

Polylib  
2.5

Generated by Doxygen 1.8.4

Wed Jun 19 2013 19:14:31



# Contents

<b>1</b>	<b>Hierarchical Index</b>	<b>1</b>
1.1	Class Hierarchy . . . . .	1
<b>2</b>	<b>Class Index</b>	<b>3</b>
2.1	Class List . . . . .	3
<b>3</b>	<b>Class Documentation</b>	<b>5</b>
3.1	PolylibNS::BBox Class Reference . . . . .	5
3.1.1	Detailed Description . . . . .	5
3.1.2	Member Function Documentation . . . . .	6
3.1.2.1	contain . . . . .	6
3.1.2.2	crossed . . . . .	6
3.1.2.3	getCrossedRegion . . . . .	6
3.1.2.4	getFace . . . . .	6
3.1.2.5	getSide . . . . .	6
3.1.2.6	vec3to2 . . . . .	7
3.2	PolylibNS::CalcAreaInfo Struct Reference . . . . .	7
3.2.1	Detailed Description . . . . .	7
3.3	PolylibNS::MPIPolylib Class Reference . . . . .	7
3.3.1	Detailed Description . . . . .	9
3.3.2	Constructor & Destructor Documentation . . . . .	9
3.3.2.1	MPIPolylib . . . . .	9
3.3.2.2	~MPIPolylib . . . . .	9
3.3.3	Member Function Documentation . . . . .	9
3.3.3.1	broadcast_config . . . . .	9
3.3.3.2	broadcast_config_from_rank0 . . . . .	9
3.3.3.3	erase_outbounded_polygons . . . . .	9
3.3.3.4	gather_polygons . . . . .	10
3.3.3.5	get_instance . . . . .	10
3.3.3.6	get_myproc . . . . .	10
3.3.3.7	get_proc . . . . .	10
3.3.3.8	init_parallel_info . . . . .	10

3.3.3.9	load	10
3.3.3.10	load_parallel	11
3.3.3.11	load_rank0	11
3.3.3.12	migrate	11
3.3.3.13	move	11
3.3.3.14	pack_num_trias	12
3.3.3.15	pack_tria_ids	12
3.3.3.16	pack_trias	12
3.3.3.17	receive_polygons_from_rank0	12
3.3.3.18	save	13
3.3.3.19	save_parallel	13
3.3.3.20	save_rank0	13
3.3.3.21	select_excluded_trias	14
3.3.3.22	send_polygons_to_all	14
3.3.3.23	send_polygons_to_rank0	14
3.3.3.24	show_group_name	14
3.3.3.25	used_memory_size	14
3.4	PolylibNS::ParallelInfo Struct Reference	15
3.4.1	Detailed Description	15
3.5	PolylibNS::PolygonGroup Class Reference	15
3.5.1	Detailed Description	17
3.5.2	Constructor & Destructor Documentation	17
3.5.2.1	PolygonGroup	17
3.5.2.2	~PolygonGroup	17
3.5.3	Member Function Documentation	17
3.5.3.1	acq_file_name	17
3.5.3.2	acq_fullpath	18
3.5.3.3	add_children	18
3.5.3.4	add_triangles	18
3.5.3.5	build_group_tree	18
3.5.3.6	build_polygon_tree	18
3.5.3.7	check_leaped	19
3.5.3.8	get_children	19
3.5.3.9	get_class_name	19
3.5.3.10	get_file_name	19
3.5.3.11	get_id	20
3.5.3.12	get_internal_id	20
3.5.3.13	get_label	20
3.5.3.14	get_movable	20
3.5.3.15	get_name	20

3.5.3.16	<a href="#">get_num_of_trias_before_move</a>	20
3.5.3.17	<a href="#">get_parent</a>	20
3.5.3.18	<a href="#">get_parent_path</a>	21
3.5.3.19	<a href="#">get_triangles</a>	21
3.5.3.20	<a href="#">get_vtree</a>	21
3.5.3.21	<a href="#">init</a>	21
3.5.3.22	<a href="#">init_check_leaped</a>	21
3.5.3.23	<a href="#">is_far</a>	22
3.5.3.24	<a href="#">linear_search</a>	23
3.5.3.25	<a href="#">linear_search</a>	23
3.5.3.26	<a href="#">load_id_file</a>	23
3.5.3.27	<a href="#">load_stl_file</a>	24
3.5.3.28	<a href="#">mk_param_tag</a>	24
3.5.3.29	<a href="#">move</a>	24
3.5.3.30	<a href="#">rebuild_polygons</a>	25
3.5.3.31	<a href="#">save_id_file</a>	25
3.5.3.32	<a href="#">save_stl_file</a>	25
3.5.3.33	<a href="#">save_stl_file</a>	25
3.5.3.34	<a href="#">search</a>	26
3.5.3.35	<a href="#">search</a>	26
3.5.3.36	<a href="#">search_nearest</a>	26
3.5.3.37	<a href="#">search_outbounded</a>	27
3.5.3.38	<a href="#">set_children</a>	27
3.5.3.39	<a href="#">set_file_name</a>	27
3.5.3.40	<a href="#">set_name</a>	27
3.5.3.41	<a href="#">set_parent</a>	27
3.5.3.42	<a href="#">set_parent_path</a>	28
3.5.3.43	<a href="#">setup_attribute</a>	28
3.5.3.44	<a href="#">show_group_info</a>	28
3.5.3.45	<a href="#">whoami</a>	28
3.5.4	<a href="#">Member Data Documentation</a>	29
3.5.4.1	<a href="#">ATT_NAME_CLASS</a>	29
3.6	<a href="#">PolylibNS::PolygonGroupFactory Class Reference</a>	29
3.6.1	<a href="#">Detailed Description</a>	29
3.6.2	<a href="#">Constructor &amp; Destructor Documentation</a>	29
3.6.2.1	<a href="#">PolygonGroupFactory</a>	29
3.6.2.2	<a href="#">~PolygonGroupFactory</a>	29
3.6.3	<a href="#">Member Function Documentation</a>	29
3.6.3.1	<a href="#">create_instance</a>	29
3.7	<a href="#">PolylibNS::Polygons Class Reference</a>	30

3.7.1	Detailed Description	30
3.7.2	Constructor & Destructor Documentation	30
3.7.2.1	Polygons	30
3.7.2.2	~Polygons	30
3.7.3	Member Function Documentation	31
3.7.3.1	add	31
3.7.3.2	build	31
3.7.3.3	get_tri_list	31
3.7.3.4	get_vtree	31
3.7.3.5	import	31
3.7.3.6	init	31
3.7.3.7	linear_search	32
3.7.3.8	linear_search	32
3.7.3.9	search	32
3.7.3.10	search	33
3.7.3.11	search_nearest	33
3.7.3.12	set_all_exid	33
3.7.3.13	triangles_num	34
3.8	PolylibNS::Polylib Class Reference	34
3.8.1	Detailed Description	35
3.8.2	Constructor & Destructor Documentation	35
3.8.2.1	Polylib	35
3.8.2.2	~Polylib	35
3.8.3	Member Function Documentation	36
3.8.3.1	add_pg_list	36
3.8.3.2	check_group_name	37
3.8.3.3	clearfilepath	37
3.8.3.4	create_polygon_group	37
3.8.3.5	get_group	37
3.8.3.6	get_group	38
3.8.3.7	get_instance	38
3.8.3.8	get_root_groups	38
3.8.3.9	load	38
3.8.3.10	load_config_file	39
3.8.3.11	load_polygons	39
3.8.3.12	load_with_idfile	39
3.8.3.13	make_group_tree	40
3.8.3.14	make_group_tree	40
3.8.3.15	move	40
3.8.3.16	polylib_config_save_file	40

3.8.3.17	<a href="#">save</a>	41
3.8.3.18	<a href="#">save_config_file</a>	41
3.8.3.19	<a href="#">save_with_rankno</a>	41
3.8.3.20	<a href="#">search_nearest_polygon</a>	42
3.8.3.21	<a href="#">search_polygons</a>	42
3.8.3.22	<a href="#">set_factory</a>	43
3.8.3.23	<a href="#">setfilepath</a>	44
3.8.3.24	<a href="#">show_group_hierarchy</a>	44
3.8.3.25	<a href="#">show_group_info</a>	44
3.8.3.26	<a href="#">show_group_name</a>	44
3.8.3.27	<a href="#">used_memory_size</a>	45
3.9	<a href="#">PolylibNS::PolylibMoveParams Class Reference</a>	45
3.9.1	<a href="#">Detailed Description</a>	45
3.10	<a href="#">PolylibNS::PolylibStat2 Class Reference</a>	45
3.10.1	<a href="#">Detailed Description</a>	45
3.10.2	<a href="#">Member Function Documentation</a>	46
3.10.2.1	<a href="#">String</a>	46
3.11	<a href="#">PolylibNS::PrivateTriangle Class Reference</a>	46
3.11.1	<a href="#">Detailed Description</a>	46
3.11.2	<a href="#">Constructor &amp; Destructor Documentation</a>	47
3.11.2.1	<a href="#">PrivateTriangle</a>	47
3.11.2.2	<a href="#">PrivateTriangle</a>	48
3.11.2.3	<a href="#">PrivateTriangle</a>	48
3.11.2.4	<a href="#">PrivateTriangle</a>	48
3.11.2.5	<a href="#">PrivateTriangle</a>	48
3.11.2.6	<a href="#">PrivateTriangle</a>	48
3.11.3	<a href="#">Member Function Documentation</a>	48
3.11.3.1	<a href="#">get_id</a>	49
3.11.3.2	<a href="#">set_id</a>	49
3.11.4	<a href="#">Member Data Documentation</a>	49
3.11.4.1	<a href="#">m_id</a>	49
3.12	<a href="#">PolylibNS::Triangle Class Reference</a>	49
3.12.1	<a href="#">Detailed Description</a>	50
3.12.2	<a href="#">Constructor &amp; Destructor Documentation</a>	50
3.12.2.1	<a href="#">Triangle</a>	50
3.12.2.2	<a href="#">Triangle</a>	50
3.12.2.3	<a href="#">Triangle</a>	50
3.12.2.4	<a href="#">Triangle</a>	51
3.12.3	<a href="#">Member Function Documentation</a>	52
3.12.3.1	<a href="#">calc_area</a>	52

3.12.3.2	<a href="#">calc_normal</a>	52
3.12.3.3	<a href="#">get_area</a>	52
3.12.3.4	<a href="#">get_exid</a>	52
3.12.3.5	<a href="#">get_normal</a>	52
3.12.3.6	<a href="#">get_shell</a>	52
3.12.3.7	<a href="#">get_vertex</a>	53
3.12.3.8	<a href="#">set_exid</a>	53
3.12.3.9	<a href="#">set_shell</a>	53
3.12.3.10	<a href="#">set_vertexes</a>	53
3.13	<a href="#">TriangleStruct Struct Reference</a>	53
3.13.1	<a href="#">Detailed Description</a>	53
3.14	<a href="#">PolylibNS::TriMesh Class Reference</a>	54
3.14.1	<a href="#">Detailed Description</a>	54
3.14.2	<a href="#">Constructor &amp; Destructor Documentation</a>	54
3.14.2.1	<a href="#">TriMesh</a>	54
3.14.2.2	<a href="#">~TriMesh</a>	54
3.14.3	<a href="#">Member Function Documentation</a>	54
3.14.3.1	<a href="#">add</a>	54
3.14.3.2	<a href="#">build</a>	55
3.14.3.3	<a href="#">get_bbox</a>	55
3.14.3.4	<a href="#">get_vtree</a>	55
3.14.3.5	<a href="#">import</a>	55
3.14.3.6	<a href="#">init</a>	55
3.14.3.7	<a href="#">linear_search</a>	56
3.14.3.8	<a href="#">linear_search</a>	56
3.14.3.9	<a href="#">search</a>	56
3.14.3.10	<a href="#">search</a>	57
3.14.3.11	<a href="#">search_nearest</a>	57
3.14.3.12	<a href="#">set_all_exid</a>	57
3.14.3.13	<a href="#">triangles_num</a>	58
3.15	<a href="#">PolylibNS::TriMeshIO Class Reference</a>	58
3.15.1	<a href="#">Detailed Description</a>	58
3.15.2	<a href="#">Member Function Documentation</a>	58
3.15.2.1	<a href="#">input_file_format</a>	58
3.15.2.2	<a href="#">load</a>	59
3.15.2.3	<a href="#">save</a>	59
3.15.3	<a href="#">Member Data Documentation</a>	59
3.15.3.1	<a href="#">FMT_STL_A</a>	59
3.16	<a href="#">PolylibNS::Vec2&lt; T &gt; Class Template Reference</a>	60
3.16.1	<a href="#">Detailed Description</a>	61



3.17 PolylibNS::Vec3< T > Class Template Reference . . . . .	61
3.17.1 Detailed Description . . . . .	62
3.18 PolylibNS::VElement Class Reference . . . . .	62
3.18.1 Detailed Description . . . . .	62
3.18.2 Constructor & Destructor Documentation . . . . .	62
3.18.2.1 VElement . . . . .	62
3.18.3 Member Function Documentation . . . . .	62
3.18.3.1 get_bbox . . . . .	62
3.18.3.2 get_pos . . . . .	62
3.18.3.3 get_triangle . . . . .	63
3.19 PolylibNS::VNode Class Reference . . . . .	63
3.19.1 Detailed Description . . . . .	63
3.19.2 Constructor & Destructor Documentation . . . . .	63
3.19.2.1 VNode . . . . .	63
3.19.2.2 ~VNode . . . . .	63
3.19.3 Member Function Documentation . . . . .	63
3.19.3.1 get_axis . . . . .	63
3.19.3.2 get_bbox . . . . .	64
3.19.3.3 get_bbox_search . . . . .	64
3.19.3.4 get_elements_num . . . . .	64
3.19.3.5 get_left . . . . .	64
3.19.3.6 get_right . . . . .	64
3.19.3.7 get_vlist . . . . .	64
3.19.3.8 is_leaf . . . . .	64
3.19.3.9 set_axis . . . . .	65
3.19.3.10 set_bbox . . . . .	66
3.19.3.11 set_bbox_search . . . . .	66
3.19.3.12 set_element . . . . .	66
3.19.3.13 split . . . . .	66
3.20 PolylibNS::VTree Class Reference . . . . .	66
3.20.1 Detailed Description . . . . .	67
3.20.2 Constructor & Destructor Documentation . . . . .	67
3.20.2.1 VTree . . . . .	67
3.20.2.2 ~VTree . . . . .	67
3.20.3 Member Function Documentation . . . . .	67
3.20.3.1 destroy . . . . .	67
3.20.3.2 memory_size . . . . .	67
3.20.3.3 search . . . . .	67
3.20.3.4 search . . . . .	68
3.20.3.5 search_nearest . . . . .	69

