## RoomKit Namespace Reference

### Classes

class **CoordGrid**

*Maintains a list of available and allocated points in a grid of the specified interval within the orthogonal bounding box of a Polygon.*

class **LineEx**

*Extends Hypar.Elements.Line with several utility methods.*

class **Messages**

*Common exception messages.*

class **Palette**

*Commonly used Colors for Space rendering.*

class **Place**

*Places 2D Polygons in various spatial relationships to each other.*

class **PolygonEx**

*Extends Hypar.Elements.Polygon with several utility methods.*

class **Room**

*A data structure recording room characteristics.*

class **RoomGroup**

*A data structure recording room characteristics.*

class **RoomRow**

*A data structure recording room characteristics.*

class **Scope**

*A data structure recording space program characteristics and status of a* ***Room*** *placing process.*

class **Shaper**

*Utilities for creating and editing Polygons.*

class **Spacer**

*Copies and places Hypar Spaces in various spatial relationships to each other.*

class **TopoBox**

*Maintains a set of points on the orthogonal bounding box of a supplied Polygon corresponding to four divisions of each side.*

class **Vector3Ex**

*Extends Hypar.Elements.Vector3 with utility methods.*

### Enumerations

enum **Corner** { **NE**, **SE**, **SW**, **NW** }

*A list of box corners as compass designations.* enum **Orient** { **C**, **N**, **NNE**, **NE**, **ENE**, **E**, **ESE**, **SE**, **SSE**, **S**, **SSW**, **SW**, **WSW**, **W**, **WNW**, **NW**, **NNW** }

*A list of compass orientations used to designate locations on a 2D bounding box.*

### Enumeration Type Documentation

#### enum RoomKit.Corner[strong]

A list of box corners as compass designations.

#### enum RoomKit.Orient[strong]

A list of compass orientations used to designate locations on a 2D bounding box.

# Class Documentation

## RoomKit.CoordGrid Class Reference

Maintains a list of available and allocated points in a grid of the specified interval within the orthogonal bounding box of a Polygon.

### Public Member Functions

**CoordGrid** (Polygon polygon, double xInterval=1, double yInterval=1)

*Creates an orthogonal 2D grid of Vector3 points from the supplied Polygon and axis intervals.*

void **Allocate** (Polygon polygon)

*Allocates the points in the grid falling within or on the supplied Polygon.*

void **Allocate** (IList< Polygon > polygons)

*Allocates points in the grid falling within the supplied Polygons.*

Vector3 **AllocatedNearTo** (Vector3 point)

*Returns the allocated grid point nearest to the supplied point.*

Vector3 **AllocatedRandom** ()

*Returns a random allocated point.*

Vector3 **AvailableMax** ()

*Returns the maximum available grid point.*

Vector3 **AvailableMin** ()

*Returns the minimum available grid point.*

Vector3 **AvailableNearTo** (Vector3 point)

*Returns the available grid point nearest to the supplied Vector3 point.*

Vector3 **AvailableRandom** ()

*Returns a random available grid point.*

### Properties

List< Vector3 > **Allocated** [get]

*The list of vector3 allocated points.*

List< Vector3 > **Available** [get]

*The list of Vector3 points available for allocation.*

Polygon **Perimeter** [get]

*The Polygon perimeter of the grid.*

### Detailed Description

Maintains a list of available and allocated points in a grid of the specified interval within the orthogonal bounding box of a Polygon.

### Constructor & Destructor Documentation

#### RoomKit.CoordGrid.CoordGrid (Polygon *polygon*, double *xInterval* = 1, double *yInterval* = 1)

Creates an orthogonal 2D grid of Vector3 points from the supplied Polygon and axis intervals.

##### Parameters:

|  |  |
| --- | --- |
| *perimeter* | The Polygon boundary of the point grid. |
| *xInterval* | The spacing of the grid along the x-axis. |
| *yInterval* | The spacing of the grid along the y-axis. |

##### Returns:

None.

///

### Member Function Documentation

#### void RoomKit.CoordGrid.Allocate (Polygon *polygon*)

Allocates the points in the grid falling within or on the supplied Polygon.

##### Parameters:

|  |  |
| --- | --- |
| *polygon* | The Polygon bounding the points to be allocated. |

##### Returns:

None.

#### void RoomKit.CoordGrid.Allocate (IList< Polygon > *polygons*)

Allocates points in the grid falling within the supplied Polygons.

