RoomKit

Hypar Inc.

Version 0.2.0

Wed Feb 20 2019

# Class Index

## Class List

[**RoomKit.CoordGrid**](#AAAAAAAAAB)

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

[**RoomKit.Room**](#AAAAAAAAAC)

**A data structure recording room characteristics.**

[**RoomKit.RoomGroup**](#AAAAAAAAAD)

**Creates and manages Rooms within a perimeter.**

[**RoomKit.RoomRow**](#AAAAAAAAAE)

**Creates and manages Rooms placed along a line**

[**RoomKit.Scope**](#AAAAAAAAAF)

**A data structure recording space program characteristics and the status of a** [**Room**](#AAAAAAAAAC) **placing process.**

[**RoomKit.Story**](#AAAAAAAAAG)

**Creates and manages the geometry of a slab and Rooms representing corridors, occupied rooms, and services.**

[**RoomKit.TopoBox**](#AAAAAAAAAH)

**Maintains a set of points on the orthogonal bounding box of a supplied Polygon corresponding to four divisions of each side. N, S, E, and W define middle points on each orthogonal side of the box. NE, NW, SE, and SW correspond to the corners of the box. Other compass points define points along the relevant side between the cardinal and corner points. C corresponds to the center of the box.**

# Namespace Documentation

## RoomKit Namespace Reference

### Classes

class **ArcEx**

*Extends Elements.Geometry.Arc with utility methods.*

class [CoordGrid](#AAAAAAAAAB)

*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 Elements.Geometry.Line with utility methods.*

class **Messages**

*Common exception messages.*

class **Palette**

*Commonly used Colors for Space rendering. These colors are translucent to allow viewing of representions several layers deep.*

class **Place**

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

class **PolygonEx**

class [Room](#AAAAAAAAAC)

*A data structure recording room characteristics.*

class [RoomGroup](#AAAAAAAAAD)

*Creates and manages Rooms within a perimeter.*

class [RoomRow](#AAAAAAAAAE)

*Creates and manages Rooms placed along a line.*

class [Scope](#AAAAAAAAAF)

*A data structure recording space program characteristics and the status of a* [*Room*](#AAAAAAAAAC) *placing process.*

class **Shaper**

*Utilities for creating and editing Polygons.*

class [Story](#AAAAAAAAAG)

*Creates and manages the geometry of a slab and Rooms representing corridors, occupied rooms, and services.*

class [TopoBox](#AAAAAAAAAH)

*Maintains a set of points on the orthogonal bounding box of a supplied Polygon corresponding to four divisions of each side. N, S, E, and W define middle points on each orthogonal side of the box. NE, NW, SE, and SW correspond to the corners of the box. Other compass points define points along the relevant side between the cardinal and corner points. C corresponds to the center of the box.*

class **Vector3Ex**

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

### Enumerations

enum [Corner](#AAAAAAAAAI) { **NE**, **SE**, **SW**, **NW** }

*A list of box corners as compass designations.* enum [Orient](#AAAAAAAAAJ) { **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 box. N, S, E, and W define middle points on each orthogonal side of the box. NE, NW, SE, and SW correspond to the corners of the box. Other compass points define points along the relevant side between the cardinal and corner points. C corresponds to the center of the box.*

### Enumeration Type Documentation

#### enum [RoomKit.Corner](#AAAAAAAAAI)[strong]

A list of box corners as compass designations.

#### enum [RoomKit.Orient](#AAAAAAAAAJ)[strong]

A list of compass orientations used to designate locations on a 2D box. N, S, E, and W define middle points on each orthogonal side of the box. NE, NW, SE, and SW correspond to the corners of the box. Other compass points define points along the relevant side between the cardinal and corner points. C corresponds to the center of the 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](#AAAAAAAAAK) (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](#AAAAAAAAAL) (Polygon polygon)

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

void [Allocate](#AAAAAAAAAM) (IList< Polygon > polygons)

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

Vector3 [AllocatedNearTo](#AAAAAAAAAN) (Vector3 point)

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

Vector3 [AllocatedRandom](#AAAAAAAAAO) ()

*Returns a random allocated point.*

Vector3 [AvailableMax](#AAAAAAAAAP) ()

*Returns the maximum available grid point.*

Vector3 [AvailableMin](#AAAAAAAAAQ) ()

*Returns the minimum available grid point.*

Vector3 [AvailableNearTo](#AAAAAAAAAR) (Vector3 point)

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

Vector3 [AvailableRandom](#AAAAAAAAAS) ()

*Returns a random available grid point.*

### Properties

List< Vector3 > [Allocated](#AAAAAAAAAT) [get]

*The list of vector3 allocated points.*

List< Vector3 > [Available](#AAAAAAAAAU) [get]

*The list of Vector3 points available for allocation.*

Polygon [Perimeter](#AAAAAAAAAV) [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:

A new Coordgrid object.

### 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:

RoomKit/CoordGrid.cs

## RoomKit.Room Class Reference

A data structure recording room characteristics.

