfoLMR(), GetNeighborRankList(), と SetNeighborInfo().

6.18.5.4 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 159 行で定義されています。

参照元 Init(), SetGlobaliDomainInfo(), SetLocalDomainInfo(), と SetNeighborInfo().

6.18.5.5 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 157 行で定義されています。

参照元 Init(), SetGlobaliDomainInfo(), SetLocalDomainInfo(), と SetNeighborInfo().

6.18.5.6 BCMOctree* cpm_VoxelInfoLMR::m_octree [protected]

生成された木情報

cpm_VoxelInfoLMR.h の 158 行で定義されています。

参照元 cpm_VoxelInfoLMR(), Init(), SetLocalDomainInfo(), SetNeighborInfo(), と ~cpm_VoxelInfoLMR().

6.18.5.7 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 164 行で定義されています。

参照元 cpm_VoxelInfoLMR(), 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 BCMOctree::makeNeighborInfo(), cpm_VoxelInfoLMR::SetGlobaliDomainInfo(), 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_ParaManagerCART.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_ParaManagerCART.h の 25 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_ParaManagerCART.h の 36 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_ParaManagerCART.h の 48 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_ParaManagerCART.h の 61 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_ParaManagerCART::GetBndCommBufferSize().

6.31.4 変数

6.31.4.1 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufX[4]

バッファ

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

参照元 cpm_ParaManagerCART::BndCommS4D(), cpm_ParaManagerCART::BndCommS4D_nowait(), cpm_ParaManagerCART::BndCommS4DEx(), cpm_ParaManagerCART::BndCommS4DEx_nowait(), cpm_ParaManagerCART::PeriodicCommS4D(), cpm_ParaManagerCART::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManagerCART::SetBndCommBuffer(), cpm_ParaManagerCART::wait_BndCommS4D(), cpm_ParaManagerCART::wait_BndCommS4DEx(), と ~S_BNDCOMM_BUFFER().

6.31.4.2 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufY[4]

バッファ

cpm_ParaManagerCART.h の 33 行で定義されています。

参照元 cpm_ParaManagerCART::BndCommS4D(), cpm_ParaManagerCART::BndCommS4D_nowait(), cpm_ParaManagerCART::BndCommS4DEx(), cpm_ParaManagerCART::BndCommS4DEx_nowait(), cpm_ParaManagerCART::PeriodicCommS4D(), cpm_ParaManagerCART::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManagerCART::SetBndCommBuffer(), cpm_ParaManagerCART::wait_BndCommS4D(), cpm_ParaManagerCART::wait_BndCommS4DEx(), と ~S_BNDCOMM_BUFFER().

6.31.4.3 REAL_BUF_TYPE* S_BNDCOMM_BUFFER::m_bufZ[4]

バッファ

cpm_ParaManagerCART.h の 34 行で定義されています。

参照元 cpm_ParaManagerCART::BndCommS4D(), cpm_ParaManagerCART::BndCommS4D_nowait(), cpm_ParaManagerCART::BndCommS4DEx(), cpm_ParaManagerCART::BndCommS4DEx_nowait(), cpm_ParaManagerCART::PeriodicCommS4D(), cpm_ParaManagerCART::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManagerCART::SetBndCommBuffer(), cpm_ParaManagerCART::wait_BndCommS4D(), cpm_ParaManagerCART::wait_BndCommS4DEx(), と ~S_BNDCOMM_BUFFER().

6.31.4.4 size_t S_BNDCOMM_BUFFER::m_maxN

最大成分数

cpm_ParaManagerCART.h の 28 行で定義されています。

参照元 S_BNDCOMM_BUFFER(), と cpm_ParaManagerCART::SetBndCommBuffer().

6.31.4.5 size_t S_BNDCOMM_BUFFER::m_maxVC

最大袖数

cpm_ParaManagerCART.h の 27 行で定義されています。

参照元 S_BNDCOMM_BUFFER(), と cpm_ParaManagerCART::SetBndCommBuffer().

6.31.4.6 size_t S_BNDCOMM_BUFFER::m_nwX

バッファサイズ

cpm_ParaManagerCART.h の 29 行で定義されています。

参照元 cpm_ParaManagerCART::BndCommS4D(), cpm_ParaManagerCART::BndCommS4D_nowait(), cpm_ParaManagerCART::BndCommS4DEx(), cpm_ParaManagerCART::BndCommS4DEx_nowait(), CalcBufferSize(), cpm_ParaManagerCART::PeriodicCommS4D(), cpm_ParaManagerCART::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManagerCART::SetBndCommBuffer(), cpm_ParaManagerCART::wait_BndCommS4D(), と cpm_ParaManagerCART::wait_BndCommS4DEx().

6.31.4.7 size_t S_BNDCOMM_BUFFER::m_nwY

バッファサイズ

cpm_ParaManagerCART.h の 30 行で定義されています。

参照元 cpm_ParaManagerCART::BndCommS4D(), cpm_ParaManagerCART::BndCommS4D_nowait(), cpm_ParaManagerCART::BndCommS4DEx(), cpm_ParaManagerCART::BndCommS4DEx_nowait(), CalcBufferSize(), cpm_ParaManagerCART::PeriodicCommS4D(), cpm_ParaManagerCART::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManagerCART::SetBndCommBuffer(), cpm_ParaManagerCART::wait_BndCommS4D(), と cpm_ParaManagerCART::wait_BndCommS4DEx().

6.31.4.8 size_t S_BNDCOMM_BUFFER::m_nwZ

バッファサイズ

cpm_ParaManagerCART.h の 31 行で定義されています。

参照元 cpm_ParaManagerCART::BndCommS4D(), cpm_ParaManagerCART::BndCommS4D_nowait(), cpm_ParaManagerCART::BndCommS4DEx(), cpm_ParaManagerCART::BndCommS4DEx_nowait(), CalcBufferSize(), cpm_ParaManagerCART::PeriodicCommS4D(), cpm_ParaManagerCART::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER(), cpm_ParaManagerCART::SetBndCommBuffer(), cpm_ParaManagerCART::wait_BndCommS4D(), と cpm_ParaManagerCART::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_ParaManagerCART.h

6.32 構造体 S_BNDCOMM_BUFFER_LMR

`#include <cpm_ParaManagerLMR.h>`

Public メソッド

- S_BNDCOMM_BUFFER_LMR ()
- ~S_BNDCOMM_BUFFER_LMR ()
- 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_nsend [6]
  送信バッファサイズ
- size_t m_nrecv [6]
  受信バッファサイズ
- size_t m_nface [6]
  通信相手の面数
- REAL_BUF_TYPE * m_bufSend [6][4]
  送信バッファ
- REAL_BUF_TYPE * m_bufRecv [6][4]
  受信バッファ

6.32.1 説明

袖通信バッファ情報

cpm_ParaManagerLMR.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_BNDCOMM_BUFFER_LMR::S_BNDCOMM_BUFFER_LMR ( ) [inline]

cpm_ParaManagerLMR.h の 35 行で定義されています。

参照先 m_bufRecv, m_bufSend, m_maxN, m_maxVC, m_nface, m_nrecv, と m_nsend.