##### Parameters:

|  |  |
| --- | --- |
| *polygon* | The Polygon bounding the points to be allocated. |

##### Returns:

None.

#### Vector3 RoomKit.CoordGrid.AllocatedNearTo (Vector3 *point*)

Returns the allocated grid point nearest to the supplied point.

##### Parameters:

|  |  |
| --- | --- |
| *point* | The Vector3 point to compare. |

##### Returns:

A Vector3 point.

#### Vector3 RoomKit.CoordGrid.AllocatedRandom ()

Returns a random allocated point.

##### Returns:

A Vector3 point.

#### Vector3 RoomKit.CoordGrid.AvailableMax ()

Returns the maximum available grid point.

##### Returns:

A Vector3 point.

#### Vector3 RoomKit.CoordGrid.AvailableMin ()

Returns the minimum available grid point.

##### Returns:

A Vector3 point.

#### Vector3 RoomKit.CoordGrid.AvailableNearTo (Vector3 *point*)

Returns the available grid point nearest to the supplied Vector3 point.

##### Parameters:

|  |  |
| --- | --- |
| *point* | The Vector3 point to compare. |

##### Returns:

A Vector3 point.

#### Vector3 RoomKit.CoordGrid.AvailableRandom ()

Returns a random available grid point.

##### Returns:

A Vector3 point.

### Property Documentation

#### List<Vector3> RoomKit.CoordGrid.Allocated[get]

The list of vector3 allocated points.

#### List<Vector3> RoomKit.CoordGrid.Available[get]

The list of Vector3 points available for allocation.

#### Polygon RoomKit.CoordGrid.Perimeter[get]

The Polygon perimeter of the grid.

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

C:/Users/Anthony/Oasis/Business/Hypar/GitHub/RoomKit/RoomKit/CoordGrid.cs

## RoomKit.Room Class Reference

A data structure recording room characteristics.

### Public Member Functions

**Room** ()

*Constructor setting all internal variables to default values.*

**Room** (string name="", int resourceID=-1, double designArea=1.0, **Color** color=null, int[] adjacentTo=null)

*Constructor setting the area of the* ***Room****.*

**Room** (Polygon perimeter, string name="", **Color** color=null)

*Constructor setting the perimeter of the* ***Room****.*

**Room** (string name="", int resourceID=-1, double designX=1.0, double designY=1.0, **Color** color=null, int[] adjacentTo=null)

*Constructor setting the X and Y diemsions of a* ***Room****.*

Polygon **MakePerimeter** ()

*Creates a Polygon perimeter at the origin with dimensions derived from* ***Room*** *characteristics. Assumes the Perimeter will be relocated and so omits setting the* ***Room****'s Perimeter.*

### Properties

int [] **AdjacentTo** [get]

*A list of Resource ID integers indicating the desired adjacencies of this* ***Room*** *type to other* ***Room*** *types.*

Color **Color** [get, set]

*Public property of color, required to allow setting an initial value.*

double **DesignArea** [get]

*The desired area of this* ***Room****. Overridden if values of DesignX and DesignY are set to positive values.*

double **DesignX** [get]

*The desired x-axis dimension of this* ***Room****. Overrides DesignArea if DesignY is also set to a positive value.*

double **DesignY** [get]

*The desired y-axis dimension of this* ***Room****. Overrides DesignArea if DesignX is also set to a positive value.*

double **Elevation** [get, set]

*The vertical position of the* ***Room****'s lowest plane, parallel to the ground plane.*

double **Height** [get, set]

*Public property of the height of the* ***Room*** *prism. Required to allow error checking for new heights.*

string **Name** [get]

*Arbitrary string identifier for this* ***Room*** *instance. Has no effect on* ***Room*** *operations.*

Polygon **Perimeter** [get, set]

*Public property of the 2D Polygon perimeter of the* ***Room****. Required to allow error checking for a non-null perimeter.*

int **ResourceID** [get]

*Arbitrary integer identifier of this* ***Room*** *type. Can be used to identify desired adjacencies.*

string **UniqueID** [get]

*A UUID for this* ***Room*** *instance, set on initialization.*

double **Area** [get]

*The area of the room's perimeter Polygon. Returns -1.0 if the* ***Room****'s Perimeter is null.*

double **AreaVariance** [get]

*The ratio between the intended area and the actual area of the* ***Room****. Returns a negative value if the* ***Room*** *has no Perimeter value.*

Space **AsSpace** [get]

*A Space created from* ***Room*** *characteristics.*

### Detailed Description

A data structure recording room characteristics.