3.20.3.6 <a href="#">search_nearest_recursive</a> . . . . .	69
---	----

# Chapter 1

## Hierarchical Index

### 1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

PolylibNS::BBox . . . . .	5
PolylibNS::CalcAreaInfo . . . . .	7
PolylibNS::ParallelInfo . . . . .	15
PolylibNS::PolygonGroup . . . . .	15
PolylibNS::PolygonGroupFactory . . . . .	29
PolylibNS::Polygons . . . . .	30
PolylibNS::TriMesh . . . . .	54
PolylibNS::Polylib . . . . .	34
PolylibNS::MPIPolylib . . . . .	7
PolylibNS::PolylibMoveParams . . . . .	45
PolylibNS::PolylibStat2 . . . . .	45
PolylibNS::Triangle . . . . .	49
PolylibNS::PrivateTriangle . . . . .	46
TriangleStruct . . . . .	53
PolylibNS::TriMeshIO . . . . .	58
PolylibNS::Vec2< T > . . . . .	60
PolylibNS::Vec3< T > . . . . .	61
PolylibNS::Vec3< float > . . . . .	61
PolylibNS::VElement . . . . .	62
PolylibNS::VNode . . . . .	63
PolylibNS::VTree . . . . .	66



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">PolylibNS::BBox</a>	5
<a href="#">PolylibNS::CalcAreaInfo</a>	7
<a href="#">PolylibNS::MPIPolylib</a>	7
<a href="#">PolylibNS::ParallelInfo</a>	15
<a href="#">PolylibNS::PolygonGroup</a>	15
<a href="#">PolylibNS::PolygonGroupFactory</a>	29
<a href="#">PolylibNS::Polygons</a>	30
<a href="#">PolylibNS::Polylib</a>	34
<a href="#">PolylibNS::PolylibMoveParams</a>	45
<a href="#">PolylibNS::PolylibStat2</a>	45
<a href="#">PolylibNS::PrivateTriangle</a>	46
<a href="#">PolylibNS::Triangle</a>	49
<a href="#">TriangleStruct</a>	53
<a href="#">PolylibNS::TriMesh</a>	54
<a href="#">PolylibNS::TriMeshIO</a>	58
<a href="#">PolylibNS::Vec2&lt; T &gt;</a>	60
<a href="#">PolylibNS::Vec3&lt; T &gt;</a>	61
<a href="#">PolylibNS::VElement</a>	62
<a href="#">PolylibNS::VNode</a>	63
<a href="#">PolylibNS::VTree</a>	66



## Chapter 3

# Class Documentation

### 3.1 PolylibNS::BBox Class Reference

```
#include <BBox.h>
```

#### Public Member Functions

- **BBox** (float \_minx, float \_miny, float \_minz, float \_maxx, float \_maxy, float \_maxz)
- **BBox** (float \_min[3], float \_max[3])
- **BBox** (const [Vec3f](#) &\_min, const [Vec3f](#) &\_max)
- void **init** ()
- void **setMinMax** (const [Vec3f](#) &\_min, const [Vec3f](#) &\_max)
- void **add** (const [Vec3f](#) &v)
- [Vec3f](#) **getPoint** (int idx) const
- [Vec3f](#) **center** () const
- [Vec3f](#) **size** () const
- float **xsize** () const
- float **ysize** () const
- float **zsize** () const
- float **length** (const AxisEnum &axis) const
- float **diameter** () const
- AxisEnum **getMaxAxis** (float &length) const
- bool **contain** (const [Vec3f](#) &pos) const
- bool **crossed** (const [BBox](#) &bbox) const
- [BBox](#) **getCrossedRegion** ([BBox](#) &other\_bbox) const
- [Vec2f](#) **vec3to2** (int axis\_id, [Vec3f](#) &v3) const
- void **getFace** (int axis\_id, [Vec3f](#) face[2][2]) const
- void **getSide** (int axis\_id, [Vec3f](#) side[4][2]) const

#### Public Attributes

- [Vec3f](#) **min**
- [Vec3f](#) **max**

#### 3.1.1 Detailed Description

クラス:[BBox](#) Bounding Box を管理するクラス

### 3.1.2 Member Function Documentation

#### 3.1.2.1 `bool PolylibNS::BBox::contain ( const Vec3f & pos ) const` `[inline]`

引数で与えられた点が、このBBox に含まれるかを判定する。

##### Parameters

<code>in</code>	<code>pos</code>	試行する点
-----------------	------------------	-------

##### Returns

含まれる場合は true。他は false。

#### 3.1.2.2 `bool PolylibNS::BBox::crossed ( const BBox & bbox ) const` `[inline]`

BBox と BBox の交差判定を行う。KD-Tree の交差判定と同じ。

##### Parameters

<code>in</code>	<code>bbox</code>	試行するBBox
-----------------	-------------------	----------

##### Returns

交差する場合は true。他は false。

#### 3.1.2.3 `BBox PolylibNS::BBox::getCrossedRegion ( BBox & other_bbox ) const` `[inline]`

BBox と BBox の重複領域の抽出を行う。自身の面と他方の辺との交差判定を行う。

##### Parameters

<code>in</code>	<code>other_bbox</code>	試行するBBox
-----------------	-------------------------	----------

##### Returns

交差する場合は true。他は false。

#### 3.1.2.4 `void PolylibNS::BBox::getFace ( int axis_id, Vec3f face[2][2] ) const` `[inline]`

引数 `axis_id`(0=x,1=y,z=2) に垂直な、このBBox の面の対角点を返す。

##### Parameters

<code>in</code>	<code>axis_id</code>	軸番号。0=x 軸、1=y 軸、2=z 軸。
<code>in</code>	<code>face</code>	BBox の面の中で、軸に垂直な面の対角点。

#### 3.1.2.5 `void PolylibNS::BBox::getSide ( int axis_id, Vec3f side[4][2] ) const` `[inline]`

引数 `axis_id`(0=x,1=y,z=2) に平行な、このBBox の辺の端点を返す。



## Parameters

in	<i>axis_id</i>	軸番号。0=x 軸、1=y 軸、2=z 軸。
in	<i>side</i>	BBox の辺の中で、軸に平行な辺の端点。

3.1.2.6 Vec2f PolylibNS::BBox::vec3to2 ( int *axis\_id*, Vec3f & *v3* ) const [inline]

引数 *axis\_id*(0=x,1=y,z=2) に垂直な成分を詰めて返す。

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/common/BBox.h

## 3.2 PolylibNS::CalcAreaInfo Struct Reference

```
#include <Polylib.h>
```

## Public Attributes

- [Vec3f m\\_bpos](#)  
基点座標
- [Vec3f m\\_bbsize](#)  
計算領域のボクセル数
- [Vec3f m\\_gcsize](#)  
ガイドセルのボクセル数
- [Vec3f m\\_dx](#)  
ボクセル 1 辺の長さ
- [Vec3f m\\_gcell\\_min](#)  
ガイドセルを含めた担当領域の最小位置
- [Vec3f m\\_gcell\\_max](#)  
ガイドセルを含めた担当領域の最大位置
- [BBox m\\_gcell\\_bbox](#)  
ガイドセルを含めた *Bounding Box*

## 3.2.1 Detailed Description

クラス:[CalcAreaInfo](#) 計算領域情報。

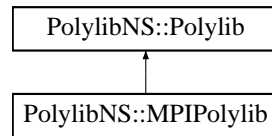
The documentation for this struct was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/Polylib.h

## 3.3 PolylibNS::MPIPolylib Class Reference

```
#include <MPIPolylib.h>
```

Inheritance diagram for PolylibNS::MPIPolylib:



## Public Member Functions

- POLYLIB\_STAT [init\\_parallel\\_info](#) (MPI\_Comm comm, float bpos[3], unsigned int bbsize[3], unsigned int gcsz[3], float dx[3])
- POLYLIB\_STAT [load](#) (std::string config\_filename)
- POLYLIB\_STAT [load\\_rank0](#) (std::string config\_filename="", float scale=1.0)
- POLYLIB\_STAT [load\\_parallel](#) (std::string config\_filename="", ID\_FORMAT id\_format=ID\_BIN)
- POLYLIB\_STAT [save](#) (std::string \*p\_config\_filename)
- POLYLIB\_STAT [save\\_rank0](#) (std::string \*p\_config\_filename, std::string stl\_format, std::string extend="")
- POLYLIB\_STAT [save\\_parallel](#) (std::string \*p\_config\_filename, std::string stl\_format, std::string extend="", ID\_FORMAT id\_format=ID\_BIN)
- POLYLIB\_STAT [move](#) ([PolylibMoveParams](#) &params)
- POLYLIB\_STAT [migrate](#) ()
- [ParallelInfo](#) [get\\_myproc](#) ()
- unsigned int [used\\_memory\\_size](#) ()

## Static Public Member Functions

- static [MPIPolylib](#) \* [get\\_instance](#) ()

## Protected Member Functions

- [MPIPolylib](#) ()
- [~MPIPolylib](#) ()
- void [show\\_group\\_name](#) ([PolygonGroup](#) \*p, std::string tab)
- POLYLIB\_STAT [broadcast\\_config](#) (std::string config\_contents)
- POLYLIB\_STAT [send\\_polygons\\_to\\_all](#) ()
- POLYLIB\_STAT [pack\\_num\\_trias](#) (std::vector< int > \*p\_vec, int group\_id, const std::vector< [PrivateTriangle](#) \* > \*p\_trias)
- POLYLIB\_STAT [pack\\_trias](#) (std::vector< float > \*p\_vec, const std::vector< [PrivateTriangle](#) \* > \*p\_trias)
- POLYLIB\_STAT [pack\\_tria\\_ids](#) (std::vector< int > \*p\_vec, const std::vector< [PrivateTriangle](#) \* > \*p\_trias)
- POLYLIB\_STAT [erase\\_outbounded\\_polygons](#) ()
- POLYLIB\_STAT [broadcast\\_config\\_from\\_rank0](#) ()
- POLYLIB\_STAT [receive\\_polygons\\_from\\_rank0](#) ()
- POLYLIB\_STAT [gather\\_polygons](#) ()
- POLYLIB\_STAT [send\\_polygons\\_to\\_rank0](#) ()
- POLYLIB\_STAT [select\\_excluded\\_trias](#) ([PolygonGroup](#) \*p\_pg)
- [ParallelInfo](#) \* [get\\_proc](#) (int rank)

## Protected Attributes

- [ParallelInfo](#) m\_myproc  
自PE 担当領域情報
- std::vector< [ParallelInfo](#) \* > m\_other\_procs  
自PE を除く全PE 担当領域情報リスト
- std::vector< [ParallelInfo](#) \* > m\_neighbour\_procs

- 隣接PE 担当領域情報リスト
- int [m\\_myrank](#)  
自プロセスのランク数
- int [m\\_numproc](#)  
全プロセス数
- MPI\_Comm [m\\_mycomm](#)  
自プロセスが利用するコミュニケーター

## Additional Inherited Members

### 3.3.1 Detailed Description

クラス:[MPIPolylib](#) ポリゴンを管理する為の並列版クラスライブラリです。

### 3.3.2 Constructor & Destructor Documentation

#### 3.3.2.1 PolylibNS::MPIPolylib::MPIPolylib ( ) [protected]

コンストラクタ。 singleton のため非公開。本クラスインスタンス取得には `get_instance()` を利用する。

#### 3.3.2.2 PolylibNS::MPIPolylib::~~MPIPolylib ( ) [protected]

デストラクタ。

### 3.3.3 Member Function Documentation

#### 3.3.3.1 POLYLIB\_STAT PolylibNS::MPIPolylib::broadcast\_config ( std::string *config\_contents* ) [protected]

設定ファイル内容を他 rank へ broadcast する。

##### Parameters

<i>in</i>	<i>config_contents</i>	初期化ファイル内容。
-----------	------------------------	------------

##### Returns

POLYLIB\_STAT で定義される値が返る。

#### 3.3.3.2 POLYLIB\_STAT PolylibNS::MPIPolylib::broadcast\_config\_from\_rank0 ( ) [protected]

ポリゴングループ定義情報を rank0 から受信し、グループ階層構造を構築。

##### Returns

POLYLIB\_STAT で定義される値が返る。

#### 3.3.3.3 POLYLIB\_STAT PolylibNS::MPIPolylib::erase\_outbounded\_polygons ( ) [protected]

自領域内ポリゴンのみ抽出してポリゴン情報を再構築。 migrate 実行後に行う。

##### Returns

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.4 POLYLIB\_STAT PolylibNS::MPIPolylib::gather\_polygons ( ) [protected]

他 rank からポリゴン情報を rank0 で受信

### 3.3.3.5 static MPIPolylib\* PolylibNS::MPIPolylib::get\_instance ( ) [static]

インスタンス取得。本クラスは singleton クラスです。

#### Returns

MPIPolylib クラスのインスタンス

### 3.3.3.6 ParallelInfo PolylibNS::MPIPolylib::get\_myproc ( ) [inline]

m\_myproc の内容を get

#### Returns

自PE 領域情報

### 3.3.3.7 ParallelInfo\* PolylibNS::MPIPolylib::get\_proc ( int rank ) [protected]

プロセス担当領域クラスのポインタを返す

#### Parameters

in	rank	ランク数
----	------	------

#### Returns

プロセス担当領域クラスのポインタ

### 3.3.3.8 POLYLIB\_STAT PolylibNS::MPIPolylib::init\_parallel\_info ( MPI\_Comm comm, float bpos[3], unsigned int bbsize[3], unsigned int gcsize[3], float dx[3] )

並列計算関連情報の設定と初期化を行う。全 rank で各々設定を行い、その領域情報を全 rank へ配信する。

#### Parameters

in	comm	MPI コミュニケーター
in	bpos	自PE 担当領域の基点座標
in	bbsize	同、計算領域のボクセル数
in	gcsize	同、ガイドセルのボクセル数
in	dx	同、ボクセル 1 辺の長さ

#### Returns

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.9 POLYLIB\_STAT PolylibNS::MPIPolylib::load ( std::string config\_filename ) [inline]

[Polylib::load\(\)](#) のオーバーライドメソッド。

#### Attention

並列環境では利用できません。

## Parameters

in	<i>config_filename</i>	初期化ファイル名。
----	------------------------	-----------

## Returns

常に PLSTAT\_NG が返ります。

### 3.3.3.10 POLYLIB\_STAT PolylibNS::MPIPolylib::load\_parallel ( std::string *config\_filename* = "", ID\_FORMAT *id\_format* = ID\_BIN )

全 rank 並列でのデータ構築。指定された設定ファイルを各 rank にて読み込み、グループ階層構造の構築、およびポリゴンデータの構築を行う。

## Attention

各 rank が読み込むファイルに記述されたグループ階層構造が一致している必要がある。

## Parameters

in	<i>config_filename</i>	初期化ファイル名。未指定時はデフォルトファイルを読む。
in	<i>id_format</i>	三角形ID ファイルの入力形式。

## Returns

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.11 POLYLIB\_STAT PolylibNS::MPIPolylib::load\_rank0 ( std::string *config\_filename* = "", float *scale* = 1.0 )

rank0 によるデータ構築。指定された設定ファイルを rank0 にて読み込み、グループ階層構造の構築 およびポリゴンデータの構築を行う。グループ階層構造は全 rank に b\_cast され、情報を共有する。ポリゴンデータは各 rank 領域毎のデータが分配される。