6.32.2.2 S_BNDCOMM_BUFFER_LMR::∼S_BNDCOMM_BUFFER_LMR ( ) [inline]

cpm_ParaManagerLMR.h の 51 行で定義されています。

参照先 m_bufRecv, と m_bufSend.

6.32.3 関数

6.32.3.1 size_t S_BNDCOMM_BUFFER_LMR::CalcBufferSize ( ) [inline]

バッファサイズの取得

戻り値

バッファサイズ [Byte]

cpm_ParaManagerLMR.h の 66 行で定義されています。

参照先 m_nrecv, m_nsend, と REAL_BUF_TYPE.

参照元 cpm_ParaManagerLMR::GetBndCommBufferSize().

6.32.4 変数

6.32.4.1 REAL_BUF_TYPE∗ S_BNDCOMM_BUFFER_LMR::m_bufRecv[6][4]

受信バッファ

cpm_ParaManagerLMR.h の 33 行で定義されています。

参照元 cpm_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerLMR::BndCommS4DEx_nowait(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER_LMR(), cpm_ParaManagerLMR::SetBndCommBuffer(), cpm_ParaManagerLMR::wait_BndCommS4D(), cpm_ParaManagerLMR::wait_BndCommS4DEx(), と ∼S_BNDCOMM_BUFFER_LMR().

6.32.4.2 REAL_BUF_TYPE∗ S_BNDCOMM_BUFFER_LMR::m_bufSend[6][4]

送信バッファ

cpm_ParaManagerLMR.h の 32 行で定義されています。

参照元 cpm_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerLMR::BndCommS4DEx_nowait(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER_LMR(), cpm_ParaManagerLMR::SetBndCommBuffer(), と ~S_BNDCOMM_BUFFER_LMR().

6.32.4.3 size_t S_BNDCOMM_BUFFER_LMR::m_maxN

最大成分数

cpm_ParaManagerLMR.h の 28 行で定義されています。

参照元 S_BNDCOMM_BUFFER_LMR(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.32.4.4 size_t S_BNDCOMM_BUFFER_LMR::m_maxVC

最大袖数

cpm_ParaManagerLMR.h の 27 行で定義されています。

参照元 S_BNDCOMM_BUFFER_LMR(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.32.4.5 size_t S_BNDCOMM_BUFFER_LMR::m_nface[6]

通信相手の面数

cpm_ParaManagerLMR.h の 31 行で定義されています。

参照元 S_BNDCOMM_BUFFER_LMR(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.32.4.6 size_t S_BNDCOMM_BUFFER_LMR::m_nrecv[6]

受信バッファサイズ

cpm_ParaManagerLMR.h の 30 行で定義されています。

参照元 cpm_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerLMR::BndCommS4DEx_nowait(), CalcBufferSize(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER_LMR(), と cpm_ParaManagerLMR::SetBndCommBuffer().

6.32.4.7 size_t S_BNDCOMM_BUFFER_LMR::m_nsend[6]

送信バッファサイズ

cpm_ParaManagerLMR.h の 29 行で定義されています。

参照元 cpm_ParaManagerLMR::BndCommS4D(), cpm_ParaManagerLMR::BndCommS4D_nowait(), cpm_ParaManagerLMR::BndCommS4DEx(), cpm_ParaManagerLMR::BndCommS4DEx_nowait(), CalcBufferSize(), cpm_ParaManagerLMR::PeriodicCommS4D(), cpm_ParaManagerLMR::PeriodicCommS4DEx(), S_BNDCOMM_BUFFER_LMR(), と cpm_ParaManagerLMR::SetBndCommBuffer().

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_ParaManagerLMR.h

6.33 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

6.33.1 説明

領域情報ファイル構造体

cpm_TextParserDomainLMR.h の 26 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 S_OCT_DOMAIN_INFO::S_OCT_DOMAIN_INFO ( ) [inline]

コンストラクタ

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

参照先 octFile, origin, region, size, と unitLength.

6.33.3 関数

6.33.3.1 void S_OCT_DOMAIN_INFO::print ( ) [inline]

cpm_TextParserDomainLMR.h の 44 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().

6.33.4 変数

6.33.4.1 std::string S_OCT_DOMAIN_INFO::octFile

oct ファイル名

cpm_TextParserDomainLMR.h の 30 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().

6.33.4.2 double S_OCT_DOMAIN_INFO::origin[3]

原点座標

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().

6.33.4.3 double S_OCT_DOMAIN_INFO::region[3]

領域幅

cpm_TextParserDomainLMR.h の 29 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().

6.33.4.4 int S_OCT_DOMAIN_INFO::size[3]

1 リーフの格子数

cpm_TextParserDomainLMR.h の 31 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().

6.33.4.5 std::string S_OCT_DOMAIN_INFO::unitLength

長さ単位文字列

cpm_TextParserDomainLMR.h の 32 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

6.34 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 63 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 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.

6.34.2.2 template<typename T> Vec3class::Vec3< T >::Vec3 (T *_x*, T *_y*, T *_z*) [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.

6.34.2.3 template<typename T> Vec3class::Vec3< T >::Vec3 (const T *v[3]*) [inline]

Vec3.h の 70 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 71 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 73 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 201 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 183 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 179 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 185 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 193 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 79 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 78 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 171 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 146 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 154 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 117 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 127 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 138 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 107 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 142 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 163 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 112 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 150 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 158 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 122 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 132 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 167 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 85 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 95 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 80 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 81 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 175 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 176 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 177 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 66 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 66 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 66 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

  BCM **用マルチルート***OCtree* **クラス**.