### Constructor & Destructor Documentation

#### RoomKit.Room.Room ()

Constructor setting all internal variables to default values.

#### RoomKit.Room.Room (string *name* = "", int *resourceID* = -1, double *designArea* = 1.0, Color *color* = null, int [] *adjacentTo* = null)

Constructor setting the area of the **Room**.

#### RoomKit.Room.Room (Polygon *perimeter*, string *name* = "", Color *color* = null)

Constructor setting the perimeter of the **Room**.

#### RoomKit.Room.Room (string *name* = "", int *resourceID* = -1, double *designX* = 1.0, double *designY* = 1.0, Color *color* = null, int [] *adjacentTo* = null)

Constructor setting the X and Y diemsions of a **Room**.

### Member Function Documentation

#### Polygon RoomKit.Room.MakePerimeter ()

Creates a Polygon perimeter at the origin with dimensions derived from **Room** characteristics. Assumes the Perimeter will be relocated and so omits setting the **Room**'s Perimeter.

##### Returns:

A new rectilinear Polygon derived either from fixed dimensions or as a rectilinear target area of a randomly determined ratio between 1 and 2 between the **Room**'s X and Y dimensions.

### Property Documentation

#### int [] RoomKit.Room.AdjacentTo[get]

A list of Resource ID integers indicating the desired adjacencies of this **Room** type to other **Room** types.

#### double RoomKit.Room.Area[get]

The area of the room's perimeter Polygon. Returns -1.0 if the **Room**'s Perimeter is null.

#### double RoomKit.Room.AreaVariance[get]

The ratio between the intended area and the actual area of the **Room**. Returns a negative value if the **Room** has no Perimeter value.

#### Space RoomKit.Room.AsSpace[get]

A Space created from **Room** characteristics.

#### Color RoomKit.Room.Color[get], [set]

Public property of color, required to allow setting an initial value.

#### double RoomKit.Room.DesignArea[get]

The desired area of this **Room**. Overridden if values of DesignX and DesignY are set to positive values.

#### double RoomKit.Room.DesignX[get]

The desired x-axis dimension of this **Room**. Overrides DesignArea if DesignY is also set to a positive value.

#### double RoomKit.Room.DesignY[get]

The desired y-axis dimension of this **Room**. Overrides DesignArea if DesignX is also set to a positive value.

#### double RoomKit.Room.Elevation[get], [set]

The vertical position of the **Room**'s lowest plane, parallel to the ground plane.

#### double RoomKit.Room.Height[get], [set]

Public property of the height of the **Room** prism. Required to allow error checking for new heights.

#### string RoomKit.Room.Name[get]

Arbitrary string identifier for this **Room** instance. Has no effect on **Room** operations.

#### Polygon RoomKit.Room.Perimeter[get], [set]

Public property of the 2D Polygon perimeter of the **Room**. Required to allow error checking for a non-null perimeter.

#### int RoomKit.Room.ResourceID[get]

Arbitrary integer identifier of this **Room** type. Can be used to identify desired adjacencies.

#### string RoomKit.Room.UniqueID[get]

A UUID for this **Room** instance, set on initialization.

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

C:/Users/Anthony/Oasis/Business/Hypar/GitHub/RoomKit/RoomKit/Room.cs

## RoomKit.RoomGroup Class Reference

A data structure recording room characteristics.

### Public Member Functions

**RoomGroup** (Polygon perimeter, string name="")

**RoomGroup** (Line row, string name="")

bool **AddRoom** (**Room** room, IList< Polygon > among=null)

*Attempts to place a room within the perimeter of the group or on its row line, depending on the grooup's initial geometry.*

### Properties

string **Name** [get, set]

Polygon **Perimeter** [get]

IList< **Room** > **Rooms** [get, set]

Line **Row** [get]

double **AvailableArea** [get]

*The unallocated area of the* ***RoomGroup*** *Perimeter.*

double **AvailableLength** [get]

*The unallocated area of the* ***RoomGroup*** *Row.*

double **AreaPlaced** [get]

*The area allocated within the* ***RoomGroup****.*

IList< Polygon > **PerimetersRooms** [get]

*A list of all placed room perimeters.*

### Detailed Description

A data structure recording room characteristics.