## Parameters

in	<i>config_filename</i>	初期化ファイル名。未指定時はデフォルトファイルを読む。
----	------------------------	-----------------------------

## Returns

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.12 POLYLIB\_STAT PolylibNS::MPIPolylib::migrate ( )

ポリゴンデータのPE 間移動。本クラスインスタンス配下の全PolygonGroup のポリゴンデータについて、move メソッドにより移動した三角形ポリゴン情報を隣接PE 間でやり取りする。

## Returns

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.13 POLYLIB\_STAT PolylibNS::MPIPolylib::move ( PolylibMoveParams & *params* )

ポリゴン座標の移動。本クラスインスタンス配下の全PolygonGroup の move メソッドが呼び出される。

## Parameters

in	<i>params</i>	移動計算要パラメタセット。
----	---------------	---------------

## Returns

POLYLIB\_STAT で定義される値が返る。

**3.3.3.14 POLYLIB\_STAT PolylibNS::MPIPolylib::pack\_num\_trias ( std::vector< int > \* *p\_vec*, int *group\_id*, const std::vector< PrivateTriangle \* > \* *p\_trias* )** [protected]

グループID & グループ内三角形数の送信情報を作成。

## Parameters

in, out	<i>p_vec</i>	情報追加先ベクタ
in	<i>group_id</i>	グループID
in	<i>p_trias</i>	グループ内三角形リスト

## Returns

POLYLIB\_STAT で定義される値が返る。

**3.3.3.15 POLYLIB\_STAT PolylibNS::MPIPolylib::pack\_tria\_ids ( std::vector< int > \* *p\_vec*, const std::vector< PrivateTriangle \* > \* *p\_trias* )** [protected]

三角形ID の送信情報を作成。

## Parameters

in, out	<i>p_vec</i>	情報追加先ベクタ
in	<i>p_trias</i>	グループ内三角形リスト

## Returns

POLYLIB\_STAT で定義される値が返る。

**3.3.3.16 POLYLIB\_STAT PolylibNS::MPIPolylib::pack\_trias ( std::vector< float > \* *p\_vec*, const std::vector< PrivateTriangle \* > \* *p\_trias* )** [protected]

三角形の送信情報を作成。

## Parameters

in, out	<i>p_vec</i>	情報追加先ベクタ
in	<i>p_trias</i>	グループ内三角形リスト

## Returns

POLYLIB\_STAT で定義される値が返る。

**3.3.3.17 POLYLIB\_STAT PolylibNS::MPIPolylib::receive\_polygons\_from\_rank0 ( )** [protected]

自領域に必要なポリゴン情報を rank0 から受信

## Returns

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.18 POLYLIB\_STAT PolylibNS::MPIPolylib::save ( std::string \* *p\_config\_filename* ) [inline]

[Polylib::save\(\)](#)のオーバーロードメソッド。

#### Attention

並列環境では利用できません。

#### Parameters

out	<i>p_config_filename</i>	初期化ファイル名。
-----	--------------------------	-----------

#### Returns

常に PLSTAT\_NG が返ります。

### 3.3.3.19 POLYLIB\_STAT PolylibNS::MPIPolylib::save\_parallel ( std::string \* *p\_config\_filename*, std::string *stl\_format*, std::string *extend* = " ", ID\_FORMAT *id\_format* = ID\_BIN )

全 rank 並列でのデータ保存。各 rank の本クラスインスタンスが保持するグループ階層構造を設定ファイルに各 rank 毎に書き出す。同時にポリゴンデータも指定されたフォーマットの STL ファイルに各 rank 毎に書き出す。設定ファイル命名規則は以下の通り `polylib_config_ランク番号_付加文字列.xml` `polylib_config_ランク番号_付加文字列.tpp` STL ファイル命名規則は以下の通り `ポリゴングループ名称_ランク番号_付加文字列.拡張子`

#### Parameters

out	<i>p_config_filename</i>	設定ファイル名返却用 string インスタンスへのポインタ
in	<i>stl_format</i>	STL ファイルフォーマット。"stl_a":アスキー形式 "stl_b":バイナリ形式
in	<i>extend</i>	ファイル名に付加する文字列。省略可。省略した場合は、付加文字列として本メソッド呼び出し時の年月日時分秒 (YYYYMMDD24hmmss) を用いる。
in	<i>id_format</i>	三角形ID ファイルの出力形式。

#### Returns

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.20 POLYLIB\_STAT PolylibNS::MPIPolylib::save\_rank0 ( std::string \* *p\_config\_filename*, std::string *stl\_format*, std::string *extend* = " " )

rank0 によるデータ保存。rank0 の本クラスインスタンスが保持するグループ階層構造を設定ファイルに書き出す。同時に各 rank に分散するポリゴンデータも rank0 に集められ、指定されたフォーマットの STL ファイルに rank0 で書き出す。設定ファイル命名規則は以下の通り `polylib_config_付加文字列.xml` `polylib_config_付加文字列.tpp` STL ファイル命名規則は以下の通り `ポリゴングループ名称_付加文字列.拡張子`

#### Parameters

out	<i>p_config_filename</i>	設定ファイル名返却用 string インスタンスへのポインタ
-----	--------------------------	--------------------------------

in	<i>stl_format</i>	STL ファイルフォーマット。"stl_a":アスキー形式 "stl_b":バイナリ形式
in	<i>extend</i>	ファイル名に付加する文字列。省略可。省略した場合は、付加文字列として本メソッド呼び出し時の年月日時分秒 (YYYYMMDD24hhmmss) を用いる。

**Returns**

POLYLIB\_STAT で定義される値が返る。

**Attention**

出力引数 p\_config\_filename の返却値は rank0 でのみ有効

### 3.3.3.21 POLYLIB\_STAT PolylibNS::MPIPolylib::select\_excluded\_trias ( PolygonGroup \* p\_pg ) [protected]

移動除外三角形ID リストの作成

### 3.3.3.22 POLYLIB\_STAT PolylibNS::MPIPolylib::send\_polygons\_to\_all ( ) [protected]

各PE 領域内ポリゴン情報を全 rank に送信

**Returns**

POLYLIB\_STAT で定義される値が返る。

### 3.3.3.23 POLYLIB\_STAT PolylibNS::MPIPolylib::send\_polygons\_to\_rank0 ( ) [protected]

rank0 へポリゴン情報を送信

### 3.3.3.24 void PolylibNS::MPIPolylib::show\_group\_name ( PolygonGroup \* p, std::string tab ) [protected]

指定されたグループ以下の階層構造をツリー形式で標準出力に出力する。

**Parameters**

<i>p</i>	表示対象となるグループのポインタ。
<i>tab</i>	階層の深さを示すスペース。

**Attention**

プロセス毎に動作する。出力にランク数に加わる以外は非並列版と同じ。

### 3.3.3.25 unsigned int PolylibNS::MPIPolylib::used\_memory\_size ( )

MPIPolylib が利用中の概算メモリ量を返す

**Returns**

利用中のメモリ量 (byte)

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/MPIPolylib.h



## 3.4 PolylibNS::ParallelInfo Struct Reference

```
#include <MPIPolylib.h>
```

### Public Attributes

- MPI\_Comm [m\\_comm](#)  
MPI コミュニケーター
- int [m\\_rank](#)  
ランク数
- [CalcAreaInfo](#) [m\\_area](#)  
計算領域情報
- std::map< int, std::vector< int > > [m\\_exclusion\\_map](#)  
*migrate* 除外三角形ID マップ (*k*:グループID, *v*:三角形ID リスト)

### 3.4.1 Detailed Description

クラス:[ParallelInfo](#) 並列プロセス情報。

The documentation for this struct was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/MPIPolylib.h

## 3.5 PolylibNS::PolygonGroup Class Reference

```
#include <PolygonGroup.h>
```

### Public Member Functions

- [PolygonGroup](#) ()
- virtual [~PolygonGroup](#) ()
- POLYLIB\_STAT [init](#) (const std::vector< [PrivateTriangle](#) \* > \*tri\_list, bool clear=true)
- virtual POLYLIB\_STAT [build\\_group\\_tree](#) ([Polylib](#) \*polylib, [PolygonGroup](#) \*parent, [TextParser](#) \*tp)
- POLYLIB\_STAT [build\\_polygon\\_tree](#) ()
- POLYLIB\_STAT [load\\_stl\\_file](#) (float scale=1.0)
- POLYLIB\_STAT [load\\_id\\_file](#) (ID\_FORMAT id\_format)
- POLYLIB\_STAT [save\\_stl\\_file](#) (std::string rank\_no, std::string extend, std::string format)
- POLYLIB\_STAT [save\\_stl\\_file](#) (std::string rank\_no, std::string extend, std::string format, std::map< std::string, std::string > &stl\_fname\_map)
- POLYLIB\_STAT [save\\_id\\_file](#) (std::string rank\_no, std::string extend, ID\_FORMAT id\_format)
- virtual POLYLIB\_STAT [mk\\_param\\_tag](#) ([TextParser](#) \*pt, std::string rank\_no, std::string extend, std::string format)
- virtual POLYLIB\_STAT [move](#) ([PolylibMoveParams](#) &params)
- const std::vector< [PrivateTriangle](#) \* > \* [search](#) ([BBox](#) \*bbox, bool every) const
- POLYLIB\_STAT [search](#) ([BBox](#) \*bbox, bool every, std::vector< [PrivateTriangle](#) \* > \*tri\_list) const
- const std::vector< [PrivateTriangle](#) \* > \* [linear\\_search](#) ([BBox](#) \*bbox, bool every) const
- POLYLIB\_STAT [linear\\_search](#) ([BBox](#) \*bbox, bool every, std::vector< [PrivateTriangle](#) \* > \*tri\_list) const
- const [PrivateTriangle](#) \* [search\\_nearest](#) (const [Vec3f](#) &pos) const
- std::string [acq\\_fullpath](#) ()
- std::string [acq\\_file\\_name](#) ()

- const std::vector  
< [PrivateTriangle](#) \* > \* [search\\_outbounded](#) ([BBox](#) neighbour\_bbox, std::vector< int > \*exclude\_tri\_ids)
- POLYLIB\_STAT [add\\_triangles](#) (std::vector< [PrivateTriangle](#) \* > \*tri\_list)
- POLYLIB\_STAT [rebuild\\_polygons](#) ()
- POLYLIB\_STAT [show\\_group\\_info](#) (int irank=-1)
- int [get\\_group\\_num\\_tri](#) (void)  
ポリゴングループの要素数を返す
- float [get\\_group\\_area](#) (void)  
ポリゴンの面積を積算して返す
- POLYLIB\_STAT [rescale\\_polygons](#) (float scale)  
ポリゴンの縮尺変換 & KD 木再構築
- virtual std::string [whoami](#) ()
- void [set\\_file\\_name](#) (std::map< std::string, std::string > fname)
- std::map< std::string,  
std::string > [get\\_file\\_name](#) () const
- void [set\\_name](#) (std::string name)
- std::string [get\\_name](#) (void)
- void [set\\_parent\\_path](#) (std::string ppath)
- std::string [get\\_parent\\_path](#) (void)
- void [set\\_parent](#) ([PolygonGroup](#) \*p)
- [PolygonGroup](#) \* [get\\_parent](#) (void)
- void [set\\_children](#) (std::vector< [PolygonGroup](#) \* > &p)
- std::vector< [PolygonGroup](#) \* > & [get\\_children](#) (void)
- void [add\\_children](#) ([PolygonGroup](#) \*p)
- std::vector< [PrivateTriangle](#) \* > \* [get\\_triangles](#) ()
- [VTree](#) \* [get\\_vtree](#) ()
- int [get\\_internal\\_id](#) ()
- std::string [get\\_label](#) ()
- int [get\\_id](#) ()
- int [get\\_movable](#) ()
- size\_t [get\\_num\\_of\\_tris\\_before\\_move](#) ()

### Static Public Member Functions

- static std::string [get\\_class\\_name](#) ()

### Static Public Attributes

- static const char \* [ATT\\_NAME\\_CLASS](#)

### Protected Member Functions

- POLYLIB\_STAT [setup\\_attribute](#) ([Polylib](#) \*polylib, [PolygonGroup](#) \*parent, [TextParser](#) \*tp)
- POLYLIB\_STAT [init\\_check\\_leaped](#) ()
- POLYLIB\_STAT [check\\_leaped](#) ([Vec3f](#) origin, [Vec3f](#) cell\_size)
- bool [is\\_far](#) ([Vec3f](#) origin, [Vec3f](#) cell\_size, [Vec3f](#) pos1, [Vec3f](#) pos2)

## Protected Attributes

- int `m_internal_id`  
グループID。
- std::string `m_name`  
自グループ名。
- std::string `m_parent_path`  
親グループのパス名。
- `PolygonGroup` \* `m_parent`  
親グループへのポインタ。
- std::vector< `PolygonGroup` \* > `m_children`  
子グループへのポインタリスト。
- std::map< std::string, std::string > `m_file_name`  
STL ファイル名とファイル形式。
- `Polygons` \* `m_polygons`  
三角形 `Polygons` クラス。
- bool `m_movable`  
`move` メソッドにより移動するグループか？
- bool `m_need_rebuild`  
KD 木の再構築が必要か？
- std::vector< `PrivateTriangle` \* > \* `m_trias_before_move`  
`move()` による移動前三角形一時保存リスト。
- std::string `m_label`  
ユーザ定義ラベル : (追加 2012.08.31)

### 3.5.1 Detailed Description

クラス: `PolygonGroup` ポリゴングループを管理するクラスです。

### 3.5.2 Constructor & Destructor Documentation

#### 3.5.2.1 PolylibNS::PolygonGroup::PolygonGroup ( )

コンストラクタ

#### 3.5.2.2 virtual PolylibNS::PolygonGroup::~PolygonGroup ( ) [virtual]

デストラクタ

### 3.5.3 Member Function Documentation

#### 3.5.3.1 std::string PolylibNS::PolygonGroup::acq\_file\_name ( )

カンマ区切りでSTL ファイル名リストを取得。

#### Returns

ファイル名リスト。

### 3.5.3.2 std::string PolylibNS::PolygonGroup::acq\_fullpath ( )

PolygonGroup のフルパス名を取得する。

#### Returns

フルパス名。

### 3.5.3.3 void PolylibNS::PolygonGroup::add\_children ( PolygonGroup \* p ) [inline]

子グループを追加。

#### Parameters

in	<i>p</i>	子グループ。
----	----------	--------

### 3.5.3.4 POLYLIB\_STAT PolylibNS::PolygonGroup::add\_triangles ( std::vector< PrivateTriangle \* > \* tri\_list )

三角形リストの追加。

#### Parameters

in	<i>tri_list</i>	三角形ポリゴンリストのポインタ。
----	-----------------	------------------

#### Returns

POLYLIB\_STAT で定義される値が返る。

#### Attention

三角形ID が重複した三角形は追加しない。KD 木の再構築はしない。

### 3.5.3.5 virtual POLYLIB\_STAT PolylibNS::PolygonGroup::build\_group\_tree ( Polylib \* polylib, PolygonGroup \* parent, TextParser \* tp ) [virtual]