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

作者

University of Tokyo

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 42 行で定義されています。

7.7 cpm_Define.h

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

cpm_Define.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

マクロ定義

- #define REAL_BUF_TYPE double
- #define _IDX_S3D(_I, _J, _K, _NI, _NJ, _NK, _VC)
- #define _IDX_S4D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC)
- #define _IDX_V3D(_I, _J, _K, _N, _NI, _NJ, _NK, _VC) (_IDX_S4D(_I,_J,_K,_N,_NI,_NJ,_NK,_VC))
- #define _IDX_S4DEX(_N, _I, _J, _K, _NN, _NI, _NJ, _NK, _VC)
- #define _IDX_V3DEX(_N, _I, _J, _K, _NI, _NJ, _NK, _VC) (_IDX_S4DEX(_N,_I,_J,_K,3,_NI,_NJ,_NK,_VC))
- #define stmpd_printf printf("%s (%d): ",__FILE__, __LINE__); printf

列挙型

- 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_VOXELINIT_LMR = 3100,
  CPM_ERROR_LMR_OPEN_OCTFILE = 3101, CPM_ERROR_LMR_INVALID_OCTFILE = 3102,
  CPM_ERROR_LMR_READ_OCT_HEADER = 3103, CPM_ERROR_LMR_READ_OCT_PEDIGREE = 3104,
  CPM_ERROR_LMR_MISMATCH_NP_NUMLEAF = 3105, CPM_ERROR_DOMAINTYPE_VOXELINIT = 3100, CPM_ERROR_DOMAINTYPE_SETBNDCOMMBUF = 3101, 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_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 }

7.7.1 説明

CPM の定義マクロ記述ヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_Define.h で定義されています。

7.7.2 マクロ定義

7.7.2.1 #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 49 行で定義されています。

参照元 cpm_VoxelInfoCART::CreateLocalDomainInfo(), cpm_VoxelInfoCART::CreateNeighborRankInfo(), cpm_VoxelInfoCART::CreateRankMap(), と cpm_ParaManagerCART::GetBndIndexExtGc().

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

値:

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

4 次元インデクス (i,j,k,n) -> 1 次元インデクス変換マクロ

引数

in	*_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 66 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_ParaManagerCART::packX(), cpm_ParaManagerCART::packY(), cpm_ParaManagerCART::packZ(), cpm_ParaManagerLMR::unpackMX(), cpm_ParaManagerLMR::unpackMY(), cpm_ParaManagerLMR::unpackMZ(), cpm_ParaManagerLMR::unpackPX(), cpm_ParaManagerLMR::unpackPY(), cpm_ParaManagerLMR::unpackPZ(), cpm_ParaManagerCART::unpackX(), cpm_ParaManagerCART::unpackY(), と cpm_ParaManagerCART::unpackZ().

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

値:

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

4 次元インデクス (n,i,j,k) -> 1 次元インデクス変換マクロ

引数

in	*_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 95 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_ParaManagerCART::packXEx(), cpm_ParaManagerCART::packYEx(), cpm_ParaManagerCART::packZEx(), cpm_ParaManagerLMR::unpackMXEx(), cpm_ParaManagerLMR::unpackMYEx(), cpm_ParaManagerLMR::unpackMZEx(), cpm_ParaManagerLMR::unpackPXEx(), cpm_ParaManagerLMR::unpackPYEx(), cpm_ParaManagerLMR::unpackPZEx(), cpm_ParaManagerCART::unpackXEx(), cpm_ParaManagerCART::unpackYEx(), と cpm_ParaManagerCART::unpackZEx().

7.7.2.4 #define _IDX_V3D(*_I, _J, _K, _N, _NI, _NJ, _NK, _VC*) (_IDX_S4D(_I,_J,_K,_N,_NI,_NJ,_NK,_VC))

3 次元インデクス (i,j,k,3) -> 1 次元インデクス変換マクロ

引数

in	*_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 81 行で定義されています。

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

3 次元インデクス (3,i,j,k) -> 1 次元インデクス変換マクロ

引数

in	*_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

cpm_Define.h の 109 行で定義されています。

7.7.2.6 #define REAL_BUF_TYPE double

袖通信バッファの型指定

- デフォルトでは、REAL_BUF_TYPE=double
- コンパイル時オプション-D_BUFSIZE_FLOAT_を付与することで REAL_BUF_TYPE=float になる
- コンパイル時オプション-D_BUFSIZE_LONG_DOUBLE_を付与することで REAL_BUF_TYPE=long double になる

cpm_Define.h の 35 行で定義されています。

参照元 S_BNDCOMM_BUFFER::CalcBufferSize(), S_BNDCOMM_BUFFER_LMR::CalcBufferSize(), cpm_ParaManagerLMR::SetBndCommBuffer(), と cpm_ParaManagerCART::SetBndCommBuffer().

7.7.2.7 #define stmpd_printf printf("%s (%d): ",__FILE__, __LINE__); printf

デバッグライト用

cpm_Define.h の 311 行で定義されています。

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

7.7.3 列挙型

7.7.3.1 enum CPM_Datatype

fortran 用のデータタイプ

列挙型の値

CPM_CHAR char

CPM_UNSIGNED_CHAR unsigned char

CPM_BYTE byte(not support)

CPM_SHORT short

CPM_UNSIGNED_SHORT unsigned short

CPM_INT int

CPM_UNSIGNED unsigned

CPM_LONG long

CPM_UNSIGNED_LONG unsigned long

CPM_FLOAT float

CPM_DOUBLE double

CPM_LONG_DOUBLE long double
CPM_REAL REAL_TYPE.

cpm_Define.h の 266 行で定義されています。

7.7.3.2 enum cpm_DirFlag

軸方向フラグ

列挙型の値

X_DIR X direction.
Y_DIR Y direction.
Z_DIR Z direction.

cpm_Define.h の 131 行で定義されています。

7.7.3.3 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 147 行で定義されています。

7.7.3.4 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 112 行で定義されています。

7.7.3.5 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_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_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_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_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 154 行で定義されています。

7.7.3.6 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 120 行で定義されています。

7.7.3.7 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 293 行で定義されています。

7.7.3.8 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 139 行で定義されています。

7.8 cpm_DomainInfo.cpp

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

cpm_DomainInfo.cpp のインクルード依存関係図

7.8.1 説明

DomainInfo クラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_DomainInfo.cpp で定義されています。