### Member Function Documentation

#### bool RoomKit.RoomGroup.AddRoom (Room *room*, IList< Polygon > *among* = null)

Attempts to place a room within the perimeter of the group or on its row line, depending on the grooup's initial geometry.

##### Parameters:

|  |  |
| --- | --- |
| *room* | The **Room** from which to derive the Polygon to place. |

##### Returns:

True if the room was successfully placed.

### Property Documentation

#### double RoomKit.RoomGroup.AreaPlaced[get]

The area allocated within the **RoomGroup**.

#### double RoomKit.RoomGroup.AvailableArea[get]

The unallocated area of the **RoomGroup** Perimeter.

#### double RoomKit.RoomGroup.AvailableLength[get]

The unallocated area of the **RoomGroup** Row.

#### IList<Polygon> RoomKit.RoomGroup.PerimetersRooms[get]

A list of all placed room perimeters.

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

C:/Users/Anthony/Oasis/Business/Hypar/GitHub/RoomKit/RoomKit/RoomGroup.cs

## RoomKit.RoomRow Class Reference

A data structure recording room characteristics.

### Public Member Functions

**RoomRow** ()

*Constructor initializes the* ***RoomRow*** *with default empty values.*

**RoomRow** (Line row, string name="")

*Constructor initializes the* ***RoomRow*** *with a new Line and an optional name.*

bool **AddRoom** (**Room** room, Polygon within=null, IList< Polygon > among=null, double circ=2.0)

### Properties

Polygon **Circulation** [get]

*The circulation envelope around the row.*

double **Depth** = 0.0 [get]

*The depth of the deepest room along the row.*

string **Name** [get, set]

*Arbitrary string identifier for this* ***RoomRow*** *instance. Has no effect on* ***RoomRow*** *operations.*

IList< **Room** > **Rooms** [get]

*The list of Rooms placed along this Row.*

Line **Row** [get]

*The Line along which Rooms can be placed.*

double **AvailableLength** [get]

*The unallocated length of the* ***RoomRow****.*

double **AreaPlaced** [get]

*The aggregate area of the Rooms placed on this Row.*

IList< Polygon > **PerimetersRooms** [get]

*A list of all placed* ***Room*** *perimeters.*

### Detailed Description

A data structure recording room characteristics.

### Constructor & Destructor Documentation

#### RoomKit.RoomRow.RoomRow ()

Constructor initializes the **RoomRow** with default empty values.

#### RoomKit.RoomRow.RoomRow (Line *row*, string *name* = "")

Constructor initializes the **RoomRow** with a new Line and an optional name.

### Member Function Documentation

#### bool RoomKit.RoomRow.AddRoom (Room *room*, Polygon *within* = null, IList< Polygon > *among* = null, double *circ* = 2.0)

Attempts to place a **Room** perimeter on the next open segment of the Row, with optional restrictions of a perimeter within which the **Room**'s Polygon must fit and a list of Polygons which it cannot intersect.

##### Parameters:

|  |  |
| --- | --- |
| *room* | The **Room** from which to derive the Polygon to place. |

##### Returns:

True if the room was successfully placed.

### Property Documentation

#### double RoomKit.RoomRow.AreaPlaced[get]

The aggregate area of the Rooms placed on this Row.

#### double RoomKit.RoomRow.AvailableLength[get]

The unallocated length of the **RoomRow**.

#### Polygon RoomKit.RoomRow.Circulation[get]

The circulation envelope around the row.

#### double RoomKit.RoomRow.Depth = 0.0[get]

The depth of the deepest room along the row.

#### string RoomKit.RoomRow.Name[get], [set]

Arbitrary string identifier for this **RoomRow** instance. Has no effect on **RoomRow** operations.

#### IList<Polygon> RoomKit.RoomRow.PerimetersRooms[get]

A list of all placed **Room** perimeters.

#### IList<Room> RoomKit.RoomRow.Rooms[get]

The list of Rooms placed along this Row.

#### Line RoomKit.RoomRow.Row[get]

The Line along which Rooms can be placed.

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

C:/Users/Anthony/Oasis/Business/Hypar/GitHub/RoomKit/RoomKit/RoomRow.cs

## RoomKit.Scope Class Reference

A data structure recording space program characteristics and status of a **Room** placing process.