PolygonGroup ツリーの作成。 設定ファイルの内容を再帰的に呼び出し、PolygonGroup ツリーを作成する。

#### Parameters

in	<i>polylib</i>	Polygon クラスのインスタンス
in	<i>parent</i>	親グループ
in	<i>tp</i>	TextParser のインスタンス

#### Returns

POLYLIB\_STAT で定義される値が返る。

### 3.5.3.6 POLYLIB\_STAT PolylibNS::PolygonGroup::build\_polygon\_tree ( )

三角形ポリゴンの法線ベクトルの計算、面積の計算、KD 木の生成を行う。 三角形ポリゴンはTriMesh クラスが管理している。

#### Returns

POLYLIB\_STAT で定義される値が返る。

**Attention**

TriMesh クラスの build() 参照。

**3.5.3.7 POLYLIB\_STAT PolylibNS::PolygonGroup::check\_leaped ( Vec3f *origin*, Vec3f *cell\_size* ) [protected]**

[move\(\)](#) メソッド実行により、頂点が隣接セルよりも遠くへ移動した三角形情報を報告（後処理）。該当する三角形について、以下の情報を cerr へ出力する。・ポリゴングループID・三角形ID・移動前/後の頂点座標

**Parameters**

in	<i>origin</i>	計算領域起点座標
in	<i>cell_size</i>	ボクセルサイズ

**Returns**

POLYLIB\_STAT で定義される値が返る。

**Attention**

本メソッドはデバッグ用です。派生クラスでオーバーライドした move() メソッド内で、座標移動 処理後に呼ぶこと。

**3.5.3.8 std::vector<PolygonGroup\*> &PolylibNS::PolygonGroup::get\_children ( void ) [inline]**

子グループを取得。

**Returns**

子グループのリスト。

**3.5.3.9 static std::string PolylibNS::PolygonGroup::get\_class\_name ( ) [inline],[static]**

クラス名を取得。

**Returns**

クラス名。

**Attention**

本クラスを継承する場合、継承後のクラス名を返すように変更すること。

**3.5.3.10 std::map<std::string, std::string> PolylibNS::PolygonGroup::get\_file\_name ( ) const [inline]**

STL ファイル名とファイルフォーマットの対応マップ取得。

**Returns**

STL ファイル名とファイルフォーマットの対応マップ。

### 3.5.3.11 int PolylibNS::PolygonGroup::get\_id ( ) [inline]

ユーザ定義ID を取得。追加 2010.10.20

#### Returns

ユーザ定義ID。

### 3.5.3.12 int PolylibNS::PolygonGroup::get\_internal\_id ( ) [inline]

ポリゴングループID を取得。メンバー名修正 ( m\_id -> m\_internal\_id) 2010.10.20

#### Returns

ポリゴングループID。

### 3.5.3.13 std::string PolylibNS::PolygonGroup::get\_label ( ) [inline]

ユーザ定義ラベルを取得。追加 2012.08.31

#### Returns

ユーザ定義ラベル。

### 3.5.3.14 int PolylibNS::PolygonGroup::get\_movable ( ) [inline]

移動対象フラグを取得。

#### Returns

移動対象フラグ。

### 3.5.3.15 std::string PolylibNS::PolygonGroup::get\_name ( void ) [inline]

グループ名を取得。

#### Returns

グループ名。

### 3.5.3.16 size\_t PolylibNS::PolygonGroup::get\_num\_of\_trias\_before\_move ( ) [inline]

[move\(\)](#)による移動前三角形一時保存リストの個数を取得。

#### Returns

一時保存リストサイズ。

### 3.5.3.17 PolygonGroup\* PolylibNS::PolygonGroup::get\_parent ( void ) [inline]

親グループを取得。

#### Returns

親グループのポインタ。

**3.5.3.18** `std::string PolylibNS::PolygonGroup::get_parent_path ( void ) [inline]`

親グループのフルパス名を取得。

#### Returns

親グループのフルパス名。

**3.5.3.19** `std::vector<PrivateTriangle*>* PolylibNS::PolygonGroup::get_triangles ( ) [inline]`

Polygon クラスが管理する三角形ポリゴンリストを取得。

#### Returns

三角形ポリゴンリスト。

**3.5.3.20** `VTree* PolylibNS::PolygonGroup::get_vtree ( ) [inline]`

Polygon クラスが管理するKD 木クラスを取得。

#### Returns

KD 木ポリゴンリスト。

**3.5.3.21** `POLYLIB_STAT PolylibNS::PolygonGroup::init ( const std::vector< PrivateTriangle * > * tri_list, bool clear = true )`

引数で与えられる三角形ポリゴンリストを複製し、KD 木の生成を行う。

#### Parameters

in	<i>tri_list</i>	設定する三角形ポリゴンリスト。
in	<i>clear</i>	true:ポリゴン複製、面積計算、KD 木生成を行う。false:面積計算、KD 木生成だけを行う。

#### Returns

POLYLIB\_STAT で定義される値が返る。

#### Attention

TriMesh クラスの init() 参照。オーバーロードメソッドあり。

**3.5.3.22** `POLYLIB_STAT PolylibNS::PolygonGroup::init_check_leaped ( ) [protected]`

[move\(\)](#) メソッド実行により、頂点が隣接セルよりも遠くへ移動した三角形情報を報告（前処理）。

#### Returns

POLYLIB\_STAT で定義される値が返る。

#### Attention

本メソッドはデバッグ用です。派生クラスでオーバーライドした [move\(\)](#) メソッド内で、座標移動 処理前に呼ぶこと。

**3.5.3.23** `bool PolylibNS::PolygonGroup::is_far ( Vec3f origin, Vec3f cell_size, Vec3f pos1, Vec3f pos2 )`  
[protected]

2 点が隣接ボクセルよりも離れているか？



## Parameters

in	<i>origin</i>	計算領域起点座標。
in	<i>cell_size</i>	ボクセルサイズ。
in	<i>pos1</i>	点 (1)。
in	<i>pos2</i>	点 (2)。

## Returns

true:2 点が隣接ボクセルよりも離れている。

**3.5.3.24** `const std::vector<PrivateTriangle*>* PolylibNS::PolygonGroup::linear_search ( BBox * bbox, bool every ) const`

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

## Returns

抽出したポリゴンリストのポインタ。

## Attention

オーバーロードメソッドあり。

**3.5.3.25** `POLYLIB_STAT PolylibNS::PolygonGroup::linear_search ( BBox * bbox, bool every, std::vector< PrivateTriangle * > * tri_list ) const`

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in, out	<i>tri_list</i>	抽出した三角形ポリゴンリストのポインタ。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

オーバーロードメソッドあり。

**3.5.3.26** `POLYLIB_STAT PolylibNS::PolygonGroup::load_id_file ( ID_FORMAT id_format )`

三角形ポリゴンID ファイルからポリゴンID を読み込み、m\_internal\_id に登録する。

## Parameters

<i>in</i>	<i>id_format</i>	三角形ID ファイルの入力形式。
-----------	------------------	------------------

## Returns

POLYLIB\_STAT で定義される値が返る。

3.5.3.27 POLYLIB\_STAT PolylibNS::PolygonGroup::load\_stl\_file ( float *scale* = 1.0 )

STL ファイルからポリゴン情報を読み込み、TriMesh クラスに登録する。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

TriMesh クラスの import() 参照。

3.5.3.28 virtual POLYLIB\_STAT PolylibNS::PolygonGroup::mk\_param\_tag ( TextParser \* *pt*, std::string *rank\_no*, std::string *extend*, std::string *format* ) [virtual]

設定ファイルに出力するTextParser のリーフを編集する. デフォルトでは何もしない。 CarGroup.cxx の例を参照。

## Parameters

<i>in</i>	<i>pointer</i>	to TextParser
<i>in</i>	<i>rank_no</i>	ファイル名に付加するランク番号。
<i>in</i>	<i>extend</i>	ファイル名に付加する自由文字列。
<i>in</i>	<i>format</i>	STL ファイルフォーマット。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

do nothing by default

3.5.3.29 virtual POLYLIB\_STAT PolylibNS::PolygonGroup::move ( PolylibMoveParams & *params* ) [virtual]

三角形ポリゴン移動メソッド。virtual 用の関数なので処理はない。

## Parameters

<i>in</i>	<i>params</i>	Polylib.h で宣言しているパラメタセットクラス。
-----------	---------------	------------------------------

## Returns

POLYLIB\_STAT で定義される値が返る。

**3.5.3.30 POLYLIB\_STAT PolylibNS::PolygonGroup::rebuild\_polygons ( )**

ポリゴン情報を再構築する。(KD 木の再構築をおこなう)

**Returns**

POLYLIB\_STAT で定義される値が返る。

**3.5.3.31 POLYLIB\_STAT PolylibNS::PolygonGroup::save\_id\_file ( std::string *rank\_no*, std::string *extend*, ID\_FORMAT *id\_format* )**

三角形ポリゴンID ファイルにポリゴンID を出力する。ID ファイル名は、階層化されたグループ名\_ランク番号\_自由文字列.id。

**Parameters**

in	<i>rank_no</i>	ファイル名に付加するランク番号。
in	<i>extend</i>	ファイル名に付加する自由文字列。
in	<i>id_format</i>	三角形ID ファイルの出力形式。

**Returns**

POLYLIB\_STAT で定義される値が返る。

**3.5.3.32 POLYLIB\_STAT PolylibNS::PolygonGroup::save\_stl\_file ( std::string *rank\_no*, std::string *extend*, std::string *format* )**

TriMesh クラスが管理しているポリゴン情報をSTL ファイルに出力する。

**Parameters**

in	<i>rank_no</i>	ファイル名に付加するランク番号。
in	<i>extend</i>	ファイル名に付加する自由文字列。
in	<i>format</i>	STL ファイルフォーマット。

**Returns**

POLYLIB\_STAT で定義される値が返る。

**Attention**

TriMeshIO クラスの save() 参照。オーバーロードメソッドあり。

**3.5.3.33 POLYLIB\_STAT PolylibNS::PolygonGroup::save\_stl\_file ( std::string *rank\_no*, std::string *extend*, std::string *format*, std::map< std::string, std::string > & *stl\_fname\_map* )**

TriMesh クラスが管理しているポリゴン情報をSTL ファイルに出力する。TextParser 対応版

**Parameters**

in	<i>rank_no</i>	ファイル名に付加するランク番号。
in	<i>extend</i>	ファイル名に付加する自由文字列。

<i>in</i>	<i>format</i>	STL ファイルフォーマット。
<i>in, out</i>	<i>stl_fname_map</i>	stl ファイル名とポリゴングループのパス

**Returns**

POLYLIB\_STAT で定義される値が返る。

**Attention**

TriMeshIO クラスの `save()` 参照。オーバーロードメソッドあり。

### 3.5.3.34 `const std::vector<PrivateTriangle*>* PolylibNS::PolygonGroup::search ( BBox * bbox, bool every ) const`

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

**Parameters**

<i>in</i>	<i>bbox</i>	矩形領域。
<i>in</i>	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

**Returns**

抽出したポリゴンリストのポインタ。

**Attention**

オーバーロードメソッドあり。

### 3.5.3.35 `POLYLIB_STAT PolylibNS::PolygonGroup::search ( BBox * bbox, bool every, std::vector< PrivateTriangle * > * tri_list ) const`

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

**Parameters**

<i>in</i>	<i>bbox</i>	矩形領域。
<i>in</i>	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
<i>in, out</i>	<i>tri_list</i>	抽出した三角形ポリゴンリストのポインタ。

**Returns**

POLYLIB\_STAT で定義される値が返る。

**Attention**

オーバーロードメソッドあり。

### 3.5.3.36 `const PrivateTriangle* PolylibNS::PolygonGroup::search_nearest ( const Vec3f & pos ) const`

KD 木探索により、指定位置に最も近いポリゴンを検索する。

## Parameters

<i>in</i>	<i>pos</i>	指定位置
-----------	------------	------

## Returns

検索されたポリゴン

**3.5.3.37** `const std::vector<PrivateTriangle*>* PolylibNS::PolygonGroup::search_outbounded ( BBox neighbour_bbox, std::vector< int > * exclude_tria_ids )`

PE 領域間移動する三角形ポリゴンリストの取得。

## Parameters

<i>in</i>	<i>neighbour_bbox</i>	隣接PE 領域バウンディングボックス。
<i>in</i>	<i>exclude_tria_ids</i>	領域移動対象外三角形ID リスト。

## Returns

検索結果三角形リスト。

**3.5.3.38** `void PolylibNS::PolygonGroup::set_children ( std::vector< PolygonGroup * > &p ) [inline]`

子グループを設定。

## Parameters

<i>in</i>	<i>p</i>	子グループのリスト。
-----------	----------	------------

**3.5.3.39** `void PolylibNS::PolygonGroup::set_file_name ( std::map< std::string, std::string > fname ) [inline]`

STL ファイル名とファイルフォーマットを設定。

## Parameters

<i>in</i>	<i>fname</i>	STL ファイル名とファイルフォーマットの対応マップ。
-----------	--------------	-----------------------------

**3.5.3.40** `void PolylibNS::PolygonGroup::set_name ( std::string name ) [inline]`

グループ名を設定。

## Parameters

<i>in</i>	<i>name</i>	グループ名。
-----------	-------------	--------

**3.5.3.41** `void PolylibNS::PolygonGroup::set_parent ( PolygonGroup * p ) [inline]`

親グループを設定。

## Parameters

in	<i>p</i>	親グループのポインタ。
----	----------	-------------

### 3.5.3.42 void PolylibNS::PolygonGroup::set\_parent\_path ( std::string *ppath* ) [inline]

親グループのフルパス名を設定。

#### Parameters

in	<i>ppath</i>	親グループのフルパス名。
----	--------------	--------------

### 3.5.3.43 POLYLIB\_STAT PolylibNS::PolygonGroup::setup\_attribute ( Polylib \* *polylib*, PolygonGroup \* *parent*, TextParser \* *tp* ) [protected]

設定ファイルから取得したPolygonGroup の情報をインスタンスにセットする。

"filepath" に関して、先に filepath が複数 (filepath[0]) が存在するかどうかをチェックして、複数ならばその処理を行い、filepath の処理は終了する。複数でないことが分かったら、filepath が単体で存在するかをチェックして、存在するならば、処理を行う。

#### Parameters

in	<i>polylib</i>	Polygon クラスのインスタンス。
in	<i>parent</i>	親グループ。
in	<i>tp</i>	TextParser クラスのインスタンス