7.9 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

7.9.1 説明

領域情報クラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_DomainInfo.h で定義されています。

7.10 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)

7.10.1 説明

CPM エンディアンユーティリティヘッダーファイル

作者

University of Tokyo

日付

2013/04/02

cpm_EndianUtil.h で定義されています。

7.11 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

7.11.1 説明

汎用オブジェクトの管理クラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ObjList.h で定義されています。

7.11.2 型定義

7.11.2.1 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 28 行で定義されています。

7.12 cpm_ParaManager.cpp

```
#include "cpm_ParaManager.h"
#include "cpm_ParaManagerCART.h"
#include "cpm_ParaManagerLMR.h"
```

cpm_ParaManager.cpp のインクルード依存関係図

構成

- class C_PARAMANAGER

7.12.1 説明

パラレルマネージャクラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager.cpp で定義されています。

7.13 cpm_ParaManager.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 "cpm_ParaManagerCART.h"
#include "cpm_ParaManagerLMR.h"
#include "inline/cpm_ParaManager_inline.h"
#include "inline/cpm_ParaManager_BndComm.h"
#include "inline/cpm_ParaManager_BndCommEx.h"
```

cpm_ParaManager.h のインクルード依存関係図このグラフは、どのファイルから直接、間接的にインクルードされているかを示しています。

構成

- class cpm_ParaManager

型定義

- typedef std::map< int, cpm_VoxelInfo * > VoxelInfoMap

7.13.1 説明

パラレルマネージャクラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager.h で定義されています。

7.13.2 型定義

7.13.2.1 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 32 行で定義されています。

7.14 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

7.14.1 説明

パラレルマネージャクラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager_Alloc.cpp で定義されています。

7.15 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。

7.15.1 説明

パラレルマネージャクラスのインラインヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager_BndComm.h で定義されています。

7.16 cpm_ParaManager_BndComm_CART.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)

7.16.1 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

作者

University of Tokyo

日付

2015/03/27

cpm_ParaManager_BndComm_CART.h で定義されています。

7.16.2 マクロ定義

7.16.2.1 #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_CART.h の 22 行で定義されています。

参照元 cpm_ParaManagerCART::packX(), と cpm_ParaManagerCART::unpackX().

7.16.2.2 #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_CART.h の 29 行で定義されています。

参照元 cpm_ParaManagerCART::packY(), と cpm_ParaManagerCART::unpackY().

7.16.2.3 #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_CART.h の 36 行で定義されています。

参照元 cpm_ParaManagerCART::packZ(), と cpm_ParaManagerCART::unpackZ().

7.17 cpm_ParaManager_BndComm_LMR.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)

7.17.1 説明

LMR 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

作者

University of Tokyo

日付

2015/03/27

cpm_ParaManager_BndComm_LMR.h で定義されています。

7.17.2 マクロ定義

7.17.2.1 #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_LMR.h の 22 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().

7.17.2.2 #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_LMR.h の 29 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().

7.17.2.3 #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_LMR.h の 36 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().

7.18 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。

7.18.1 説明

パラレルマネージャクラスのインラインヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

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

7.19 cpm_ParaManager_BndCommEx_CART.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)

7.19.1 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

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager_BndCommEx_CART.h で定義されています。

7.19.2 マクロ定義

7.19.2.1 #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_CART.h の 22 行で定義されています。

参照元 cpm_ParaManagerCART::packXEx(), と cpm_ParaManagerCART::unpackXEx().

7.19.2.2 #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_CART.h の 31 行で定義されています。

参照元 cpm_ParaManagerCART::packYEx(), と cpm_ParaManagerCART::unpackYEx().

7.19.2.3 #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_CART.h の 40 行で定義されています。

参照元 cpm_ParaManagerCART::packZEx(), と cpm_ParaManagerCART::unpackZEx().

7.20 cpm_ParaManager_BndCommEx_LMR.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)

7.20.1 マクロ定義

7.20.1.1 #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_ParaManager_BndCommEx_LMR.h の 22 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().

7.20.1.2 #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_ParaManager_BndCommEx_LMR.h の 31 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().

7.20.1.3 #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_ParaManager_BndCommEx_LMR.h の 40 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().

7.21 cpm_ParaManager_frtIF.cpp

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

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

マクロ定義

- #define CPM_EXTERN extern "C"
- #define cpm_Initialize_ cpm_initialize_
- #define cpm_VoxelInit_ cpm_voxelinit_
- #define cpm_VoxelInit_nodiv_ cpm_voxelinit_nodiv_
- #define cpm_IsParallel_ cpm_isparallel_
- #define cpm_GetDivNum_ cpm_getdivnum_
- #define cpm_GetPitch_ cpm_getpitch_
- #define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_
- #define cpm_GetGlobalOrigin_ cpm_getglobalorigin_
- #define cpm_GetGlobalRegion_ cpm_getglobalregion_
- #define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_
- #define cpm_GetLocalOrigin_ cpm_getlocalorigin_
- #define cpm_GetLocalRegion_ cpm_getlocalregion_
- #define cpm_GetDivPos_ cpm_getdivpos_
- #define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_
- #define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_
- #define cpm_GetNeighborRankID_ cpm_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 *domainType, 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_VoxelInit_nodiv_ (int *vox, 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_GetGlobalOrigin_ (double *worg, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetGlobalRegion_ (double *wrgn, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalVoxelSize_ (int *lsz, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetLocalOrigin_ (double *lorg, int *procGrpNo, int *ierr)

- CPM_EXTERN void cpm_GetLocalRegion_ (double *lrgn, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetDivPos_ (int *pos, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetVoxelHeadIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetVoxelTailIndex_ (int *idx, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNeighborRankID_ (int *nID, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetPeriodicRankID_ (int *nID, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetMyRankID_ (int *id, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_GetNumRank_ (int *nrank, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Abort_ (int *errorcode)
- CPM_EXTERN void cpm_Barrier_ (int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Wait_ (int *reqNo, int *ierr)
- CPM_EXTERN void cpm_Waitall_ (int *count, int *reqlist, int *ierr)
- CPM_EXTERN void cpm_Bcast_ (void *buf, int *count, int *datatype, int *root, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Send_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Recv_ (void *buf, int *count, int *datatype, int *source, int *procGrpNo, int *ierr)
- CPM_EXTERN void cpm_Isend_ (void *buf, int *count, int *datatype, int *dest, int *procGrpNo, int *reqNo, int *ierr)
- CPM_EXTERN void cpm_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)

7.21.1 説明