### Public Member Functions

**Room** **Find** (double area)

*Finds the room with the design area closest to the supplied area.*

**Room** **Find** (double designX, double designY)

*Finds the room with the designed x and y dimensions closest to the supplied values.*

**Room** **FindUnplaced** (double area)

*Finds the unplaced* ***Room*** *with the design area closest to the supplied area.*

**Room** **FindUnplaced** (double designX, double designY)

*Finds the unplaced* ***Room*** *with the designed x and y dimensions closest to the supplied values.*

**Room** **FindUnplaced** (int resourceID)

*Finds the first unplaced* ***Room*** *with the specifed ResourceID.*

### Properties

List< **Room** > **Circulation** [get]

List< **Room** > **Occupant** [get]

List< **Room** > **Service** [get]

List< **Room** > **Tenant** [get]

double **AreaDesignCirculation** [get]

*The area available for horizontal circulation.*

double **AreaRooms** [get]

*The allocated aggregate area of all placed occupant rooms.*

double **AreaService** [get]

*The aggregate area of all services.*

double **AreaTenant** [get]

*The aggregate of all occupiable tenant areas.*

double **DesignAreaOccupant** [get]

*The intended aggregate area of all occupant rooms.*

double **MaxRoomDim** [get]

*The maximum fixed dimension of Occupant Rooms.*

double **MinRoomDim** [get]

*The minimum fixed dimension of Occupant Rooms.*

List< Polygon > **PerimetersAllocated** [get]

*A list of allocated Circulation, Occupant, and Service Polygon areas.*

List< Polygon > **PerimetersCirculation** [get]

*Returns a list of all Polygons in the Circulation category.*

List< Polygon > **PerimetersOccupant** [get]

*Returns a list of all Polygons in the Occupant category.*

List< Polygon > **PerimetersService** [get]

*Returns a list of all Polygons in the Service category.*

List< Polygon > **PerimetersTenant** [get]

*Returns a list of all Polygons in the Tenant category.*

IList< **Room** > **Placed** [get]

*Returns all placed Rooms.*

bool **PlacedAll** [get]

*Returns whether all spaces in Spaces have been placed.*

double **QuantityPlaced** [get]

*The quantity of placed Rooms.*

double **QuantityUnplaced** [get]

*The quantity of unplaced Rooms.*

double **RatioCirculation** [get]

*Returns the ratio of the aggregate area of all rooms against the circulation area.*

IList< **Room** > **Unplaced** [get]

*Returns all unplaced Rooms.*

### Detailed Description

A data structure recording space program characteristics and status of a **Room** placing process.

### Member Function Documentation

#### Room RoomKit.Scope.Find (double *area*)

Finds the room with the design area closest to the supplied area.

##### Parameters:

|  |  |
| --- | --- |
| *area* | The area to match from the list of all **Room** definitions. |

##### Returns:

A **Room**.

#### Room RoomKit.Scope.Find (double *designX*, double *designY*)

Finds the room with the designed x and y dimensions closest to the supplied values.

##### Parameters:

|  |  |
| --- | --- |
| *designX* | The x-axis dimension to match. |
| *designY* | The y-axis dimension to match. |

##### Returns:

A **Room**.

#### Room RoomKit.Scope.FindUnplaced (double *area*)

Finds the unplaced **Room** with the design area closest to the supplied area.

##### Parameters:

|  |  |
| --- | --- |
| *area* | The area to match from the list of all unplaced **Room** definitions. |

##### Returns:

An unplaced **Room**.

#### Room RoomKit.Scope.FindUnplaced (double *designX*, double *designY*)

Finds the unplaced **Room** with the designed x and y dimensions closest to the supplied values.

##### Parameters:

|  |  |
| --- | --- |
| *designX* | The x-axis dimension to match. |
| *designY* | The y-axis dimension to match. |

##### Returns:

An unplaced **Room**.

#### Room RoomKit.Scope.FindUnplaced (int *resourceID*)

Finds the first unplaced **Room** with the specifed ResourceID.

##### Parameters:

|  |  |
| --- | --- |
| *resourceID* | The integer ID of a **Room** type. |

##### Returns:

A **Room**.

### Property Documentation

#### double RoomKit.Scope.AreaDesignCirculation[get]

The area available for horizontal circulation.