#### Returns

POLYLIB\_STAT で定義される値が返る。

### 3.5.3.44 POLYLIB\_STAT PolylibNS::PolygonGroup::show\_group\_info ( int *irank* = -1 )

グループ情報（ランク番号、親グループ名、自分のグループ名、ファイル名、頂点数、各頂点のXYZ 座標値、法線ベクトルのXYZ 座標値、面積）を出力する。

#### Parameters

in	<i>irank</i>	ランク数。
----	--------------	-------

#### Returns

POLYLIB\_STAT で定義される値が返る。

### 3.5.3.45 virtual std::string PolylibNS::PolygonGroup::whoami ( ) [inline],[virtual]

クラス名を取得。

#### Returns

クラス名。

#### Attention

継承するクラスのクラス名取得関数 get\_class\_name() を呼び出す。

### 3.5.4 Member Data Documentation

#### 3.5.4.1 const char\* PolylibNS::PolygonGroup::ATT\_NAME\_CLASS [static]

config ファイルに記述するParam タグのクラス名 (value="...")。

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/groups/PolygonGroup.h

## 3.6 PolylibNS::PolygonGroupFactory Class Reference

```
#include <PolygonGroupFactory.h>
```

### Public Member Functions

- [PolygonGroupFactory](#) ()
- virtual [~PolygonGroupFactory](#) ()
- virtual [PolygonGroup](#) \* [create\\_instance](#) (std::string class\_name)

### 3.6.1 Detailed Description

クラス:[PolygonGroupFactory](#)

### 3.6.2 Constructor & Destructor Documentation

#### 3.6.2.1 PolylibNS::PolygonGroupFactory::PolygonGroupFactory ( ) [inline]

コンストラクタ。

#### 3.6.2.2 virtual PolylibNS::PolygonGroupFactory::~~PolygonGroupFactory ( ) [inline],[virtual]

デストラクタ。

### 3.6.3 Member Function Documentation

#### 3.6.3.1 virtual PolygonGroup\* PolylibNS::PolygonGroupFactory::create\_instance ( std::string class\_name ) [inline],[virtual]

インスタンス作成。

#### Parameters

in	<i>class_name</i>	作成するクラス名。
----	-------------------	-----------

#### Returns

作成に失敗した場合はNULL が返る。

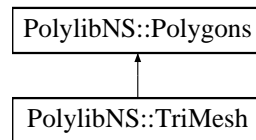
The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/groups/PolygonGroupFactory.h

### 3.7 PolylibNS::Polygons Class Reference

```
#include <Polygons.h>
```

Inheritance diagram for PolylibNS::Polygons:



#### Public Member Functions

- [Polygons](#) ()
- virtual [~Polygons](#) ()=0
- virtual void [init](#) (const std::vector< [PrivateTriangle](#) \* > \*trias)=0
- virtual void [add](#) (const std::vector< [PrivateTriangle](#) \* > \*trias)=0
- virtual POLYLIB\_STAT [import](#) (const std::map< std::string, std::string > fname, float scale=1.0)=0
- virtual POLYLIB\_STAT [build](#) ()=0
- virtual int [triangles\\_num](#) ()=0
- virtual const std::vector< [PrivateTriangle](#) \* > \* [search](#) (BBox \*bbox, bool every) const =0
- virtual POLYLIB\_STAT [search](#) (BBox \*bbox, bool every, std::vector< [PrivateTriangle](#) \* > \*tri\_list) const =0
- virtual const std::vector< [PrivateTriangle](#) \* > \* [linear\\_search](#) (BBox \*bbox, bool every) const =0
- virtual POLYLIB\_STAT [linear\\_search](#) (BBox \*bbox, bool every, std::vector< [PrivateTriangle](#) \* > \*tri\_list) const =0
- virtual const [PrivateTriangle](#) \* [search\\_nearest](#) (const [Vec3f](#) &pos) const =0
- virtual POLYLIB\_STAT [set\\_all\\_exid](#) (const int id) const =0
- std::vector< [PrivateTriangle](#) \* > \* [get\\_tri\\_list](#) () const
- virtual [VTree](#) \* [get\\_vtree](#) () const =0

#### Protected Attributes

- std::vector< [PrivateTriangle](#) \* > \* [m\\_tri\\_list](#)  
三角形ポリゴンのリスト。

#### 3.7.1 Detailed Description

クラス:[Polygons](#) 三角形ポリゴン集合を管理する純粋仮想クラスです。

#### 3.7.2 Constructor & Destructor Documentation

##### 3.7.2.1 PolylibNS::Polygons::Polygons ( ) [inline]

コンストラクタ。

##### 3.7.2.2 virtual PolylibNS::Polygons::~~Polygons ( ) [pure virtual]

デストラクタ。



### 3.7.3 Member Function Documentation

**3.7.3.1** `virtual void PolylibNS::Polygons::add ( const std::vector< PrivateTriangle * > * trias ) [pure virtual]`

三角形ポリゴンリストに引数で与えられる三角形を追加する。

Parameters

in	<i>trias</i>	設定する三角形ポリゴンリスト。
----	--------------	-----------------

Implemented in [PolylibNS::TriMesh](#).

**3.7.3.2** `virtual POLYLIB_STAT PolylibNS::Polygons::build ( ) [pure virtual]`

Polygons クラスに含まれる全ポリゴン情報からKD 木を作成する。

Returns

POLYLIB\_STAT で定義される値が返る。

Implemented in [PolylibNS::TriMesh](#).

**3.7.3.3** `std::vector<PrivateTriangle*>* PolylibNS::Polygons::get_tri_list ( ) const [inline]`

三角形ポリゴンのリストを取得。

Returns

三角形ポリゴンのリスト。

**3.7.3.4** `virtual VTree* PolylibNS::Polygons::get_vtree ( ) const [pure virtual]`

KD 木クラスを取得。

Returns

KD 木クラス。

Implemented in [PolylibNS::TriMesh](#).

**3.7.3.5** `virtual POLYLIB_STAT PolylibNS::Polygons::import ( const std::map< std::string, std::string > fname, float scale = 1.0 ) [pure virtual]`

STL ファイルを読み込みデータの初期化。

Parameters

in	<i>fname</i>	ファイル名とファイルフォーマットの map。
----	--------------	------------------------

Returns

POLYLIB\_STAT で定義される値が返る。

Implemented in [PolylibNS::TriMesh](#).

**3.7.3.6** `virtual void PolylibNS::Polygons::init ( const std::vector< PrivateTriangle * > * trias ) [pure virtual]`

引数で与えられる三角形ポリゴンリストの複製を設定する。

## Parameters

in	<i>trias</i>	設定する三角形ポリゴンリスト。
----	--------------	-----------------

## Attention

オーバーロードメソッドあり。

Implemented in [PolylibNS::TriMesh](#).

3.7.3.7 `virtual const std::vector<PrivateTriangle*>* PolylibNS::Polygons::linear_search ( BBox * bbox, bool every ) const` [pure virtual]

線形探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

## Returns

抽出したポリゴンリストのポインタ。

## Attention

MPIPolylib 内でのみ利用するため、ユーザは使用しないで下さい。  
オーバーロードメソッドあり。

Implemented in [PolylibNS::TriMesh](#).

3.7.3.8 `virtual POLYLIB_STAT PolylibNS::Polygons::linear_search ( BBox * bbox, bool every, std::vector< PrivateTriangle * > * tri_list ) const` [pure virtual]

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in, out	<i>tri_list</i>	抽出した三角形ポリゴンリストのポインタ。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

オーバーロードメソッドあり。

Implemented in [PolylibNS::TriMesh](#).

3.7.3.9 `virtual const std::vector<PrivateTriangle*>* PolylibNS::Polygons::search ( BBox * bbox, bool every ) const` [pure virtual]

KD 木探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

## Returns

抽出したポリゴンリストのポインタ。

## Attention

MPIPolylib 内でのみ利用するため、ユーザは使用しないで下さい。  
オーバーロードメソッドあり。

Implemented in [PolylibNS::TriMesh](#).

```
3.7.3.10 virtual POLYLIB_STAT PolylibNS::Polygons::search ( BBox * bbox, bool every, std::vector< PrivateTriangle * >
* tri_list ) const [pure virtual]
```

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in, out	<i>tri_list</i>	抽出した三角形ポリゴンリストへのポインタ。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

オーバーロードメソッドあり。

Implemented in [PolylibNS::TriMesh](#).

```
3.7.3.11 virtual const PrivateTriangle* PolylibNS::Polygons::search_nearest ( const Vec3f & pos ) const [pure
virtual]
```

KD 木探索により、指定位置に最も近いポリゴンを検索する。

## Parameters

in	<i>pos</i>	指定位置
----	------------	------

## Returns

検索されたポリゴン

Implemented in [PolylibNS::TriMesh](#).

```
3.7.3.12 virtual POLYLIB_STAT PolylibNS::Polygons::set_all_exid ( const int id ) const [pure virtual]
```

配下の全ポリゴンの m\_exid 値を指定値にする。

## Parameters

<code>in</code>	<code>id</code>	指定値
-----------------	-----------------	-----

Implemented in [PolylibNS::TriMesh](#).

### 3.7.3.13 virtual int PolylibNS::Polygons::triangles\_num ( ) [pure virtual]

Polygons クラスで保持する三角形ポリゴンの総数を返す。

## Returns

三角形ポリゴンの総数。

Implemented in [PolylibNS::TriMesh](#).

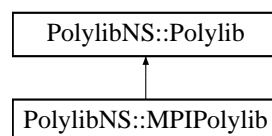
The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/polygons/Polygons.h

## 3.8 PolylibNS::Polylib Class Reference

```
#include <Polylib.h>
```

Inheritance diagram for PolylibNS::Polylib:



### Public Member Functions

- void [set\\_factory](#) ([PolygonGroupFactory](#) \*factory=NULL)
- POLYLIB\_STAT [load](#) (std::string config\_name="polylib\_config.tpp", float scale=1.0)
- POLYLIB\_STAT [save](#) (std::string \*p\_config\_name, std::string stl\_format, std::string extend="")
- POLYLIB\_STAT [move](#) ([PolylibMoveParams](#) &params)
- std::vector< [PolygonGroup](#) \* > \* [get\\_root\\_groups](#) () const
- std::vector< [Triangle](#) \* > \* [search\\_polygons](#) (std::string group\_name, [Vec3f](#) min\_pos, [Vec3f](#) max\_pos, bool every) const
- const [Triangle](#) \* [search\\_nearest\\_polygon](#) (std::string group\_name, const [Vec3f](#) &pos) const
- POLYLIB\_STAT [check\\_group\\_name](#) (const std::string &pg\_name, const std::string &parent\_path)
- [PolygonGroup](#) \* [create\\_polygon\\_group](#) (std::string class\_name)
- void [add\\_pg\\_list](#) ([PolygonGroup](#) \*pg)
- void [show\\_group\\_hierarchy](#) (FILE \*fp=NULL)
- POLYLIB\_STAT [show\\_group\\_info](#) (std::string group\_name)
- unsigned int [used\\_memory\\_size](#) ()
- [PolygonGroup](#) \* [get\\_group](#) (std::string name) const

### Static Public Member Functions

- static [Polylib](#) \* [get\\_instance](#) ()

## Protected Member Functions

- [Polylib](#) ()
- [~Polylib](#) ()
- POLYLIB\_STAT [make\\_group\\_tree](#) (TextParser \*tp\_ptr)
- POLYLIB\_STAT [make\\_group\\_tree](#) (std::string config\_contents)
- POLYLIB\_STAT [load\\_config\\_file](#) (std::string \*contents, std::string fname="")
- POLYLIB\_STAT [load\\_with\\_idfile](#) (std::string config\_name, ID\_FORMAT id\_format, float scale=1.0)
- POLYLIB\_STAT [load\\_polygons](#) (bool with\_id\_file, ID\_FORMAT id\_format, float scale=1.0)
- char \* [save\\_config\\_file](#) (std::string rank\_no, std::string extend, std::string format)
- POLYLIB\_STAT [clearfilepath](#) (TextParser \*tp\_ptr)
- POLYLIB\_STAT [setfilepath](#) (std::map< std::string, std::string > &stl\_fname\_map)
- char \* [polylib\\_config\\_save\\_file](#) (std::string rank\_no, std::string extend)  
     設定ファイルの保存。 *PolylibConfig* 内部にあったものをここへ。
- POLYLIB\_STAT [save\\_with\\_rankno](#) (std::string \*p\_config\_name, int myrank, int maxrank, std::string extend, std::string stl\_format, ID\_FORMAT id\_format)
- void [show\\_group\\_name](#) (PolygonGroup \*p, std::string tab, FILE \*fp)
- PolygonGroup \* [get\\_group](#) (int internal\_id) const

## Protected Attributes

- PolygonGroupFactory \* [m\\_factory](#)  
     *PolygonGroup* のファクトリークラス
- std::vector< [PolygonGroup](#) \* > [m\\_pg\\_list](#)  
     ポリゴングループリスト
- TextParser \* [tp](#)

## Static Protected Attributes

- static [Polylib](#) \* [m\\_instance](#)  
     自クラスのインスタンス (*singleton*)

### 3.8.1 Detailed Description

クラス:[Polylib](#) ポリゴンを管理する為のクラスライブラリです。

### 3.8.2 Constructor & Destructor Documentation

#### 3.8.2.1 PolylibNS::Polylib::Polylib ( ) [protected]

コンストラクタ

#### Attention

*singleton* のため、子クラス以外からの呼び出し不可とする

#### 3.8.2.2 PolylibNS::Polylib::~~Polylib ( ) [protected]

デストラクタ

### 3.8.3 Member Function Documentation

#### 3.8.3.1 void PolylibNS::Polylib::add\_pg\_list ( PolygonGroup \* *pg* )

PolygonGroup の追加。本クラスが管理している PolygonGroup のリストに PolygonGroup を追加する。

## Parameters

in	<i>pg</i>	<a href="#">PolygonGroup</a>
----	-----------	------------------------------

## Attention

Polylib 内部で使用する関数であり、通常は利用者が用いるものではない。

3.8.3.2 POLYLIB\_STAT PolylibNS::Polylib::check\_group\_name ( const std::string & *pg\_name*, const std::string & *parent\_path* )

引数のグループ名が既存グループと重複しないかチェック。

## Parameters

in	<i>pg_name</i>	グループ名
in	<i>parent_path</i>	親グループまでのフルパス

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

Polylib 内部で使用する関数であり、通常は利用者が用いるものではない。

3.8.3.3 POLYLIB\_STAT PolylibNS::Polylib::clearfilepath ( TextParser \* *tp\_ptr* ) [protected]

TextParser 内部データから "filepath" "filepath[\*]" というリーフをすべて削除する。

recursive の動作の為、引数に *tp\_ptr* が必要

## Parameters

in	<i>tp_ptr</i>	TextParser へのポインタ。
----	---------------	--------------------

## Returns

POLYLIB\_STAT で定義される値が返る。

3.8.3.4 PolygonGroup\* PolylibNS::Polylib::create\_polygon\_group ( std::string *class\_name* )

PolygonGroup のインスタンスの生成。本クラスが管理しているFactory クラスを利用して、引数で渡されたクラス名に応じたPolygonGroup のインスタンスを生成する。