パラレルマネージャクラスのFortran インターフェイスのソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager_frtIF.cpp で定義されています。

7.21.2 マクロ定義

7.21.2.1 #define cpm_Abort_ cpm_abort_

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

7.21.2.2 #define cpm_Allgather_ cpm_allgather_

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

7.21.2.3 #define cpm_Allgatherv_ cpm_allgatherv_

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

7.21.2.4 #define cpm_Allreduce_ cpm_allreduce_

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

7.21.2.5 #define cpm_Barrier_ cpm_barrier_

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

7.21.2.6 #define cpm_Bcast_ cpm_bcast_

cpm_ParaManager_frtIF.cpp の 52 行で定義されています。

7.21.2.7 #define cpm_BndCommS3D_ cpm_bndcomms3d_

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

7.21.2.8 #define cpm_BndCommS3D_nowait_ cpm_bndcomms3d_nowait_

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

7.21.2.9 #define cpm_BndCommS4D_ cpm_bndcomms4d_

cpm_ParaManager_frtIF.cpp の 65 行で定義されています。
参照元 cpm_BndCommS3D_(), と cpm_BndCommV3D_().

7.21.2.10 #define cpm_BndCommS4D_nowait_ cpm_bndcomms4d_nowait_

cpm_ParaManager_frtIF.cpp の 68 行で定義されています。
参照元 cpm_BndCommS3D_nowait_(), と cpm_BndCommV3D_nowait_().

7.21.2.11 #define cpm_BndCommS4DEx_ cpm_bndcomms4dex_

cpm_ParaManager_frtIF.cpp の 73 行で定義されています。
参照元 cpm_BndCommV3DEx_().

7.21.2.12 #define cpm_BndCommS4DEx_nowait_ cpm_bndcomms4dex_nowait_

cpm_ParaManager_frtIF.cpp の 75 行で定義されています。
参照元 cpm_BndCommV3DEx_nowait_().

7.21.2.13 #define cpm_BndCommV3D_ cpm_bndcommv3d_

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

7.21.2.14 #define cpm_BndCommV3D_nowait_ cpm_bndcommv3d_nowait_

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

7.21.2.15 #define cpm_BndCommV3DEx_ cpm_bndcommv3dex_

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

7.21.2.16 #define cpm_BndCommV3DEx_nowait_ cpm_bndcommv3dex_nowait_

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

7.21.2.17 #define CPM_EXTERN extern "C"

extern 宣言

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

7.21.2.18 #define cpm_Gather_ cpm_gather_

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

7.21.2.19 #define cpm_Gatherv_ cpm_gatherv_

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

7.21.2.20 #define cpm_GetDivNum_ cpm_getdivnum_

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

7.21.2.21 #define cpm_GetDivPos_ cpm_getdivpos_

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

7.21.2.22 #define cpm_GetGlobalOrigin_ cpm_getglobalorigin_

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

7.21.2.23 #define cpm_GetGlobalRegion_ cpm_getglobalregion_

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

7.21.2.24 #define cpm_GetGlobalVoxelSize_ cpm_getglobalvoxelsize_

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

7.21.2.25 #define cpm_GetLocalOrigin_ cpm_getlocalorigin_

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

7.21.2.26 #define cpm_GetLocalRegion_ cpm_getlocalregion_

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

7.21.2.27 #define cpm_GetLocalVoxelSize_ cpm_getlocalvoxelsize_

cpm_ParaManager_frtIF.cpp の 38 行で定義されています。

7.21.2.28 #define cpm_GetMyRankID_ cpm_getmyrankid_

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

7.21.2.29 #define cpm_GetNeighborRankID_ cpm_getneighborrankid_

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

7.21.2.30 #define cpm_GetNumRank_ cpm_getnumrank_

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

7.21.2.31 #define cpm_GetPeriodicRankID_ cpm_getperiodicrankid_

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

7.21.2.32 #define cpm_GetPitch_ cpm_getpitch_

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

7.21.2.33 #define cpm_GetVoxelHeadIndex_ cpm_getvoxelheadindex_

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

7.21.2.34 #define cpm_GetVoxelTailIndex_ cpm_getvoxeltailindex_

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

7.21.2.35 #define cpm_Initialize_ cpm_initialize_

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

7.21.2.36 #define cpm_Irecv_ cpm_irecv_

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

7.21.2.37 #define cpm_Isend_ cpm_isend_

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

7.21.2.38 #define cpm_IsParallel_ cpm_isparallel_

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

7.21.2.39 #define cpm_PeriodicCommS3D cpm_periodiccomms3d_

cpm_ParaManager_frtIF.cpp の 78 行で定義されています。

7.21.2.40 #define cpm_PeriodicCommS4D cpm_periodiccomms4d_

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

7.21.2.41 #define cpm_PeriodicCommS4DEx cpm_periodiccomms4dex_

cpm_ParaManager_frtIF.cpp の 82 行で定義されています。

7.21.2.42 #define cpm_PeriodicCommV3D cpm_periodiccommv3d_

cpm_ParaManager_frtIF.cpp の 79 行で定義されています。

7.21.2.43 #define cpm_PeriodicCommV3DEx cpm_periodiccommv3dex_

cpm_ParaManager_frtIF.cpp の 81 行で定義されています。

7.21.2.44 #define cpm_Recv_ cpm_recv_

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

7.21.2.45 #define cpm_Send_ cpm_send_

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

7.21.2.46 #define cpm_SetBndCommBuffer_ cpm_setbndcommbuffer_

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

7.21.2.47 #define cpm_VoxelInit_ cpm_voxelinit_

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

7.21.2.48 #define cpm_VoxelInit_nodiv_ cpm_voxelinit_nodiv_

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

7.21.2.49 #define cpm_Wait_ cpm_wait_

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

7.21.2.50 #define cpm_wait_BndCommS3D_ cpm_wait_bndcomms3d_

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

7.21.2.51 #define cpm_wait_BndCommS4D_ cpm_wait_bndcomms4d_

cpm_ParaManager_frtIF.cpp の 71 行で定義されています。
参照元 cpm_wait_BndCommS3D_(), と cpm_wait_BndCommV3D_().

7.21.2.52 #define cpm_wait_BndCommS4DEx_ cpm_wait_bndcomms4dex_

cpm_ParaManager_frtIF.cpp の 77 行で定義されています。
参照元 cpm_wait_BndCommV3DEx_().

7.21.2.53 #define cpm_wait_BndCommV3D_ cpm_wait_bndcommv3d_

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

7.21.2.54 #define cpm_wait_BndCommV3DEx_ cpm_wait_bndcommv3dex_

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

7.21.2.55 #define cpm_Waitall_ cpm_waitall_

cpm_ParaManager_frtIF.cpp の 51 行で定義されています。

7.21.3 関数

7.21.3.1 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 942 行で定義されています。

参照先 cpm_ParaManager::Abort(), と cpm_ParaManager::get_instance().

7.21.3.2 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 1359 行で定義されています。

参照先 cpm_ParaManager::Allgather(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.3 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 1461 行で定義されています。

参照先 cpm_ParaManager::Allgatherv(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.4 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 1258 行で定義されています。

参照先 cpm_ParaManager::Allreduce(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_MPI_INVALID_OPERATOR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::GetMPI_Op().

7.21.3.5 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 970 行で定義されています。

参照先 cpm_ParaManager::Barrier(), CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, と cpm_ParaManager::get_instance().

7.21.3.6 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 1059 行で定義されています。

参照先 cpm_ParaManager::Bcast(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.7 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 1589 行で定義されています。