### Public Member Functions

[Room](#AAAAAAAAAW) ()

*Constructor setting all internal variables to default values, a 1.0 x 1.0 x 1.0 white cube placed on the zero plane with a blank name, no perimeter, and a numeric ID of -1.*

Polygon [MakePerimeter](#AAAAAAAAAX) (Vector3 moveTo=null)

*Places a Polygon east of another Polygon, attempting to align bounding box corners or the horizontal bounding box axis.*

Polygon [MakePerimeter](#AAAAAAAAAY) (Line axis, double width)

*Creates and sets a rectangular* [*Room*](#AAAAAAAAAC) *perimeter with dimensions derived from a supplied Line and a width. Intended for creating corridors.*

Polygon [MakePerimeter](#AAAAAAAAAZ) (Vector3 start, Vector3 end, double width)

*Creates and sets a rectangular* [*Room*](#AAAAAAAAAC) *perimeter with dimensions derived from two points and a width. Intended for creating corridors.*

### Properties

int [] [AdjacentTo](#AAAAAAAABA) [get, set]

*A list of Resource ID integers indicating the desired adjacencies of this* [*Room*](#AAAAAAAAAC) *type to other* [*Room*](#AAAAAAAAAC) *types.*

Color **Color** [get, set]

double **DesignArea** [get, set]

double **DesignX** [get, set]

double **DesignY** [get, set]

double [Elevation](#AAAAAAAABF) [get, set]

*The vertical position of the* [*Room*](#AAAAAAAAAC)*'s lowest plane, parallel to the ground plane.*

double **Height** [get, set]

string [Name](#AAAAAAAABH) [get, set]

*Arbitrary string identifier for this* [*Room*](#AAAAAAAAAC) *instance. Has no effect on* [*Room*](#AAAAAAAAAC) *operations.*

Polygon **Perimeter** [get, set]

int [ResourceID](#AAAAAAAABJ) [get, set]

*Arbitrary integer identifier of this* [*Room*](#AAAAAAAAAC) *type. Can be used to identify desired adjacencies.*

string [UniqueID](#AAAAAAAABK) [get]

*A UUID for this* [*Room*](#AAAAAAAAAC) *instance, set on initialization.*

double [Area](#AAAAAAAABL) [get]

*The area of the room's perimeter Polygon. Returns -1.0 if the* [*Room*](#AAAAAAAAAC)*'s Perimeter is null.*

double [AreaVariance](#AAAAAAAABM) [get]

*The ratio between the intended area and the actual area of the* [*Room*](#AAAAAAAAAC)*. Returns a negative value if the* [*Room*](#AAAAAAAAAC) *has no Perimeter value.*

Space [AsSpace](#AAAAAAAABN) [get]

*A Space created from* [*Room*](#AAAAAAAAAC) *characteristics.*

### Detailed Description

A data structure recording room characteristics.

### Constructor & Destructor Documentation

#### RoomKit.Room.Room ()

Constructor setting all internal variables to default values, a 1.0 x 1.0 x 1.0 white cube placed on the zero plane with a blank name, no perimeter, and a numeric ID of -1.

### Member Function Documentation

#### Polygon RoomKit.Room.MakePerimeter (Vector3 *moveTo* = null)

Places a Polygon east of another Polygon, attempting to align bounding box corners or the horizontal bounding box axis.

##### Parameters:

|  |  |
| --- | --- |
| *polygon* | The Polygon to be placed adjacent to another Polygon. |
| *adjTo* | The Polygon adjacent to which the new Polygon will be located. |
| *perimeter* | The Polygon that must cover the resulting Polygon. |
| *among* | The collection of Polygons that must not intersect the resulting Polygon. |

##### Returns:

A new Polygon or null if the conditions of placement cannot be satisfied.

Creates and sets a rectangular [Room](#AAAAAAAAAC) Perimeter with dimensions derived from [Room](#AAAAAAAAAC) characteristics with its southwest corner at the supplied Vector3 point. If no point is supplied, the southwest corner is placed at the origin.

##### Parameters:

|  |  |
| --- | --- |
| *moveTo* | The Vector3 indication the location of new Polygon's southwest corner. |

##### Returns:

A new rectilinear Polygon derived either from fixed DesignX and DesignY dimensions or as a rectilinear target area of a random ratio between 1 and 2 of the [Room](#AAAAAAAAAC)'s X to Y dimensions.

#### Polygon RoomKit.Room.MakePerimeter (Line *axis*, double *width*)

Creates and sets a rectangular [Room](#AAAAAAAAAC) perimeter with dimensions derived from a supplied Line and a width. Intended for creating corridors.

##### Parameters:

|  |  |
| --- | --- |
| *axis* | The Line defining the centerline of the perimeter. |
| *width* | The width of the perimeter along the axis Line. |

##### Returns:

A new rectilinear Polygon derived from the axis and the width.

#### Polygon RoomKit.Room.MakePerimeter (Vector3 *start*, Vector3 *end*, double *width*)

Creates and sets a rectangular [Room](#AAAAAAAAAC) perimeter with dimensions derived from two points and a width. Intended for creating corridors.