## Parameters

in	<i>class_name</i>	クラス名
----	-------------------	------

## Returns

生成したPolygonGroup

## Attention

Polylib 内部で使用する関数であり、通常は利用者が用いるものではない。

3.8.3.5 PolygonGroup\* PolylibNS::Polylib::get\_group ( std::string *name* ) const

グループの取得。 *name* で与えられた名前のPolygonGroup を返す。

## Parameters

<i>in</i>	<i>name</i>	グループ名
-----------	-------------	-------

## Returns

ポリゴングループクラスのポインタ。エラー時はNULL が返る。

## Attention

オーバーロードメソッドあり。

### 3.8.3.6 PolygonGroup\* PolylibNS::Polylib::get\_group ( int *internal\_id* ) const [protected]

グループの取得。 *internal\_id* で与えられた *m\_internal\_id* を持つ PolygonGroup を返す。

## Parameters

<i>in</i>	<i>internal_id</i>	ポリゴングループID
-----------	--------------------	------------

## Returns

ポリゴングループクラスのポインタ。エラー時はNULL が返る。

## Attention

オーバーロードメソッドあり。

### 3.8.3.7 static Polylib\* PolylibNS::Polylib::get\_instance ( ) [static]

singleton の Polylib インスタンス取得。デフォルトの Factory クラスである PolygonGroupFactory を使用してインスタンスを生成する。

## Returns

Polylib クラスのインスタンス。

## Attention

呼び出し側で delete はできません。

### 3.8.3.8 std::vector<PolygonGroup\*> PolylibNS::Polylib::get\_root\_groups ( ) const

PolygoGroup ツリーの最上位ノードの取得。

## Returns

最上位ノードの vector。

## Attention

返却した PolygonGroup は、削除不可。vector は要削除。

### 3.8.3.9 POLYLIB\_STAT PolylibNS::Polylib::load ( std::string *config\_name* = "polylib\_config.tpp", float *scale* = 1.0 )

PolygoGroup、三角形ポリゴン情報の読み込み。引数で指定された設定ファイル (TextParser 形式) を読み込み、グループツリーを作成する。続いて設定ファイルで指定された STL ファイルを読み込み、KD 木を作成する。



## Parameters

in	<i>config_name</i>	設定ファイル名。
----	--------------------	----------

## Returns

POLYLIB\_STAT で定義される値が返る。

### 3.8.3.10 POLYLIB\_STAT PolylibNS::Polylib::load\_config\_file ( std::string \* contents, std::string fname = " " ) [protected]

設定ファイルを読み込み、内容を contents に設定。

## Parameters

out	<i>contents</i>	設定ファイルの内容 (XML 形式)。
in	<i>fname</i>	設定ファイル名。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

MPIPolylib クラスがMPI 環境で利用することを想定している。

### 3.8.3.11 POLYLIB\_STAT PolylibNS::Polylib::load\_polygons ( bool with\_id\_file, ID\_FORMAT id\_format, float scale = 1.0 ) [protected]

STL ファイルの読み込み。グループツリーの全リーフについて、設定されているSTL ファイルから ポリゴン情報を読み込む。読み込んだ後、KD 木の生成、法線の計算、面積の 計算を行う。

## Parameters

in	<i>with_id_file</i>	true ならば、三角形ポリゴンID ファイルを読み込んで m_id を設定する。 false ならば、STL 読み込み時に m_id を自動生成。
in	<i>id_format</i>	三角形ID ファイルの入力形式。

## Returns

POLYLIB\_STAT で定義される値が返る。

### 3.8.3.12 POLYLIB\_STAT PolylibNS::Polylib::load\_with\_idfile ( std::string config\_name, ID\_FORMAT id\_format, float scale = 1.0 ) [protected]

三角形ID ファイルの存在が必須な load 関数。load と同様の動作を行う。但し読み込み時には、三角形ID ファイルが必要であり、このファイルに記述されているID を用いて m\_id を設定する。

## Parameters

in	<i>config_name</i>	設定ファイル名。
in	<i>id_format</i>	三角形ID ファイルの入力形式。

## Returns

POLYLIB\_STAT で定義される値が返る。

**Attention**

MPIPolylib クラスがMPI 環境で利用することを想定している。

**3.8.3.13 POLYLIB\_STAT PolylibNS::Polylib::make\_group\_tree ( TextParser \* *tp\_ptr* ) [protected]**

グループツリー作成。TextParser クラスを使い、PolygonGroup を作成し、グループツリーに登録する。

**Parameters**

<i>in</i>	<i>TextParser</i>	のインスタンス
-----------	-------------------	---------

**Returns**

POLYLIB\_STAT で定義される値が返る。

**Attention**

オーバーロードメソッドあり。

**3.8.3.14 POLYLIB\_STAT PolylibNS::Polylib::make\_group\_tree ( std::string *config\_contents* ) [protected]**

引数の内容でグループ階層構造を構築。

**Parameters**

<i>in</i>	<i>config_contents</i>	設定ファイルの内容 (XML 形式)。
-----------	------------------------	---------------------

**Returns**

POLYLIB\_STAT で定義される値が返る。

**Attention**

MPIPolylib クラスがMPI 環境で利用することを想定している。  
オーバーロードメソッドあり。

**3.8.3.15 POLYLIB\_STAT PolylibNS::Polylib::move ( PolylibMoveParams & *params* )**

三角形ポリゴン座標の移動。本クラスインスタンス配下の全PolygonGroup の move メソッドが呼び出される。move メソッドは、PolygonGroup クラスを拡張したクラスに利用者が記述する。

**Parameters**

<i>in</i>	<i>params</i>	Polylib.h で宣言された移動計算パラメータセット。
-----------	---------------	-------------------------------

**Returns**

POLYLIB\_STAT で定義される値が返る。

**3.8.3.16 char\* PolylibNS::Polylib::polylib\_config\_save\_file ( std::string *rank\_no*, std::string *extend* ) [protected]**

設定ファイルの保存。PolylibConfig 内部にあったものをここへ。

## Parameters

in	<i>rank_no</i>	ランク番号
in	<i>extend</i>	ファイル名に付加する文字列
in	<i>format</i>	TriMeshIO クラスで定義されている STL ファイルのフォーマット。

## Returns

作成した設定ファイルの名称。エラー時はNULL が返る。

**3.8.3.17** POLYLIB\_STAT PolylibNS::Polylib::save ( std::string \* *p\_config\_name*, std::string *stl\_format*, std::string *extend* = " " )

PolygoGroup ツリー、三角形ポリゴン情報の保存。 グループツリーの情報を設定ファイルへ出力。三角形ポリゴン情報を STL ファイルへ出力。

## Parameters

out	<i>p_config_name</i>	保存した設定ファイル名の返却用。
in	<i>stl_format</i>	TriMeshIO クラスで定義されている STL ファイルのフォーマット。
in	<i>extend</i>	ファイル名に付加する文字列。省略可。省略した場合は、付加文字列として本メソッド呼び出し時の年月日時分秒 (YYYYMMDD24hhmmss) を用いる。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

ファイル名命名規約は次の通り。 定義ファイル : polylib\_config\_ランク番号\_付加文字.xml。 STL ファイル : ポリゴングループ名\_ランク番号\_付加文字.拡張子。

**3.8.3.18** char\* PolylibNS::Polylib::save\_config\_file ( std::string *rank\_no*, std::string *extend*, std::string *format* )  
[protected]

設定ファイルの保存。 メモリに展開しているグループツリー情報から設定ファイルを生成する。

## Parameters

in	<i>rank_no</i>	ランク番号
in	<i>extend</i>	ファイル名に付加する文字列
in	<i>format</i>	TriMeshIO クラスで定義されている STL ファイルのフォーマット。

## Returns

作成した設定ファイルの名称。エラー時はNULL が返る。

**3.8.3.19** POLYLIB\_STAT PolylibNS::Polylib::save\_with\_rankno ( std::string \* *p\_config\_name*, int *myrank*, int *maxrank*, std::string *extend*, std::string *stl\_format*, ID\_FORMAT *id\_format* ) [protected]

PolygoGroup ツリー、三角形ポリゴン情報の保存。 グループツリー情報を設定ファイルへ出力。三角形ポリゴン情報を STL ファイルへ出力。ID 情報を ID ファイルへ出力。ファイル名にランク番号を付加する。

## Parameters

out	<i>p_config_name</i>	保存した設定ファイル名の返却用。
in	<i>myrank</i>	自ランク番号。
in	<i>maxrank</i>	最大ランク番号。
in	<i>extend</i>	ファイ名に付加される文字列。
in	<i>stl_format</i>	STL ファイルフォーマット指定。
in	<i>id_format</i>	三角形ID ファイルの出力形式。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

ファイル名命名規約は次の通り。定義ファイル: polylib\_config\_ランク番号\_付加文字.xml。STL ファイル: ポリゴングループ名\_ランク番号\_付加文字.拡張子。ID ファイル: ポリゴングループ名\_ランク番号\_付加文字.ID。

MPIPolylib クラスがMPI 環境で利用することを想定している。

### 3.8.3.20 const Triangle\* PolylibNS::Polylib::search\_nearest\_polygon ( std::string group\_name, const Vec3f & pos ) const

指定した点に最も近い三角形ポリゴンの検索。

## Parameters

in	<i>group_name</i>	抽出グループ名。
in	<i>pos</i>	指定した点。

## Returns

検索されたポリゴン

### 3.8.3.21 std::vector<Triangle\*>\* PolylibNS::Polylib::search\_polygons ( std::string group\_name, Vec3f min\_pos, Vec3f max\_pos, bool every ) const

三角形ポリゴンの検索。位置ベクトル min\_pos と max\_pos により特定される矩形領域に含まれる、三角形ポリゴンを group\_name で指定されたグループの下から探索する。

## Parameters

in	<i>group_name</i>	抽出グループ名。
in	<i>min_pos</i>	抽出する矩形領域の最小値。
in	<i>max_pos</i>	抽出する矩形領域の最大値。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:3 頂点の一部でも検索領域と重なるものを抽出。

## Returns

抽出した三角形ポリゴンの vector。

## Attention

返却した三角形ポリゴンは、削除不可。vector は要削除。

**3.8.3.22 void PolylibNS::Polylib::set\_factory ( PolygonGroupFactory \* *factory* = NULL )**

PolygonGroup クラスを生成するためのFactory クラスを登録。本メソッドは、独自のFactory クラスを登録しない限り、呼び出し不要である。コンストラクタで生成したFactory クラスを破棄し、代わりに引数で指定された Factory クラスを登録する。

## Parameters

<i>in</i>	<i>factory</i>	Factory クラス。
-----------	----------------	--------------

## Attention

PolygonGroup を拡張した場合、拡張後のPolygonGroup のFactory クラスを登録する。

### 3.8.3.23 POLYLIB\_STAT PolylibNS::Polylib::setfilepath ( std::map< std::string, std::string > & *stl\_fname\_map* ) [protected]

TextParser 内部データに save した stl ファイルの "filepath"を書き込む。

save したSTL ファイルとPolygonGroup の階層は、save\_stl\_file に map を渡し保持してもらう。その map の内容に基づき、TextParser 内部のデータを 変更する。

## Parameters

<i>in</i>	<i>stl_fname_map</i>	save したSTL ファイルとその階層の map 型データ
-----------	----------------------	--------------------------------

## Returns

POLYLIB\_STAT で定義される値が返る。

### 3.8.3.24 void PolylibNS::Polylib::show\_group\_hierarchy ( FILE \* *fp* = NULL )

グループ階層構造を標準出力に出力。 2010.10.20 引数FILE \* 追加。

## Parameters

<i>in</i>	<i>fp</i>	出力先ファイル。指定されて行ければ、標準出力へ出力する。
-----------	-----------	------------------------------

## Attention

テスト用の関数であり、通常は利用者が用いるものではない。

### 3.8.3.25 POLYLIB\_STAT PolylibNS::Polylib::show\_group\_info ( std::string *group\_name* )

グループの情報と配下の三角形ポリゴン情報を標準出力に出力。 親グループ名、自身の名前、STL ファイル名、登録三角形数、3 頂点ベクトルの座標、法線ベクトルの座標、面積。

## Parameters

<i>in</i>	<i>group_name</i>	グループ名。
-----------	-------------------	--------

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

テスト用の関数であり、通常は利用者が用いるものではない。

### 3.8.3.26 void PolylibNS::Polylib::show\_group\_name ( PolygonGroup \* *p*, std::string *tab*, FILE \* *fp* ) [protected]

グループ名の表示。 指定されたグループ以下の階層構造をツリー形式で標準出力に出力する。 2010.10.20 引数FILE \* 追加。

## Parameters

in	<i>p</i>	検索の基点となるPolygonGroup のポインタ
in	<i>tab</i>	階層の深さを示すスペース
in	<i>fp</i>	出力先ファイル。指定されて行ければ、標準出力へ出力する。

## 3.8.3.27 unsigned int PolylibNS::Polylib::used\_memory\_size ( )

Polylib が利用中の概算メモリ量を返す

## Returns

利用中のメモリ量 (byte)

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/Polylib.h

## 3.9 PolylibNS::PolylibMoveParams Class Reference

```
#include <Polylib.h>
```

## Public Attributes

- int [m\\_current\\_step](#)  
現在の計算ステップ番号
- int [m\\_next\\_step](#)  
移動後の計算ステップ番号
- double [m\\_delta\\_t](#)  
1 計算ステップあたりの時間変異

## 3.9.1 Detailed Description

クラス:[PolylibMoveParams](#) [Polylib::move\(\)](#)の引数として利用するパラメタセットクラスです。本クラスメンバ変数ではパラメタが不足する場合は、継承クラスをユーザ定義してください。

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/Polylib.h

## 3.10 PolylibNS::PolylibStat2 Class Reference

```
#include <PolylibStat.h>
```

## Static Public Member Functions

- static std::string [String](#) (POLYLIB\_STAT stat)

## 3.10.1 Detailed Description

PolylibStat 文字列出力用クラス

### 3.10.2 Member Function Documentation

#### 3.10.2.1 static std::string PolylibNS::PolylibStat2::String ( POLYLIB\_STAT stat ) [inline],[static]

PolylibStat 文字列出力。

##### Parameters

<i>in</i>	<i>stat</i>	PolylibStat 値。
-----------	-------------	----------------

##### Returns

PolylibStat 値を文字列化したもの。

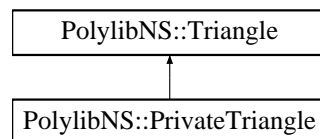
The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/common/PolylibStat.h

## 3.11 PolylibNS::PrivateTriangle Class Reference

```
#include <Triangle.h>
```

Inheritance diagram for PolylibNS::PrivateTriangle:



### Public Member Functions

- [PrivateTriangle](#) ([Vec3f](#) vertex[3], int id)
- [PrivateTriangle](#) ([Vec3f](#) vertex[3], [Vec3f](#) normal, int id)
- [PrivateTriangle](#) ([Vec3f](#) vertex[3], [Vec3f](#) normal, float area, int id)
- [PrivateTriangle](#) ([Triangle](#) tri, int id)
- [PrivateTriangle](#) (const [PrivateTriangle](#) &tri)
- [PrivateTriangle](#) (float \*dim, int id)
- void [set\\_id](#) (int id)
- int [get\\_id](#) () const