参照先 cpm_ParaManager::BndCommS3D(), cpm_BndCommS4D_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.8 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 1731 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().

7.21.3.9 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 1544 行で定義されています。

参照先 cpm_ParaManager::BndCommS4D(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.10 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 1691 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().

7.21.3.11 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 1954 行で定義されています。

参照先 cpm_ParaManager::BndCommS4DEx(), CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.12 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 2051 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().

7.21.3.13 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 1639 行で定義されています。

参照先 cpm_ParaManager::BndCommV3D(), cpm_BndCommS4D_, CPM_ERROR_INVALID_PTR, CPM_ERRO-R_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_Para-Manager::GetMPI_Datatype().

7.21.3.14 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 1777 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().

7.21.3.15 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 1999 行で定義されています。

参照先 cpm_ParaManager::BndCommV3DEx(), cpm_BndCommS4DEx_, CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.16 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 2091 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().

7.21.3.17 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 1309 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::Gather(), cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.18 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 1410 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::Gatherv(), cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMPI_Datatype().

7.21.3.19 CPM_EXTERN void cpm_GetDivNum_ (int * *div,* int * *procGrpNo,* int * *ierr*)

領域分割数を取得

- GetDivNum のFortran インターフェイス関数

  引数

out	*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 314 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().

7.21.3.20 CPM_EXTERN void cpm_GetDivPos_ (int * *pos,* int * *procGrpNo,* int * *ierr*)

自ランクの領域分割位置を取得

- GetDivPos のFortran インターフェイス関数

  引数

in	*procGrpNo*	プロセスグループ番号
out	*pos*	自ランクの領域分割位置 (3word の整数配列)
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 650 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().

7.21.3.21 CPM_EXTERN void cpm_GetGlobalOrigin_ (double * *worg,* int * *procGrpNo,* int * *ierr*)

全体空間の原点を取得

- GetGlobalOrigin のFortran インターフェイス関数

  引数

in	*procGrpNo*	プロセスグループ番号
out	*worg*	全体空間の原点 (3word の実数配列)
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 440 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALORIGIN, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetGlobalOrigin().

7.21.3.22 CPM_EXTERN void cpm_GetGlobalRegion_ (double * *wrgn,* int * *procGrpNo,* int * *ierr*)

全体空間サイズを取得

- GetGlobalRegion のFortran インターフェイス関数

  引数

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

cpm_ParaManager_frtIF.cpp の 482 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALREGION, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetGlobalRegion().

7.21.3.23 CPM_EXTERN void cpm_GetGlobalVoxelSize_ (int * *wsz,* int * *procGrpNo,* int * *ierr*)

全体ボクセル数を取得

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

  引数

in	*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 398 行で定義されています。

参照先 CPM_ERROR_GET_GLOBALVOXELSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetGlobalVoxelSize().

7.21.3.24 CPM_EXTERN void cpm_GetLocalOrigin_ (double * *lorg,* int * *procGrpNo,* int * *ierr*)

自ランクの空間原点を取得

- GetLocalOrigin のFortran インターフェイス関数

  引数

in	*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 566 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().

7.21.3.25 CPM_EXTERN void cpm_GetLocalRegion_ (double * *lrgn,* int * *procGrpNo,* int * *ierr*)

自ランクの空間サイズを取得

- GetLocalRegion のFortran インターフェイス関数

  引数

in	*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 608 行で定義されています。

参照先 CPM_ERROR_GET_LOCALREGION, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetLocalOrigin().

7.21.3.26 CPM_EXTERN void cpm_GetLocalVoxelSize_ (int * *lsz,* int * *procGrpNo,* int * *ierr*)

自ランクのボクセル数を取得

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

  引数

in	*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 524 行で定義されています。

参照先 CPM_ERROR_GET_LOCALVOXELSIZE, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetLocalVoxelSize().

7.21.3.27 CPM_EXTERN void cpm_GetMyRankID_ (int * *id,* int * *procGrpNo,* int * *ierr*)

ランク番号の取得

- GetMyRankID のFortran インターフェイス関数

  引数

in	*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 868 行で定義されています。

参照先 CPM_ERROR_GET_MYRANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetMyRankID().

7.21.3.28 CPM_EXTERN void cpm_GetNeighborRankID_ (int * *nID,* int * *procGrpNo,* int * *ierr*)

自ランクの隣接ランク番号を取得

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

引数

in	*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 778 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().

7.21.3.29 CPM_EXTERN void cpm_GetNumRank_ (int * *nrank,* int * *procGrpNo,* int * *ierr*)

ランク数の取得

- GetNumRank のFortran インターフェイス関数

  引数

in	*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 906 行で定義されています。

参照先 CPM_ERROR_GET_NUMRANK, CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, cpm_ParaManager::get_instance(), と cpm_ParaManager::GetNumRank().

7.21.3.30 CPM_EXTERN void cpm_GetPeriodicRankID_ (int * *nID,* int * *procGrpNo,* int * *ierr*)

自ランクの周期境界の隣接ランク番号を取得

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

  引数

in	*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 823 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().

7.21.3.31 CPM_EXTERN void cpm_GetPitch_ (double * *pch,* int * *procGrpNo,* int * *ierr*)

ピッチを取得

- GetPitch のFortran インターフェイス関数

  引数

in	*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 356 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().

7.21.3.32 CPM_EXTERN void cpm_GetVoxelHeadIndex_ (int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの始点VOXEL の全体空間でのインデクスを取得

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

引数

in	*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 693 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().

7.21.3.33 CPM_EXTERN void cpm_GetVoxelTailIndex_ (int * *idx*, int * *procGrpNo*, int * *ierr*)

自ランクの終点VOXEL の全体空間でのインデクスを取得

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

引数

in	*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 736 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().

7.21.3.34 CPM_EXTERN void cpm_Initialize_ (int * *domainType*, 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

in	*domainType*	領域分割タイプ (0:カーテシアン、1:LMR))
out	*ierr*	終了コード (0=正常終了、0 以外=cpm_ErrorCode の値)

cpm_ParaManager_frtIF.cpp の 149 行で定義されています。

参照先 CPM_DOMAIN_CARTESIAN, CPM_DOMAIN_LMR, CPM_DOMAIN_UNKNOWN, CPM_ERROR_PM_INSTANCE, CPM_SUCCESS, と cpm_ParaManager::get_instance().

7.21.3.35 CPM_EXTERN void cpm_Irecv_ (void * *buf*, int * *count*, int * *datatype*, int * *source*, int * *procGrpNo*, int * *reqNo*, int * *ierr*)

Irecv

- Irecv のFortran インターフェイス関数

  引数

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 1219 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Irecv(), CPM_SUCCESS, と cpm_ParaManager::get_instance().

7.21.3.36 CPM_EXTERN void cpm_Isend_ (void * *buf*, int * *count*, int * *datatype*, int * *dest*, int * *procGrpNo*, int * *reqNo*, int * *ierr*)