##### Parameters:

|  |  |
| --- | --- |
| *start* | The start point of an axis defining centerline of the perimeter. |
| *end* | The end point of an axis defining centerline of the perimeter. |
| *width* | The width of the perimeter along the axis Line. |

##### Returns:

A new rectilinear Polygon derived from the axis and the width.

### Property Documentation

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

A list of Resource ID integers indicating the desired adjacencies of this [Room](#AAAAAAAAAC) type to other [Room](#AAAAAAAAAC) types.

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

The area of the room's perimeter Polygon. Returns -1.0 if the [Room](#AAAAAAAAAC)'s Perimeter is null.

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

The ratio between the intended area and the actual area of the [Room](#AAAAAAAAAC). Returns a negative value if the [Room](#AAAAAAAAAC) has no Perimeter value.

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

A Space created from [Room](#AAAAAAAAAC) characteristics.

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

The vertical position of the [Room](#AAAAAAAAAC)'s lowest plane, parallel to the ground plane.

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

Arbitrary string identifier for this [Room](#AAAAAAAAAC) instance. Has no effect on [Room](#AAAAAAAAAC) operations.

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

Arbitrary integer identifier of this [Room](#AAAAAAAAAC) type. Can be used to identify desired adjacencies.

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

A UUID for this [Room](#AAAAAAAAAC) instance, set on initialization.

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

RoomKit/Room.cs

## RoomKit.RoomGroup Class Reference

Creates and manages Rooms within a perimeter.

### Public Member Functions

[RoomGroup](#AAAAAAAABO) (Polygon perimeter, int xRooms=1, int yRooms=1, string name="")

*Creates a group of rooms by dividing the supplied Polygon perimeter by the quantity of supplied divisions along the orthogonal x and y axes.* [*Room*](#AAAAAAAAAC) *perimeters conform to fit within the supplied Polygon.*

void [SetElevation](#AAAAAAAABP) (double elevation)

*Uniformly sets the elevation of all Rooms in the* [*RoomGroup*](#AAAAAAAAAD)*.*

void [SetHeight](#AAAAAAAABQ) (double height)

*Uniformly sets the height of all Rooms in the* [*RoomGroup*](#AAAAAAAAAD)*.*

### Properties

string [Name](#AAAAAAAABR) [get, set]

*An arbitrary string identifier for this* [*RoomGroup*](#AAAAAAAAAD)*.*

Polygon [Perimeter](#AAAAAAAABS) [get]

*The Polygon within which all Rooms are placed.*

IList< [Room](#AAAAAAAAAC) > [Rooms](#AAAAAAAABT) [get]

*The list of Rooms placed within the Perimeter.*

double [AvailableArea](#AAAAAAAABU) [get]

*The unallocated area of the* [*RoomGroup*](#AAAAAAAAAD) *perimeter.*

double [AreaPlaced](#AAAAAAAABV) [get]

*The area allocated within the* [*RoomGroup*](#AAAAAAAAAD)*.*

IList< Polygon > [PerimetersRooms](#AAAAAAAABW) [get]

*A list of all placed* [*Room*](#AAAAAAAAAC) *perimeters.*

### Detailed Description

Creates and manages Rooms within a perimeter.

### Constructor & Destructor Documentation

#### RoomKit.RoomGroup.RoomGroup (Polygon *perimeter*, int *xRooms* = 1, int *yRooms* = 1, string *name* = "")

Creates a group of rooms by dividing the supplied Polygon perimeter by the quantity of supplied divisions along the orthogonal x and y axes. [Room](#AAAAAAAAAC) perimeters conform to fit within the supplied Polygon.

##### Parameters:

|  |  |
| --- | --- |
| *perimeter* | The Polygon to divide with a number of [Room](#AAAAAAAAAC) perimeters. |
| *xRooms* | The quantity of Rooms along the x axis. |
| *yRooms* | The quantity of Rooms along the y axis. |
| *name* | An arbitrary string identifier for this [RoomGroup](#AAAAAAAAAD). |

##### Returns:

A new [RoomGroup](#AAAAAAAAAD).

### Member Function Documentation

#### void RoomKit.RoomGroup.SetElevation (double *elevation*)

Uniformly sets the elevation of all Rooms in the [RoomGroup](#AAAAAAAAAD).

##### Parameters:

|  |  |
| --- | --- |
| *elevation* | The new elevation of the Rooms. |

##### Returns:

None.

#### void RoomKit.RoomGroup.SetHeight (double *height*)

Uniformly sets the height of all Rooms in the [RoomGroup](#AAAAAAAAAD).

##### Parameters:

|  |  |
| --- | --- |
| *elevation* | The new height of the Rooms. |

##### Returns:

None.

### Property Documentation

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

The area allocated within the [RoomGroup](#AAAAAAAAAD).

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

The unallocated area of the [RoomGroup](#AAAAAAAAAD) perimeter.

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

An arbitrary string identifier for this [RoomGroup](#AAAAAAAAAD).

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

The Polygon within which all Rooms are placed.

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

A list of all placed [Room](#AAAAAAAAAC) perimeters.