### Protected Attributes

- int [m\\_id](#)

### Additional Inherited Members

#### 3.11.1 Detailed Description

クラス:PrivateTriangle クラス Polylib 内のデータ保存用の基本クラスです。



### 3.11.2 Constructor & Destructor Documentation

#### 3.11.2.1 PolylibNS::PrivateTriangle::PrivateTriangle ( Vec3f *vertex*[3], int *id* ) [inline]

コンストラクタ。

## Parameters

in	<i>vertex</i>	ポリゴンの頂点。
in	<i>id</i>	三角形ポリゴンID。

3.11.2.2 PolylibNS::PrivateTriangle::PrivateTriangle ( Vec3f *vertex*[3], Vec3f *normal*, int *id* ) [inline]

コンストラクタ。

## Parameters

in	<i>vertex</i>	ポリゴンの頂点。
in	<i>normal</i>	法線。
in	<i>id</i>	三角形ポリゴンID。

3.11.2.3 PolylibNS::PrivateTriangle::PrivateTriangle ( Vec3f *vertex*[3], Vec3f *normal*, float *area*, int *id* ) [inline]

コンストラクタ。

## Parameters

in	<i>vertex</i>	ポリゴンの頂点。
in	<i>normal</i>	法線。
in	<i>area</i>	ポリゴンの面積。
in	<i>id</i>	三角形ポリゴンID。

3.11.2.4 PolylibNS::PrivateTriangle::PrivateTriangle ( Triangle *tri*, int *id* ) [inline]

コンストラクタ。

## Parameters

in	<i>tri</i>	ポリゴン。
in	<i>id</i>	三角形ポリゴンID。

3.11.2.5 PolylibNS::PrivateTriangle::PrivateTriangle ( const PrivateTriangle & *tri* ) [inline]

コンストラクタ。

## Parameters

in	<i>tri</i>	ポリゴン。
----	------------	-------

3.11.2.6 PolylibNS::PrivateTriangle::PrivateTriangle ( float \* *dim*, int *id* ) [inline]

コンストラクタ。

## Parameters

in	<i>dim</i>	ポリゴン頂点座標配列。
in	<i>id</i>	三角形ポリゴンID。

## 3.11.3 Member Function Documentation

3.11.3.1 `int PolylibNS::PrivateTriangle::get_id ( ) const` `[inline]`

三角形ポリゴンID を返す。

#### Returns

三角形ポリゴンID。

3.11.3.2 `void PolylibNS::PrivateTriangle::set_id ( int id )` `[inline]`

三角形ポリゴンID を設定。

#### Parameters

<code>in</code>	<code>id</code>	三角形ポリゴンID。
-----------------	-----------------	------------

### 3.11.4 Member Data Documentation

3.11.4.1 `int PolylibNS::PrivateTriangle::m_id` `[protected]`

PolygonGroup 内で一意となる三角形ポリゴンID。

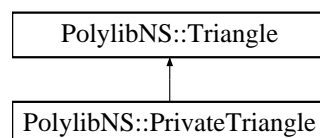
The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/polygons/Triangle.h

## 3.12 PolylibNS::Triangle Class Reference

```
#include <Triangle.h>
```

Inheritance diagram for PolylibNS::Triangle:



### Public Member Functions

- [Triangle](#) ()
- [Triangle](#) (Vec3f vertex[3])
- [Triangle](#) (Vec3f vertex[3], Vec3f normal)
- [Triangle](#) (Vec3f vertex[3], Vec3f normal, float area)
- void [set\\_vertexes](#) (Vec3f vertex[3], bool [calc\\_normal](#), bool [calc\\_area](#))
- Vec3f \* [get\\_vertex](#) () const
- Vec3f [get\\_normal](#) () const
- float [get\\_area](#) () const
- void [set\\_exid](#) (int id)
- int [get\\_exid](#) () const
- void [set\\_shell](#) (int val)
- int [get\\_shell](#) () const

## Protected Member Functions

- void `calc_normal` ()
- void `calc_area` ()

## Protected Attributes

- `Vec3f m_vertex` [3]  
三角形の頂点座標（反時計回りで並んでいる）。
- `Vec3f m_normal`  
三角形の法線ベクトル。
- float `m_area`  
三角形の面積。
- int `m_exid`  
三角形のユーザ定義ID
- int `m_shell`  
三角形のユーザ定義状態変数

### 3.12.1 Detailed Description

クラス: `Triangle` 入出力用インターフェースクラスであり、本ヘッダに対応する.cxx ファイルは存在 しない。

### 3.12.2 Constructor & Destructor Documentation

#### 3.12.2.1 `PolylibNS::Triangle::Triangle ( )` [inline]

コンストラクタ。

#### 3.12.2.2 `PolylibNS::Triangle::Triangle ( Vec3f vertex[3] )` [inline]

コンストラクタ。

##### Parameters

in	<i>vertex</i>	ポリゴンの頂点。
----	---------------	----------

##### Attention

面積と法線は `vertex` を元に自動計算される。

#### 3.12.2.3 `PolylibNS::Triangle::Triangle ( Vec3f vertex[3], Vec3f normal )` [inline]

コンストラクタ。

##### Parameters

in	<i>vertex</i>	ポリゴンの頂点。
in	<i>normal</i>	法線。

##### Attention

面積は `vertex` を元に自動計算される。

3.12.2.4 PolylibNS::Triangle ( Vec3f *vertex*[3], Vec3f *normal*, float *area* ) [inline]

コンストラクタ。

## Parameters

in	<i>vertex</i>	ポリゴンの頂点。
in	<i>normal</i>	法線。
in	<i>area</i>	ポリゴンの面積。

## 3.12.3 Member Function Documentation

## 3.12.3.1 void PolylibNS::Triangle::calc\_area ( ) [inline],[protected]

面積算出。

## 3.12.3.2 void PolylibNS::Triangle::calc\_normal ( ) [inline],[protected]

法線ベクトル算出。

## 3.12.3.3 float PolylibNS::Triangle::get\_area ( ) const [inline]

面積を取得。

## Returns

面積。

## 3.12.3.4 int PolylibNS::Triangle::get\_exid ( ) const [inline]

ユーザ定義ID を取得。

## Returns

ユーザ定義ID。

## 3.12.3.5 Vec3f PolylibNS::Triangle::get\_normal ( ) const [inline]

法線ベクトルを取得。

## Returns

法線ベクトル。

## 3.12.3.6 int PolylibNS::Triangle::get\_shell ( ) const [inline]

ユーザ定義状態変数を取得。

## Returns

ユーザ定義状態変数。

**3.12.3.7 Vec3f\* PolylibNS::Triangle::get\_vertex ( ) const** [inline]

vertex の配列を取得。

**Returns**

vertex の配列。

**3.12.3.8 void PolylibNS::Triangle::set\_exid ( int id )** [inline]

ユーザ定義ID を設定。

**3.12.3.9 void PolylibNS::Triangle::set\_shell ( int val )** [inline]

ユーザ定義状態変数を設定。

**3.12.3.10 void PolylibNS::Triangle::set\_vertexes ( Vec3f vertex[3], bool calc\_normal, bool calc\_area )** [inline]

頂点を設定。

**Parameters**

in	<i>vertex</i>	三角形の 3 頂点。
in	<i>calc_normal</i>	法線ベクトルを再計算するか？
in	<i>calc_area</i>	面積を再計算するか？

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/polygons/Triangle.h

## 3.13 TriangleStruct Struct Reference

```
#include <CPolylib.h>
```

**Public Attributes**

- float [m\\_vertex](#) [9]  
3 頂点座標
- float [m\\_normal](#) [3]  
法線ベクトル
- float [m\\_area](#)  
面積

### 3.13.1 Detailed Description

三角形ポリゴン情報構造体

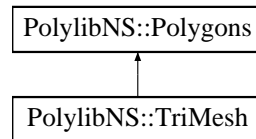
The documentation for this struct was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/c\_lang/CPolylib.h

### 3.14 PolylibNS::TriMesh Class Reference

```
#include <TriMesh.h>
```

Inheritance diagram for PolylibNS::TriMesh:



#### Public Member Functions

- [TriMesh](#) ()
- [~TriMesh](#) ()
- void [init](#) (const std::vector< [PrivateTriangle](#) \* > \*trias)
- void [add](#) (const std::vector< [PrivateTriangle](#) \* > \*trias)
- POLYLIB\_STAT [import](#) (const std::map< std::string, std::string > fmap, float scale=1.0)
- POLYLIB\_STAT [build](#) ()
- int [triangles\\_num](#) ()
- const std::vector< [PrivateTriangle](#) \* > \* [search](#) (BBox \*bbox, bool every) const
- POLYLIB\_STAT [search](#) (BBox \*bbox, bool every, std::vector< [PrivateTriangle](#) \* > \*tri\_list) const
- const std::vector< [PrivateTriangle](#) \* > \* [linear\\_search](#) (BBox \*q\_bbox, bool every) const
- POLYLIB\_STAT [linear\\_search](#) (BBox \*q\_bbox, bool every, std::vector< [PrivateTriangle](#) \* > \*tri\_list) const
- const [PrivateTriangle](#) \* [search\\_nearest](#) (const [Vec3f](#) &pos) const
- POLYLIB\_STAT [set\\_all\\_exid](#) (const int id) const
- [BBox](#) [get\\_bbox](#) () const
- [VTree](#) \* [get\\_vtree](#) () const

#### Additional Inherited Members

##### 3.14.1 Detailed Description

クラス:[TriMesh](#) 三角形ポリゴン集合を管理するクラス (KD 木用に特化したクラス)。

##### 3.14.2 Constructor & Destructor Documentation

###### 3.14.2.1 PolylibNS::TriMesh::TriMesh ( )

コンストラクタ。

###### 3.14.2.2 PolylibNS::TriMesh::~~TriMesh ( )

デストラクタ。

##### 3.14.3 Member Function Documentation

###### 3.14.3.1 void PolylibNS::TriMesh::add ( const std::vector< [PrivateTriangle](#) \* > \*trias ) [virtual]

三角形ポリゴンリストに引数で与えられる三角形の複製を追加する。



## Parameters

<i>in</i>	<i>trias</i>	設定する三角形ポリゴンリスト。
-----------	--------------	-----------------

## Attention

*m\_id* が重複するインスタンスは追加されない。  
KD 木の再構築は行わない。

Implements [PolylibNS::Polygons](#).

## 3.14.3.2 POLYLIB\_STAT PolylibNS::TriMesh::build ( ) [virtual]

Polygons クラスに含まれる全ポリゴン情報からKD 木を作成する。

## Returns

POLYLIB\_STAT で定義される値が返る。

Implements [PolylibNS::Polygons](#).

## 3.14.3.3 BBox PolylibNS::TriMesh::get\_bbox ( ) const [inline]

TriMesh クラスが管理しているBoundingBox を返す。

## 3.14.3.4 VTree\* PolylibNS::TriMesh::get\_vtree ( ) const [inline],[virtual]

KD 木クラスを取得。

## Returns

KD 木クラス。

Implements [PolylibNS::Polygons](#).

3.14.3.5 POLYLIB\_STAT PolylibNS::TriMesh::import ( const std::map< std::string, std::string > *fmap*, float *scale* = 1.0 ) [virtual]

ファイルからデータの初期化。

## Parameters

<i>in</i>	<i>fmap</i>	ファイル名、ファイルフォーマット。
-----------	-------------	-------------------

## Returns

PLSTAT\_OK=成功/false=失敗

Implements [PolylibNS::Polygons](#).

3.14.3.6 void PolylibNS::TriMesh::init ( const std::vector< PrivateTriangle \* > \* *trias* ) [virtual]

TriMesh クラスで管理する三角形ポリゴンリストを初期化し、引数で与えられる三角形ポリゴンリストを設定する。三角形ポリゴン用のメモリ領域は、Polylib 内で新たに確保される。

## Parameters

in	<i>trias</i>	設定する三角形ポリゴンリスト。
----	--------------	-----------------

Implements [PolylibNS::Polygons](#).

**3.14.3.7** `const std::vector<PrivateTriangle*>* PolylibNS::TriMesh::linear_search ( BBox * q_bbox, bool every ) const`  
[virtual]

線形探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

## Parameters

in	<i>q_bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

## Returns

抽出したポリゴンリストのポインタ。

## Attention

三角形ポリゴンのメモリ領域は新たにPolylib 内で確保される。  
MPIPolylib 内での利用が目的なので、ユーザは使用しないこと。

Implements [PolylibNS::Polygons](#).

**3.14.3.8** `POLYLIB_STAT PolylibNS::TriMesh::linear_search ( BBox * q_bbox, bool every, std::vector< PrivateTriangle * > * tri_list ) const` [virtual]

線形探索により、指定矩形領域に含まれるポリゴンを抽出する。

## Parameters

in	<i>q_bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in, out	<i>tri_list</i>	抽出した三角形ポリゴンリストへのポインタ。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

*tri\_list* で戻される三角形ポリゴンのポインタは、Polylib 内で 保持されるアドレス値なので、ユーザは delete しないで下さい。  
オーバーロードメソッドあり。

Implements [PolylibNS::Polygons](#).

**3.14.3.9** `const std::vector<PrivateTriangle*>* PolylibNS::TriMesh::search ( BBox * bbox, bool every ) const`  
[virtual]

KD 木探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

## Returns

抽出したポリゴンリストのポインタ。

## Attention

三角形ポリゴンのメモリ領域は新たにPolylib 内で確保される。  
MPIPolylib 内での利用が目的なので、ユーザは使用しないこと。  
オーバーロードメソッドあり。

Implements [PolylibNS::Polygons](#).

**3.14.3.10** `POLYLIB_STAT PolylibNS::TriMesh::search ( BBox * bbox, bool every, std::vector< PrivateTriangle * > * tri_list ) const` `[virtual]`

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in, out	<i>tri_list</i>	抽出した三角形ポリゴンリストへのポインタ。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

*tri\_list* で戻される三角形ポリゴンのポインタは、Polylib 内で 保持されるアドレス値なので、ユーザは delete しないで下さい。  
オーバーロードメソッドあり。

Implements [PolylibNS::Polygons](#).

**3.14.3.11** `const PrivateTriangle* PolylibNS::TriMesh::search_nearest ( const Vec3f & pos ) const` `[virtual]`

KD 木探索により、指定位置に最も近いポリゴンを検索する。

## Parameters

in	<i>pos</i>	指定位置
----	------------	------

## Returns

検索されたポリゴン

Implements [PolylibNS::Polygons](#).

**3.14.3.12** `POLYLIB_STAT PolylibNS::TriMesh::set_all_exid ( const int id ) const` `[virtual]`

配下の全ポリゴンの *m\_exid* 値を指定値にする。

## Parameters

<code>in</code>	<code>id</code>	指定値
-----------------	-----------------	-----

Implements [PolylibNS::Polygons](#).