Isend

- Isend のFortran インターフェイス関数

  引数

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 1180 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Isend(), CPM_SUCCESS, と cpm_ParaManager::get_instance().

7.21.3.37 CPM_EXTERN void cpm_IsParallel_ (int * *ipara*, int * *ierr*)

並列実行であるかチェックする

- IsParallel のFortran インターフェイス関数

引数

out	*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 280 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), と cpm_ParaManager::IsParallel().

7.21.3.38 CPM_EXTERN void cpm_PeriodicCommS3D_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *dir*, int * *pm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

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

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

引数

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 2288 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4D_(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS3D().

7.21.3.39 CPM_EXTERN void cpm_PeriodicCommS4D_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *nmax*, int * *vc*, int * *vc_comm*, int * *dir*, int * *pm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

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

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

引数

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 2225 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS4D().

参照元 cpm_PeriodicCommS3D_(), と cpm_PeriodicCommV3D_().

7.21.3.40 CPM_EXTERN void cpm_PeriodicCommS4DEx_ (void * *array*, int * *nmax*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *dir*, int * *pm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

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

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

引数

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 2427 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommS4DEx().

参照元 cpm_PeriodicCommV3DEx_().

7.21.3.41 CPM_EXTERN void cpm_PeriodicCommV3D_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *dir*, int * *pm*, int * *datatype*, int * *procGrpNo*, int * *ierr*)

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

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

引数

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 2357 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4D_(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3D().

7.21.3.42 CPM_EXTERN void cpm_PeriodicCommV3DEx_ (void * *array,* int * *imax,* int * *jmax,* int * *kmax,* int * *vc,* int * *vc_comm,* int * *dir,* int * *pm,* int * *datatype,* int * *procGrpNo,* int * *ierr*)

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

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

引数

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 2490 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PERIODIC_INVALID_DIR, CPM_ERROR_PERIODIC_INVALID_PM, CPM_ERROR_PM_INSTANCE, cpm_PeriodicCommS4DEx_(), cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::PeriodicCommV3DEx().

7.21.3.43 CPM_EXTERN void cpm_Recv_ (void * *buf,* int * *count,* int * *datatype,* int * *source,* int * *procGrpNo,* int * *ierr*)

Recv

- Recv のFortran インターフェイス関数

引数

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 1139 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::Recv().

7.21.3.44 CPM_EXTERN void cpm_Send_ (void * *buf,* int * *count,* int * *datatype,* int * *dest,* int * *procGrpNo,* int * *ierr*)

Send

- Send のFortran インターフェイス関数

引数

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 1099 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_MPI_INVALID_DATATYPE, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::get_instance(), cpm_ParaManager::GetMPI_Datatype(), と cpm_ParaManager::Send().

7.21.3.45 CPM_EXTERN void cpm_SetBndCommBuffer_ (int * *maxVC,* int * *maxN,* int * *procGrpNo,* int * *ierr*)

袖通信バッファのセット (Fortran インターフェイス)

- 袖通信バッファ確保処理のFortran インターフェイス関数

引数

in	*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 1507 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().

7.21.3.46 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

- 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 202 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().

7.21.3.47 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

- 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 245 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().

7.21.3.48 CPM_EXTERN void cpm_Wait_ (int ∗ *reqNo*, int ∗ *ierr*)

Wait

- Wait のFortran インターフェイス関数

引数

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

cpm_ParaManager_frtIF.cpp の 998 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Wait(), と cpm_ParaManager::get_instance().

7.21.3.49 CPM_EXTERN void cpm_wait_BndCommS3D_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

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

- (imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS3D のFortran インターフェイス関数

引数

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 1864 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().

7.21.3.50 CPM_EXTERN void cpm_wait_BndCommS4D_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *nmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

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

- (imax,jmax,kmax,nmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS4D のFortran インターフェイス関数

引数

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 1824 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().

7.21.3.51 CPM_EXTERN void cpm_wait_BndCommS4DEx_ (void * *array*, int * *nmax*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

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

- (nmax,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommS4DEx のFortran インターフェイス関数

引数

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 2138 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().

7.21.3.52 CPM_EXTERN void cpm_wait_BndCommV3D_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

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

- (imax,jmax,kmax,3) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommV3D のFortran インターフェイス関数

引数

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 1909 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().

7.21.3.53 CPM_EXTERN void cpm_wait_BndCommV3DEx_ (void * *array*, int * *imax*, int * *jmax*, int * *kmax*, int * *vc*, int * *vc_comm*, int * *datatype*, int * *reqlist*, int * *procGrpNo*, int * *ierr*)

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

- (3,imax,jmax,kmax) の形式の配列の非同期版袖通信の wait と展開を行う
- wait_BndCommV3DEx のFortran インターフェイス関数
  引数

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 2178 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().

7.21.3.54 CPM_EXTERN void cpm_Waitall_ (int * *count*, int * *reqlist*, int * *ierr*)

Waitall

- Waitall のFortran インターフェイス関数
  引数

in	*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 1027 行で定義されています。

参照先 CPM_ERROR_INVALID_PTR, CPM_ERROR_PM_INSTANCE, cpm_ParaManager::cpm_Waitall(), と cpm_ParaManager::get_instance().

7.22 cpm_ParaManager_inline.h

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

7.22.1 説明

パラレルマネージャクラスの inline 関数ヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager_inline.h で定義されています。

7.23 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

7.23.1 説明

パラレルマネージャクラスのMPI インターフェイス関数ソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManager_MPI.cpp で定義されています。

7.24 cpm_ParaManagerCART.cpp