#### IList<[Room](#AAAAAAAAAC)> RoomKit.RoomGroup.Rooms[get]

The list of Rooms placed within the Perimeter.

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

RoomKit/RoomGroup.cs

## RoomKit.RoomRow Class Reference

Creates and manages Rooms placed along a line.

### Public Member Functions

[RoomRow](#AAAAAAAABX) ()

*Constructor initializes the* [*RoomRow*](#AAAAAAAAAE) *with default values.*

[RoomRow](#AAAAAAAABY) (Line row, string name="")

*Constructor initializes the* [*RoomRow*](#AAAAAAAAAE) *with a new Line and an optional name.*

[RoomRow](#AAAAAAAABZ) (Vector3 start, Vector3 end, string name="")

*Constructor initializes the* [*RoomRow*](#AAAAAAAAAE) *with line endpoints and an optional name.*

bool [AddRoom](#AAAAAAAACA) ([Room](#AAAAAAAAAC) room, Polygon within=null, IList< Polygon > among=null, double circ=2.0)

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

### Properties

Polygon [Circulation](#AAAAAAAACB) [get]

*The circulation envelope around the row.*

double [Depth](#AAAAAAAACC) = 0.0 [get]

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

string [Name](#AAAAAAAACD) [get, set]

*Arbitrary string identifier for this* [*RoomRow*](#AAAAAAAAAE)*.*

IList< [Room](#AAAAAAAAAC) > [Rooms](#AAAAAAAACE) [get]

*The list of Rooms placed along the Row.*

Line [Row](#AAAAAAAACF) [get]

*The Line along which Rooms can be placed.*

double [AvailableLength](#AAAAAAAACG) [get]

*The unallocated length of the* [*RoomRow*](#AAAAAAAAAE)*.*

double [AreaPlaced](#AAAAAAAACH) [get]

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

IList< Polygon > [PerimetersRooms](#AAAAAAAACI) [get]

*A list of all placed* [*Room*](#AAAAAAAAAC) *perimeters.*

### Detailed Description

Creates and manages Rooms placed along a line.

### Constructor & Destructor Documentation

#### RoomKit.RoomRow.RoomRow ()

Constructor initializes the [RoomRow](#AAAAAAAAAE) with default values.

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

Constructor initializes the [RoomRow](#AAAAAAAAAE) with a new Line and an optional name.

#### RoomKit.RoomRow.RoomRow (Vector3 *start*, Vector3 *end*, string *name* = "")

Constructor initializes the [RoomRow](#AAAAAAAAAE) with line endpoints and an optional name.

### Member Function Documentation

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

Attempts to place a [Room](#AAAAAAAAAC) perimeter on the next open segment of the Row, with optional restrictions of a perimeter within which the [Room](#AAAAAAAAAC)'s perimeter must fit and a list of Polygons which it cannot intersect.

##### Parameters:

|  |  |
| --- | --- |
| *room* | The [Room](#AAAAAAAAAC) from which to derive the Polygon to place. |
| *within* | The optional Polygon perimeter within which a new [Room](#AAAAAAAAAC) must fit. |
| *among* | The optional list of Polygon perimeters the new [Room](#AAAAAAAAAC) cannot intersect. |
| *circ* | The optional additional allowance opposite the Row to allow for circulation to the Rooms. |

##### 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](#AAAAAAAAAE).

#### 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](#AAAAAAAAAE).

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

A list of all placed [Room](#AAAAAAAAAC) perimeters.

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

The list of Rooms placed along the 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:

RoomKit/RoomRow.cs

## RoomKit.Scope Class Reference

A data structure recording space program characteristics and the status of a [Room](#AAAAAAAAAC) placing process.

### Public Member Functions

[Scope](#AAAAAAAACJ) ()

*Contructor creates empty* [*Room*](#AAAAAAAAAC) *lists for Circulation, Occupation, Service, and Tenant.*

[Room](#AAAAAAAAAC) [Find](#AAAAAAAACK) (double area)

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

[Room](#AAAAAAAAAC) [Find](#AAAAAAAACL) (double designX, double designY)

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

[Room](#AAAAAAAAAC) [FindUnplaced](#AAAAAAAACM) (double area)

*Finds the unplaced* [*Room*](#AAAAAAAAAC) *with the design area closest to the supplied area.*

[Room](#AAAAAAAAAC) [FindUnplaced](#AAAAAAAACN) (double designX, double designY)

*Finds the unplaced* [*Room*](#AAAAAAAAAC) *with the designed x and y dimensions closest to the supplied values.*

[Room](#AAAAAAAAAC) [FindUnplaced](#AAAAAAAACO) (int resourceID)

*Finds the first unplaced* [*Room*](#AAAAAAAAAC) *with the specifed ResourceID.*

### Properties

List< [Room](#AAAAAAAAAC) > [Circulation](#AAAAAAAACP) [get]

*A list of Rooms designated as circulation.*

List< [Room](#AAAAAAAAAC) > [Occupant](#AAAAAAAACQ) [get]

*A list of Rooms designated for occupation, rather than circulation.*

List< [Room](#AAAAAAAAAC) > [Service](#AAAAAAAACR) [get]