**3.14.3.13** `int PolylibNS::TriMesh::triangles_num ( ) [virtual]`

TriMesh クラスが管理している三角形ポリゴン数を返す。

Implements [PolylibNS::Polygons](#).

The documentation for this class was generated from the following file:

- `/home/tkawanab/Polylib/Polylib-2.5/include/polygons/TriMesh.h`

## 3.15 PolylibNS::TriMeshIO Class Reference

```
#include <TriMeshIO.h>
```

### Static Public Member Functions

- static POLYLIB\_STAT `load` (`std::vector< PrivateTriangle * > *tri_list`, `const std::map< std::string, std::string > &fmap`, `float scale=1.0`)
- static POLYLIB\_STAT `save` (`std::vector< PrivateTriangle * > *tri_list`, `std::string fname`, `std::string fmt=""`)
- static `std::string input_file_format` (`const std::string &filename`)

### Static Public Attributes

- static const `std::string FMT_STL_A`  
アスキーファイル
- static const `std::string FMT_STL_AA`  
アスキーファイル
- static const `std::string FMT_STL_B`  
バイナリファイル
- static const `std::string FMT_STL_BB`  
バイナリファイル
- static const `std::string DEFAULT_FMT`  
`TrimeshIO.cxx` で定義している値

### 3.15.1 Detailed Description

クラス:[TriMeshIO](#) 三角形ポリゴン入出力管理。

### 3.15.2 Member Function Documentation

**3.15.2.1** `static std::string PolylibNS::TriMeshIO::input_file_format ( const std::string & filename ) [static]`

ファイル名を元に入力ファイルのフォーマットを取得する。

## Parameters

in	<i>filename</i>	入力ファイル名。
----	-----------------	----------

## Returns

判定したファイルフォーマット。

## Attention

ファイル拡張子が"stl"の場合、ファイルを読み込んで判定する。

**3.15.2.2** `static POLYLIB_STAT PolylibNS::TriMeshIO::load ( std::vector< PrivateTriangle * > * tri_list, const std::map< std::string, std::string > & fmap, float scale = 1.0 ) [static]`

STL ファイルを読み込み、*tri\_list* にセットする。

## Parameters

in, out	<i>tri_list</i>	三角形ポリゴンリストの領域。
in	<i>fmap</i>	ファイル名、ファイルフォーマットのセット。

## Returns

POLYLIB\_STAT で定義される値が返る。

**3.15.2.3** `static POLYLIB_STAT PolylibNS::TriMeshIO::save ( std::vector< PrivateTriangle * > * tri_list, std::string fname, std::string fmt = "" ) [static]`

*tri\_list* の内容をSTL 形式でファイルへ保存。

## Parameters

in	<i>tri_list</i>	三角形ポリゴンのリスト (出力内容)。
in	<i>fname</i>	ファイル名。
in	<i>fmt</i>	ファイルフォーマット。

## Returns

POLYLIB\_STAT で定義される値が返る。

## 3.15.3 Member Data Documentation

**3.15.3.1** `const std::string PolylibNS::TriMeshIO::FMT_STL_A [static]`

アスキーファイル

STL ファイルのフォーマット種別

## Attention

STL ファイルの拡張子とは異なるので注意すること。

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/file\_io/TriMeshIO.h

### 3.16 PolylibNS::Vec2< T > Class Template Reference

```
#include <Vec2.h>
```

#### Public Member Functions

- **Vec2** (T v=0)
- **Vec2** (T \_x, T \_y)
- **Vec2** (const T v[2])
- **Vec2**< T > & **assign** (T \_x, T \_y)
- **operator T** \* ()
- **operator const T** \* () const
- T \* **ptr** ()
- const T \* **ptr** () const
- T & **operator[]** (int i)
- const T & **operator[]** (int i) const
- **Vec2**< T > & **operator+=** (const **Vec2**< T > &v)
- **Vec2**< T > & **operator-=** (const **Vec2**< T > &v)
- **Vec2**< T > & **operator\*=** (const **Vec2**< T > &v)
- **Vec2**< T > & **operator/=** (const **Vec2**< T > &v)
- **Vec2**< T > & **operator\*=** (T s)
- **Vec2**< T > & **operator/=** (T s)
- **Vec2**< T > **operator+** (const **Vec2**< T > &v) const
- **Vec2**< T > **operator-** (const **Vec2**< T > &v) const
- **Vec2**< T > **operator\*** (const **Vec2**< T > &v) const
- **Vec2**< T > **operator/** (const **Vec2**< T > &v) const
- **Vec2**< T > **operator\*** (T s) const
- **Vec2**< T > **operator/** (T s) const
- **Vec2**< T > **operator-** () const
- bool **operator==** (const **Vec2**< T > &v) const
- bool **operator!=** (const **Vec2**< T > &v) const
- float **lengthSquared** () const
- float **length** () const
- **Vec2**< T > & **normalize** ()
- **Vec2**< T > & **normalize** (float \*len)
- float **average** () const

#### Static Public Member Functions

- static **Vec2**< T > **xaxis** ()
- static **Vec2**< T > **yaxis** ()

#### Public Attributes

- T **x**
- T **y**

### 3.16.1 Detailed Description

```
template<typename T>class PolylibNS::Vec2< T >
```

クラス:Vec2<T>

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/common/Vec2.h

## 3.17 PolylibNS::Vec3< T > Class Template Reference

```
#include <Vec3.h>
```

### Public Member Functions

- **Vec3** (T v=0)
- **Vec3** (T \_x, T \_y, T \_z)
- **Vec3** (const T v[3])
- **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
- float **lengthSquared** () const
- float **length** () const
- **Vec3**< T > & **normalize** ()
- **Vec3**< T > & **normalize** (float \*len)
- float **average** () const

### Static Public Member Functions

- static **Vec3**< T > **xaxis** ()
- static **Vec3**< T > **yaxis** ()
- static **Vec3**< T > **zaxis** ()

## Public Attributes

- `T t [3]`

### 3.17.1 Detailed Description

`template<typename T>class PolylibNS::Vec3< T >`

クラス:Vec3<T>

The documentation for this class was generated from the following file:

- `/home/tkawanab/Polylib/Polylib-2.5/include/common/Vec3.h`

## 3.18 PolylibNS::VElement Class Reference

`#include <VTree.h>`

## Public Member Functions

- [VElement](#) ([PrivateTriangle](#) \*tri)
- [PrivateTriangle](#) \* [get\\_triangle](#) ()
- [Vec3f](#) [get\\_pos](#) () const
- [BBox](#) [get\\_bbox](#) () const

### 3.18.1 Detailed Description

クラス:[VElement](#) KD 木構造の要素クラスです。

### 3.18.2 Constructor & Destructor Documentation

#### 3.18.2.1 PolylibNS::VElement::VElement ( [PrivateTriangle](#) \* *tri* )

コンストラクタ。

#### Parameters

<i>in</i>	<i>tri</i>	ポリゴン情報のポインタ。
-----------	------------	--------------

#### Attention

ポインタを格納するが、参照のみ。delete は行わない。

### 3.18.3 Member Function Documentation

#### 3.18.3.1 BBox PolylibNS::VElement::get\_bbox ( ) const `[inline]`

Bounding box of this triangle

#### 3.18.3.2 Vec3f PolylibNS::VElement::get\_pos ( ) const `[inline]`

Center position of bbox on triangle.



### 3.18.3.3 PrivateTriangle\* PolylibNS::VElement::get\_triangle ( ) [inline]

triangle。

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/polygons/VTree.h

## 3.19 PolylibNS::VNode Class Reference

```
#include <VTree.h>
```

### Public Member Functions

- [VNode](#) ()
- [~VNode](#) ()
- void [split](#) (const int &max\_elem)
- bool [is\\_leaf](#) () const
- [BBox](#) [get\\_bbox](#) () const
- void [set\\_bbox](#) (const [BBox](#) &bbox)
- [BBox](#) [get\\_bbox\\_search](#) () const
- void [set\\_bbox\\_search](#) (const [VElement](#) \*p)
- [VNode](#) \* [get\\_left](#) ()
- [VNode](#) \* [get\\_right](#) ()
- AxisEnum [get\\_axis](#) () const
- void [set\\_axis](#) (const AxisEnum axis)
- std::vector< [VElement](#) \* > & [get\\_vlist](#) ()
- void [set\\_element](#) ([VElement](#) \*elm)
- int [get\\_elements\\_num](#) () const

### 3.19.1 Detailed Description

VNode クラス KD 木構造のノードクラスです。

### 3.19.2 Constructor & Destructor Documentation

#### 3.19.2.1 PolylibNS::VNode::VNode ( )

コンストラクタ。

#### 3.19.2.2 PolylibNS::VNode::~~VNode ( )

デストラクタ。

### 3.19.3 Member Function Documentation

#### 3.19.3.1 AxisEnum PolylibNS::VNode::get\_axis ( ) const [inline]

Axis を取得。

Returns

axis。

### 3.19.3.2 BBox PolylibNS::VNode::get\_bbox ( ) const [inline]

BBox の値を取得。

#### Returns

bbox。

### 3.19.3.3 BBox PolylibNS::VNode::get\_bbox\_search ( ) const [inline]

検索用BBox を取得。

#### Returns

検索用 bbox。

### 3.19.3.4 int PolylibNS::VNode::get\_elements\_num ( ) const [inline]

ノードが所持する要素の数を取得。

#### Returns

要素数。

### 3.19.3.5 VNode\* PolylibNS::VNode::get\_left ( ) [inline]

左のNode を取得。

#### Returns

左のNode。

### 3.19.3.6 VNode\* PolylibNS::VNode::get\_right ( ) [inline]

右のNode を取得。

#### Returns

右のNode。

### 3.19.3.7 std::vector<VElement\*>& PolylibNS::VNode::get\_vlist ( ) [inline]

要素のリストを取得。

#### Returns

要素のリスト。

### 3.19.3.8 bool PolylibNS::VNode::is\_leaf ( ) const [inline]

ノードがリーフかどうかの判定結果。

#### Returns

true=リーフ/false=リーフでない。

3.19.3.9 void PolylibNS::VNode::set\_axis ( const AxisEnum *axis* ) [inline]

Axis を設定。

## Parameters

in	<i>axis</i> <sub>o</sub>	
----	--------------------------	--

3.19.3.10 void PolylibNS::VNode::set\_bbox ( const BBox & *bbox* ) [inline]

BBox の値を設定。

## Parameters

in	<i>bbox</i> <sub>o</sub>	
----	--------------------------	--

3.19.3.11 void PolylibNS::VNode::set\_bbox\_search ( const VElement \* *p* ) [inline]

このノードのBounding Box を引数で与えられる要素を含めた大きさに変更する。

## Parameters

in	<i>p</i>	要素。
----	----------	-----

3.19.3.12 void PolylibNS::VNode::set\_element ( VElement \* *elm* ) [inline]

木の要素を設定。

## Parameters

in	<i>elm</i> <sub>o</sub>	
----	-------------------------	--

3.19.3.13 void PolylibNS::VNode::split ( const int & *max\_elem* )

ノードを 2 つの子供ノードに分割する。

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/polygons/VTree.h

## 3.20 PolylibNS::VTree Class Reference

```
#include <VTree.h>
```

### Public Member Functions

- [VTree](#) (int *max\_elem*, const [BBox](#) *bbox*, std::vector< [PrivateTriangle](#) \* > \**tri\_list*)
- [~VTree](#) ()
- void [destroy](#) ()
- std::vector< [PrivateTriangle](#) \* > \* [search](#) ([BBox](#) \**bbox*, bool *every*) const
- POLYLIB\_STAT [search](#) ([BBox](#) \**bbox*, bool *every*, std::vector< [PrivateTriangle](#) \* > \**tri\_list*) const
- const [PrivateTriangle](#) \* [search\\_nearest](#) (const [Vec3f](#) &*pos*) const
- const [PrivateTriangle](#) \* [search\\_nearest\\_recursive](#) ([VNode](#) \**vn*, const [Vec3f](#) &*pos*) const
- unsigned int [memory\\_size](#) ()

### 3.20.1 Detailed Description

クラス:[VTree](#) リーフを三角形ポリゴンとするKD 木クラスです。

### 3.20.2 Constructor & Destructor Documentation

#### 3.20.2.1 PolylibNS::VTree::VTree ( int *max\_elem*, const BBox *bbox*, std::vector< PrivateTriangle \* > \* *tri\_list* )

コンストラクタ。

##### Parameters

in	<i>max_elem</i>	最大要素数。
in	<i>bbox</i>	VTree の box 範囲。
in	<i>tri_list</i>	木構造の元になるポリゴンのリスト。

#### 3.20.2.2 PolylibNS::VTree::~~VTree ( )

デストラクタ。

### 3.20.3 Member Function Documentation

#### 3.20.3.1 void PolylibNS::VTree::destroy ( )

木構造を消去する。

#### 3.20.3.2 unsigned int PolylibNS::VTree::memory\_size ( )

KD 木クラスが利用しているメモリ量を返す。

##### Returns

利用中のメモリ量 (byte)

#### 3.20.3.3 std::vector<PrivateTriangle\*>\* PolylibNS::VTree::search ( BBox \* *bbox*, bool *every* ) const

KD 木探索により、指定矩形領域に含まれる三角形ポリゴンを抽出する。

##### Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。

##### Returns

抽出したポリゴンリストのポインタ。

##### Attention

MPIPolylib 用のメソッドなので、ユーザは利用しないで下さい。  
オーバーロードメソッドあり。

3.20.3.4 POLYLIB\_STAT PolylibNS::VTree::search ( BBox \* *bbox*, bool *every*, std::vector< PrivateTriangle \* > \* *tri\_list* )  
const

KD 木探索により、指定矩形領域に含まれるポリゴンを抽出する。

## Parameters

in	<i>bbox</i>	検索範囲を示す矩形領域。
in	<i>every</i>	true:3 頂点が全て検索領域に含まれるものを抽出。 false:1 頂点でも検索領域に含まれるものを抽出。
in, out	<i>tri_list</i>	抽出した三角形ポリゴンリストへのポインタ。

## Returns

POLYLIB\_STAT で定義される値が返る。

## Attention

オーバーロードメソッドあり。

## 3.20.3.5 const PrivateTriangle\* PolylibNS::VTree::search\_nearest ( const Vec3f &amp; pos ) const

KD 木探索により、指定位置に最も近いポリゴンを検索する。

## Parameters

in	<i>pos</i>	指定位置
----	------------	------

## Returns

検索されたポリゴン

## 3.20.3.6 const PrivateTriangle\* PolylibNS::VTree::search\_nearest\_recursive ( VNode \* vn, const Vec3f &amp; pos ) const

KD 木探索により、指定位置に最も近いポリゴンを検索する。

## Parameters

in	<i>vn</i>	検索対象のノードへのポインタ。
in	<i>pos</i>	指定位置

## Returns

検索されたポリゴン

The documentation for this class was generated from the following file:

- /home/tkawanab/Polylib/Polylib-2.5/include/polygons/VTree.h