```
#include "cpm_ParaManagerCART.h"
#include "cpm_VoxelInfoCART.h"
```

cpm_ParaManagerCART.cpp のインクルード依存関係図

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ソースファイル

作者

University of Tokyo

日付

2015/03/27

cpm_ParaManagerCART.cpp で定義されています。

7.25 cpm_ParaManagerCART.h

```
#include "cpm_ParaManager.h"
#include "inline/cpm_ParaManager_BndComm_CART.h"
#include "inline/cpm_ParaManager_BndCommEx_CART.h"
```

cpm_ParaManagerCART.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_ParaManagerCART

7.25.1 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

作者

University of Tokyo

日付

2015/03/27

cpm_ParaManagerCART.h で定義されています。

7.26 cpm_ParaManagerLMR.cpp

```
#include "cpm_ParaManagerLMR.h"
#include "cpm_VoxelInfoLMR.h"
```

cpm_ParaManagerLMR.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ソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_ParaManagerLMR.cpp で定義されています。

7.27 cpm_ParaManagerLMR.h

```
#include "cpm_ParaManager.h"
#include "inline/cpm_ParaManager_BndComm_LMR.h"
#include "inline/cpm_ParaManager_BndCommEx_LMR.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

- struct S_BNDCOMM_BUFFER_LMR
- class cpm_ParaManagerLMR

7.27.1 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

作者

University of Tokyo

日付

2015/03/27

cpm_ParaManagerLMR.h で定義されています。

7.28 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)

7.28.1 説明

ファイルパス文字列関連ユーティリティヘッダーファイル

作者

University of Tokyo

日付

2013/04/02

cpm_PathUtil.h で定義されています。

7.29 cpm_TextParser.cpp

```
#include "cpm_TextParser.h"
```
cpm_TextParser.cpp のインクルード依存関係図

7.29.1 説明

TextParser クラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_TextParser.cpp で定義されています。

7.30 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

7.30.1 説明

テキストパーサークラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_TextParser.h で定義されています。

7.31 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

7.31.1 説明

CPM 領域情報のTextParser クラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

LMR 用領域情報のTextParser クラスのソースファイル

作者

University of Tokyo

日付

2015/03/27

cpm_TextParserDomain.cpp で定義されています。

7.32 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

7.32.1 説明

領域情報のテキストパーサークラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_TextParserDomain.h で定義されています。

7.33 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

7.34 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

7.34.1 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

作者

University of Tokyo

日付

2012/05/31

cpm_TextParserDomainLMR.h で定義されています。

7.35 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 "2.0.0"
- #define CPM_REVISION "20150605_1500"

7.35.1 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。

7.35.2 マクロ定義

7.35.2.1 #define CPM_REVISION "20150605_1500"

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().

7.35.2.2 #define CPM_VERSION_NO "2.0.0"

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().

7.36 cpm_VoxelInfo.cpp

```
#include "cpm_VoxelInfo.h"
```

cpm_VoxelInfo.cpp のインクルード依存関係図

7.36.1 説明

VOXEL 空間情報クラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_VoxelInfo.cpp で定義されています。

7.37 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

7.37.1 説明

VOXEL 空間情報クラスのヘッダーファイル

作者

University of Tokyo

日付

2012/05/31

cpm_VoxelInfo.h で定義されています。

7.38 cpm_VoxelInfoCART.cpp

```
#include "cpm_VoxelInfoCART.h"
```

cpm_VoxelInfoCART.cpp のインクルード依存関係図

7.38.1 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

作者

University of Tokyo

日付

2015/03/27

cpm_VoxelInfoCART.cpp で定義されています。

7.39 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

7.39.1 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

作者

University of Tokyo

日付

2015/03/27

cpm_VoxelInfoCART.h で定義されています。

7.40 cpm_VoxelInfoLMR.cpp

```
#include "cpm_VoxelInfoLMR.h"
```
cpm_VoxelInfoLMR.cpp のインクルード依存関係図

7.40.1 説明

VOXEL 空間情報クラスのソースファイル

作者

University of Tokyo

日付

2012/05/31

cpm_VoxelInfoLMR.cpp で定義されています。

7.41 cpm_VoxelInfoLMR.h

```
#include "cpm_VoxelInfo.h"
#include "BCMOctree.h"
#include "BCMFileCommon.h"
#include "cpm_TextParserDomainLMR.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

7.41.1 説明

LMR 用のVOXEL 空間情報クラスのヘッダーファイル

作者

University of Tokyo

日付

2015/03/27

cpm_VoxelInfoLMR.h で定義されています。

7.42 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**).*

7.42.1 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。

7.43 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.

7.43.1 説明

隣接情報クラス

NeighborInfo.h で定義されています。

7.44 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**.

7.44.1 説明

Octree 用ノードクラス

Node.h で定義されています。

7.45 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**.

7.45.1 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。

7.46 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**.

7.46.1 説明

Octree 用Pedigree クラス

Pedigree.h で定義されています。

7.46.2 関数

7.46.2.1 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().

7.47 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

7.47.1 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。

7.48 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)

7.48.1 説明

version 1.1 2014-04-23

作者

aics

Vec3.h で定義されています。

7.48.2 マクロ定義

7.48.2.1 #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 48 行で定義されています。

Index