*A list of Rooms designated for building services.*

List< [Room](#AAAAAAAAAC) > [Tenant](#AAAAAAAACS) [get]

*A list of Rooms intended as a series of tenant space containers of other Rooms.*

double [AreaDesignCirculation](#AAAAAAAACT) [get]

*The area available for horizontal circulation.*

double [AreaCirculation](#AAAAAAAACU) [get]

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

double [AreaRooms](#AAAAAAAACV) [get]

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

double [AreaService](#AAAAAAAACW) [get]

*The aggregate area of all services.*

double [AreaTenant](#AAAAAAAACX) [get]

*The aggregate of all occupiable tenant areas.*

double [DesignAreaOccupant](#AAAAAAAACY) [get]

*The intended aggregate area of all occupant rooms.*

double [MaxRoomDim](#AAAAAAAACZ) [get]

*The maximum fixed dimension of Occupant Rooms.*

double [MinRoomDim](#AAAAAAAADA) [get]

*The minimum fixed dimension of Occupant Rooms.*

List< Polygon > [PerimetersAllocated](#AAAAAAAADB) [get]

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

List< Polygon > [PerimetersCirculation](#AAAAAAAADC) [get]

*A list of all Circulation perimeter Polygons.*

List< Polygon > [PerimetersOccupant](#AAAAAAAADD) [get]

*A list of all Occupant perimeter Polygons.*

List< Polygon > [PerimetersService](#AAAAAAAADE) [get]

*A list of all Service perimeter Polygons.*

List< Polygon > [PerimetersTenant](#AAAAAAAADF) [get]

*A list of all Tenant perimeter Polygons.*

IList< [Room](#AAAAAAAAAC) > [Placed](#AAAAAAAADG) [get]

*Returns all placed Rooms.*

bool [PlacedAll](#AAAAAAAADH) [get]

*Returns whether Occupant Rooms have been placed.*

double [QuantityPlaced](#AAAAAAAADI) [get]

*The quantity of placed Rooms.*

double [QuantityUnplaced](#AAAAAAAADJ) [get]

*The quantity of unplaced Rooms.*

double [RatioCirculation](#AAAAAAAADK) [get]

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

IList< [Room](#AAAAAAAAAC) > [Unplaced](#AAAAAAAADL) [get]

*Returns all unplaced Rooms.*

### Detailed Description

A data structure recording space program characteristics and the status of a [Room](#AAAAAAAAAC) placing process.

### Constructor & Destructor Documentation

#### RoomKit.Scope.Scope ()

Contructor creates empty [Room](#AAAAAAAAAC) lists for Circulation, Occupation, Service, and Tenant.

##### Returns:

A new [Scope](#AAAAAAAAAF).

### Member Function Documentation

#### [Room](#AAAAAAAAAC) 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](#AAAAAAAAAC) definitions. |

##### Returns:

A [Room](#AAAAAAAAAC).

#### [Room](#AAAAAAAAAC) 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](#AAAAAAAAAC).

#### [Room](#AAAAAAAAAC) RoomKit.Scope.FindUnplaced (double *area*)

Finds the unplaced [Room](#AAAAAAAAAC) with the design area closest to the supplied area.

##### Parameters:

|  |  |
| --- | --- |
| *area* | The area to match from the list of all unplaced [Room](#AAAAAAAAAC) definitions. |

##### Returns:

An unplaced [Room](#AAAAAAAAAC).

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

Finds the unplaced [Room](#AAAAAAAAAC) 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](#AAAAAAAAAC).

#### [Room](#AAAAAAAAAC) RoomKit.Scope.FindUnplaced (int *resourceID*)

Finds the first unplaced [Room](#AAAAAAAAAC) with the specifed ResourceID.

##### Parameters:

|  |  |
| --- | --- |
| *resourceID* | The integer ID of a [Room](#AAAAAAAAAC) type. |

##### Returns:

A [Room](#AAAAAAAAAC).

### Property Documentation

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

The allocated aggregate area of all placed circulation rooms.

#### 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.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Scope.Circulation[get]

A list of Rooms designated as circulation.

#### 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<[Room](#AAAAAAAAAC)> RoomKit.Scope.Occupant[get]

A list of Rooms designated for occupation, rather than circulation.

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

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

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

A list of all Circulation perimeter Polygons.

##### Returns:

A list of Polygons.

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

A list of all Occupant perimeter Polygons.

##### Returns:

A list of Polygons.

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

A list of all Service perimeter Polygons.

##### Returns:

A list of Polygons.

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

A list of all Tenant perimeter Polygons.

##### Returns:

A list of Polygons.

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

Returns all placed Rooms.

##### Returns:

A list of Rooms.

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

Returns whether Occupant Rooms have been placed.

##### Returns:

Returns true each [Room](#AAAAAAAAAC) in Occupant has a perimeter.

#### 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 available circulation area.

##### Returns:

A list of Rooms.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Scope.Service[get]

A list of Rooms designated for building services.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Scope.Tenant[get]

A list of Rooms intended as a series of tenant space containers of other Rooms.

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

Returns all unplaced Rooms.