#### double RoomKit.Scope.AreaRooms[get]

The allocated aggregate area of all placed occupant rooms.

#### double RoomKit.Scope.AreaService[get]

The aggregate area of all services.

#### double RoomKit.Scope.AreaTenant[get]

The aggregate of all occupiable tenant areas.

#### double RoomKit.Scope.DesignAreaOccupant[get]

The intended aggregate area of all occupant rooms.

#### double RoomKit.Scope.MaxRoomDim[get]

The maximum fixed dimension of Occupant Rooms.

#### double RoomKit.Scope.MinRoomDim[get]

The minimum fixed dimension of Occupant Rooms.

#### List<Polygon> RoomKit.Scope.PerimetersAllocated[get]

A list of allocated Circulation, Occupant, and Service Polygon areas.

#### List<Polygon> RoomKit.Scope.PerimetersCirculation[get]

Returns a list of all Polygons in the Circulation category.

##### Returns:

A list of Polygons.

#### List<Polygon> RoomKit.Scope.PerimetersOccupant[get]

Returns a list of all Polygons in the Occupant category.

##### Returns:

A list of Polygons.

#### List<Polygon> RoomKit.Scope.PerimetersService[get]

Returns a list of all Polygons in the Service category.

##### Returns:

A list of Polygons.

#### List<Polygon> RoomKit.Scope.PerimetersTenant[get]

Returns a list of all Polygons in the Tenant category.

##### Returns:

A list of Polygons.

#### IList<Room> RoomKit.Scope.Placed[get]

Returns all placed Rooms.

##### Returns:

A list of Rooms.

#### bool RoomKit.Scope.PlacedAll[get]

Returns whether all spaces in Spaces have been placed.

##### Returns:

Returns true if all spaces in Spaces have been marked as placed.

#### double RoomKit.Scope.QuantityPlaced[get]

The quantity of placed Rooms.

#### double RoomKit.Scope.QuantityUnplaced[get]

The quantity of unplaced Rooms.

#### double RoomKit.Scope.RatioCirculation[get]

Returns the ratio of the aggregate area of all rooms against the circulation area.

##### Returns:

A list of Rooms.

#### IList<Room> RoomKit.Scope.Unplaced[get]

Returns all unplaced Rooms.

##### Returns:

A list of Rooms.

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

C:/Users/Anthony/Oasis/Business/Hypar/GitHub/RoomKit/RoomKit/Scope.cs

## RoomKit.TopoBox Class Reference

Maintains a set of points on the orthogonal bounding box of a supplied Polygon corresponding to four divisions of each side.

### Public Member Functions

**TopoBox** (Polygon polygon)

*Constructor creates a new mathematical bounding box and populates all orientation points.*

Vector3 **PointBy** (**Orient** orient)

*Returns the requested bounding box location by orientation.*

### Properties

Vector3 **C** [get]

*Vector3 location identifiers corresponding to points on the box perimeter.*

Vector3 **N** [get]

Vector3 **NNW** [get]

Vector3 **NW** [get]

Vector3 **WNW** [get]

Vector3 **W** [get]

Vector3 **WSW** [get]

Vector3 **SW** [get]

Vector3 **SSW** [get]

Vector3 **S** [get]

Vector3 **SSE** [get]

Vector3 **SE** [get]

Vector3 **ESE** [get]

Vector3 **E** [get]

Vector3 **ENE** [get]

Vector3 **NE** [get]

Vector3 **NNE** [get]

double **SizeX** [get]

double **SizeY** [get]

### Detailed Description

Maintains a set of points on the orthogonal bounding box of a supplied Polygon corresponding to four divisions of each side.

### Constructor & Destructor Documentation

#### RoomKit.TopoBox.TopoBox (Polygon *polygon*)

Constructor creates a new mathematical bounding box and populates all orientation points.

### Member Function Documentation

#### Vector3 RoomKit.TopoBox.PointBy (Orient *orient*)

Returns the requested bounding box location by orientation.

##### Parameters:

|  |  |
| --- | --- |
| *orient* | The Orient value to index point. |

##### Returns:

A 2D Vector3 point.

### Property Documentation

#### Vector3 RoomKit.TopoBox.C[get]

Vector3 location identifiers corresponding to points on the box perimeter.

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

C:/Users/Anthony/Oasis/Business/Hypar/GitHub/RoomKit/RoomKit/TopoBox.cs

# Index

INDEX