##### Returns:

A list of Rooms.

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

RoomKit/Scope.cs

## RoomKit.Story Class Reference

Creates and manages the geometry of a slab and Rooms representing corridors, occupied rooms, and services.

### Public Member Functions

[Story](#AAAAAAAADM) ()

*Creates a* [*Story*](#AAAAAAAAAG) *at a 1.0 Height on the zero plane with new lists for Corridors, Rooms, and Services. Perimeter is set to null, Name is blank, and SlabThickness is s0.1.*

void [AddCorridor](#AAAAAAAADN) (Line axis, double width=2.0, double height=3.0, string name="", Color color=null)

*Creates a rectangular corridor* [*Room*](#AAAAAAAAAC) *from a centerline axis, width, and height, at the* [*Story*](#AAAAAAAAAG) *elevation. Adds the new* [*Room*](#AAAAAAAAAC) *to the Corrdors list. Corridors conform to Service perimeters. Corridors change intersecting* [*Room*](#AAAAAAAAAC) *perimeters to conform to the corridor's perimeter.*

void [AddCorridor](#AAAAAAAADO) (Vector3 start, Vector3 end, double width=2.0, double height=3.0, string name="", Color color=null)

*Creates a rectangular corridor* [*Room*](#AAAAAAAAAC) *from a centerline axis, width, and height, at the* [*Story*](#AAAAAAAAAG) *elevation. Adds the new* [*Room*](#AAAAAAAAAC) *to the Corrdors list. Corridors conform to Service perimeters. Corridors change intersecting* [*Room*](#AAAAAAAAAC) *perimeters to conform to the corridor's perimeter.*

void [AddCorridor](#AAAAAAAADP) (Polygon perimeter, double height=3.0, string name="", Color color=null)

*Creates a corridor* [*Room*](#AAAAAAAAAC) *from a perimeter and height, at the* [*Story*](#AAAAAAAAAG) *elevation. Adds the new* [*Room*](#AAAAAAAAAC) *to the Corrdors list. Corridors conform to Service perimeters. Corridors change intersecting* [*Room*](#AAAAAAAAAC) *perimeters to conform to the corridor's perimeter.*

void [AddRoom](#AAAAAAAADQ) (Polygon perimeter, double height=3.0, string name="", Color color=null)

*Creates an occupied* [*Room*](#AAAAAAAAAC) *from a perimeter and height, at the* [*Story*](#AAAAAAAAAG) *elevation. Adds the new* [*Room*](#AAAAAAAAAC) *to the Rooms list. Rooms conform to corridor and service perimeters.*

void [AddService](#AAAAAAAADR) (Polygon perimeter, string name="", Color color=null)

*Creates a Service from a perimeter at the* [*Story*](#AAAAAAAAAG)*'s height and elevation. Adds the new* [*Room*](#AAAAAAAAAC) *to the Services list. Corridors and Rooms conform to Service perimeters.*

void [RoomsByDivision](#AAAAAAAADS) (int xRooms=1, int yRooms=1, double height=3.0, double setback=0.0, string name="", Color color=null)

*Creates Rooms by orthogonally dividing the interior of the* [*Story*](#AAAAAAAAAG) *perimeter by a quantity of x-axis and y-axis intervals. Adds the new Rooms to the Rooms list. New Rooms conform to Corridor and Service perimeters.*

### Properties

double [Area](#AAAAAAAADT) [get]

*Area of the perimeter.*

double [AreaPlaced](#AAAAAAAADU) [get]

*Area allocated to Corridors, Rooms, and Services.*

double [AreaAvailable](#AAAAAAAADV) [get]

*Unallocated area within the* [*Story*](#AAAAAAAAAG)*.*

Color **Color** [get, set]

List< [Room](#AAAAAAAAAC) > [Corridors](#AAAAAAAADX) [get]

*List of Rooms designated as cooridors.*

List< Space > [CorridorsAsSpaces](#AAAAAAAADY) [get]

*List of Spaces created from* [*Room*](#AAAAAAAAAC) *characteristics within the Corridors list.*

Color [CorridorsColor](#AAAAAAAADZ) [set]

*Sets the Corridors color.*

double **Elevation** [get, set]

Space [Envelope](#AAAAAAAAEB) [get]

*Space created from* [*Story*](#AAAAAAAAAG) *characteristics.*

double **Height** [get, set]

double [HeightInteriors](#AAAAAAAAED) [set]

*Sets the height of all Corridors, Rooms, and Services.*

IList< Space > [InteriorsAsSpaces](#AAAAAAAAEE) [get]

*Returns all Corridors, Rooms, and Services as Spaces.*

string [Name](#AAAAAAAAEF) [get, set]

*Arbitrary string identifier.*

Polygon **Perimeter** [get, set]

List< [Room](#AAAAAAAAAC) > [Rooms](#AAAAAAAAEH) [get]

*List of Rooms designated as occupiable rooms.*

List< Space > [RoomsAsSpaces](#AAAAAAAAEI) [get]

*List of Spaces created from* [*Room*](#AAAAAAAAAC) *characteristics within the Rooms list.*

Color [RoomsColor](#AAAAAAAAEJ) [set]

*Sets the Rooms Space rendering color.*

List< [Room](#AAAAAAAAAC) > [Services](#AAAAAAAAEK) [get]

*A list of Rooms designated as building services.*

List< Space > [ServicesAsSpaces](#AAAAAAAAEL) [get]

*List of Spaces created from* [*Room*](#AAAAAAAAAC) *characteristics within the Services list.*

Color [ServicesColor](#AAAAAAAAEM) [set]

*Sets the Services Space rendering color.*

Floor [Slab](#AAAAAAAAEN) [get]

*Concrete Floor created from* [*Story*](#AAAAAAAAAG) *and Slab characteristics.*

double **SlabThickness** [get, set]

### Detailed Description

Creates and manages the geometry of a slab and Rooms representing corridors, occupied rooms, and services.

### Constructor & Destructor Documentation

#### RoomKit.Story.Story ()

Creates a [Story](#AAAAAAAAAG) at a 1.0 Height on the zero plane with new lists for Corridors, Rooms, and Services. Perimeter is set to null, Name is blank, and SlabThickness is s0.1.

##### Parameters:

|  |  |
| --- | --- |
| *ratio* | The ratio of width to depth |
| *area* | The required area of the Polygon. |

##### Returns:

A new [Story](#AAAAAAAAAG).

### Member Function Documentation

#### void RoomKit.Story.AddCorridor (Line *axis*, double *width* = 2.0, double *height* = 3.0, string *name* = "", Color *color* = null)

Creates a rectangular corridor [Room](#AAAAAAAAAC) from a centerline axis, width, and height, at the [Story](#AAAAAAAAAG) elevation. Adds the new [Room](#AAAAAAAAAC) to the Corrdors list. Corridors conform to Service perimeters. Corridors change intersecting [Room](#AAAAAAAAAC) perimeters to conform to the corridor's perimeter.

##### Parameters:

|  |  |
| --- | --- |
| *axis* | Center Line of the corridor. |
| *width* | Width of the corridor. |
| *height* | Height of the corridor. |
| *name* | String identifier. |
| *color* | Rendering color of the [Room](#AAAAAAAAAC) as a Space. |

##### Returns:

None.

#### void RoomKit.Story.AddCorridor (Vector3 *start*, Vector3 *end*, double *width* = 2.0, double *height* = 3.0, string *name* = "", Color *color* = null)

Creates a rectangular corridor [Room](#AAAAAAAAAC) from a centerline axis, width, and height, at the [Story](#AAAAAAAAAG) elevation. Adds the new [Room](#AAAAAAAAAC) to the Corrdors list. Corridors conform to Service perimeters. Corridors change intersecting [Room](#AAAAAAAAAC) perimeters to conform to the corridor's perimeter.

##### Parameters:

|  |  |
| --- | --- |
| *start* | First endpoint of the centerline of the corridor. |
| *end* | Second endpoint of the centerline of the corridor. |
| *width* | Width of the corridor. |
| *height* | Height of the corridor. |
| *name* | String identifier. |
| *color* | Rendering color of the [Room](#AAAAAAAAAC) as a Space. |

##### Returns:

None.

#### void RoomKit.Story.AddCorridor (Polygon *perimeter*, double *height* = 3.0, string *name* = "", Color *color* = null)

Creates a corridor [Room](#AAAAAAAAAC) from a perimeter and height, at the [Story](#AAAAAAAAAG) elevation. Adds the new [Room](#AAAAAAAAAC) to the Corrdors list. Corridors conform to Service perimeters. Corridors change intersecting [Room](#AAAAAAAAAC) perimeters to conform to the corridor's perimeter.

##### Parameters:

|  |  |
| --- | --- |
| *perimeter* | Polygon perimeter of the corridor. |
| *height* | Height of the corridor. |
| *name* | String identifier. |
| *color* | Rendering color of the [Room](#AAAAAAAAAC) as a Space. |

##### Returns:

None.

#### void RoomKit.Story.AddRoom (Polygon *perimeter*, double *height* = 3.0, string *name* = "", Color *color* = null)

Creates an occupied [Room](#AAAAAAAAAC) from a perimeter and height, at the [Story](#AAAAAAAAAG) elevation. Adds the new [Room](#AAAAAAAAAC) to the Rooms list. Rooms conform to corridor and service perimeters.

##### Parameters:

|  |  |
| --- | --- |
| *perimeter* | Polygon perimeter of the corridor. |
| *height* | Height of the corridor. |
| *name* | String identifier. |
| *color* | Rendering color of the [Room](#AAAAAAAAAC) as a Space. |

##### Returns:

None.

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

Creates a Service from a perimeter at the [Story](#AAAAAAAAAG)'s height and elevation. Adds the new [Room](#AAAAAAAAAC) to the Services list. Corridors and Rooms conform to Service perimeters.

##### Parameters:

|  |  |
| --- | --- |
| *perimeter* | Polygon perimeter of the corridor. |
| *name* | String identifier. |
| *color* | Rendering color of the [Room](#AAAAAAAAAC) as a Space. |

##### Returns:

None.

#### void RoomKit.Story.RoomsByDivision (int *xRooms* = 1, int *yRooms* = 1, double *height* = 3.0, double *setback* = 0.0, string *name* = "", Color *color* = null)

Creates Rooms by orthogonally dividing the interior of the [Story](#AAAAAAAAAG) perimeter by a quantity of x-axis and y-axis intervals. Adds the new Rooms to the Rooms list. New Rooms conform to Corridor and Service perimeters.

##### Parameters:

|  |  |
| --- | --- |
| *xRooms* | Quantity Rooms along the orthogonal x-axis. |
| *yRooms* | Quantity Rooms along the orthogonal y-axis. |
| *height* | Height of the new Rooms. |
| *setback* | Offset from the [Story](#AAAAAAAAAG) perimeter. |
| *name* | String identifier applied to every new [Room](#AAAAAAAAAC). |
| *color* | Rendering color of the [Room](#AAAAAAAAAC) as a Space. |

##### Returns:

None.

### Property Documentation

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

Area of the perimeter.

#### double RoomKit.Story.AreaAvailable[get]

Unallocated area within the [Story](#AAAAAAAAAG).

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

Area allocated to Corridors, Rooms, and Services.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Story.Corridors[get]

List of Rooms designated as cooridors.

#### List<Space> RoomKit.Story.CorridorsAsSpaces[get]

List of Spaces created from [Room](#AAAAAAAAAC) characteristics within the Corridors list.

#### Color RoomKit.Story.CorridorsColor[set]

Sets the Corridors color.

#### Space RoomKit.Story.Envelope[get]

Space created from [Story](#AAAAAAAAAG) characteristics.

#### double RoomKit.Story.HeightInteriors[set]

Sets the height of all Corridors, Rooms, and Services.

#### IList<Space> RoomKit.Story.InteriorsAsSpaces[get]

Returns all Corridors, Rooms, and Services as Spaces.

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

Arbitrary string identifier.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Story.Rooms[get]

List of Rooms designated as occupiable rooms.

#### List<Space> RoomKit.Story.RoomsAsSpaces[get]

List of Spaces created from [Room](#AAAAAAAAAC) characteristics within the Rooms list.

#### Color RoomKit.Story.RoomsColor[set]

Sets the Rooms Space rendering color.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Story.Services[get]

A list of Rooms designated as building services.

#### List<Space> RoomKit.Story.ServicesAsSpaces[get]

List of Spaces created from [Room](#AAAAAAAAAC) characteristics within the Services list.

#### Color RoomKit.Story.ServicesColor[set]

Sets the Services Space rendering color.

#### Floor RoomKit.Story.Slab[get]

Concrete Floor created from [Story](#AAAAAAAAAG) and Slab characteristics.

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

RoomKit/Story.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. N, S, E, and W define middle points on each orthogonal side of the box. NE, NW, SE, and SW correspond to the corners of the box. Other compass points define points along the relevant side between the cardinal and corner points. C corresponds to the center of the box.

### Public Member Functions

[TopoBox](#AAAAAAAAEP) (Polygon polygon)

*Constructor creates a new mathematical bounding box from the supplied Polygon and populates all orientation points.*

Vector3 [PointBy](#AAAAAAAAEQ) ([Orient](#AAAAAAAAAJ) orient)

*Returns the requested bounding box location by orientation.*

Vector3 [PointOpposite](#AAAAAAAAER) ([Orient](#AAAAAAAAAJ) orient)

*Returns the reciprocal bounding box location by orientation.*

### Properties

Vector3 [C](#AAAAAAAAES) [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](#AAAAAAAAFJ) [get]

*X and Y dimensions of the* [*TopoBox*](#AAAAAAAAAH) *perimeter.*

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. N, S, E, and W define middle points on each orthogonal side of the box. NE, NW, SE, and SW correspond to the corners of the box. Other compass points define points along the relevant side between the cardinal and corner points. C corresponds to the center of the box.

### Constructor & Destructor Documentation

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

Constructor creates a new mathematical bounding box from the supplied Polygon and populates all orientation points.

##### Returns:

A new [TopoBox](#AAAAAAAAAH).

### Member Function Documentation

#### Vector3 RoomKit.TopoBox.PointBy ([Orient](#AAAAAAAAAJ) *orient*)

Returns the requested bounding box location by orientation.

##### Parameters:

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

##### Returns:

A Vector3 point.

#### Vector3 RoomKit.TopoBox.PointOpposite ([Orient](#AAAAAAAAAJ) *orient*)

Returns the reciprocal bounding box location by orientation.

##### Parameters:

|  |  |
| --- | --- |
| *orient* | The Orient value to find the reciprocal point. |

##### Returns:

A Vector3 point.

### Property Documentation

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

Vector3 location identifiers corresponding to points on the box perimeter.

#### double RoomKit.TopoBox.SizeX[get]

X and Y dimensions of the [TopoBox](#AAAAAAAAAH) perimeter.

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

RoomKit/TopoBox.cs

# Index

INDEX