# RoomKit Documentation

## RoomKit Namespace Reference

### Classes

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 **Messages**

Common exception messages.

class **Place**

Rooms 2D Polygons in various spatial relationships to each other.

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)

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

class [Story](#AAAAAAAAAG)

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

class [Tower](#AAAAAAAAAH)

### Enumerations

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

A list of box corners as compass designations. NE = maximum X and Y corner. SE = maximum X and minimum Y corner. SW = minimum X and Y corner. NW = minimum X and maximum Y corner.

### Enumeration Type Documentation

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

A list of box corners as compass designations. NE = maximum X and Y corner. SE = maximum X and minimum Y corner. SW = minimum X and Y corner. NW = minimum X and maximum Y corner.

# 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](#AAAAAAAAAJ) (Polygon polygon, double xInterval=1, double yInterval=1, int randomSeed=1)

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

void [Allocate](#AAAAAAAAAK) (Polygon polygon)

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

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

Allocates points in the grid falling within the supplied Polygons.

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

Returns the allocated grid point nearest to the supplied point.

Vector3 [AllocatedRandom](#AAAAAAAAAN) ()

Returns a random allocated point.

Vector3 [AvailableMax](#AAAAAAAAAO) ()

Returns the maximum available grid point.

Vector3 [AvailableMin](#AAAAAAAAAP) ()

Returns the minimum available grid point.

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

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

Vector3 [AvailableRandom](#AAAAAAAAAR) ()

Returns a random available grid point.

### Properties

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

The list of vector3 allocated points.

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

The list of Vector3 points available for allocation.

Polygon **Perimeter** [get, set]

### 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, int randomSeed = 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](#AAAAAAAAAB).

### 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.

#### 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](#AAAAAAAAAV) ()

Constructor setting all internal variables to default values to create a 1.0 x 1.0 x 1.0 white cube with no required adjacencies placed on the zero plane with an empty string, null perimeter, and an integer TypeID of -1.

Polygon [MoveFromTo](#AAAAAAAAAW) (Vector3 from, Vector3 to)

Moves the [Room](#AAAAAAAAAC) along a 3D vector calculated between the supplied Vector3 points.

bool [Rotate](#AAAAAAAAAX) (Vector3 pivot, double angle)

Rotates the [Room](#AAAAAAAAAC) Perimeter in the horizontal plane around the supplied pivot point.

bool [SetDimensions](#AAAAAAAAAY) (Vector3 xyz, Vector3 moveTo=null)

Creates and sets a rectangular [Room](#AAAAAAAAAC) Perimeter, Height, and southwest corner location with a supplied vectors. Sets the DesignX and DesignY properties.

bool [SetPerimeter](#AAAAAAAAAZ) (Vector3 moveTo=null)

Creates and sets a rectangular [Room](#AAAAAAAAAC) Perimeter with dimensions derived from [Room](#AAAAAAAAAC) characteristics with its southwest corner at the origin or at the 2D location implied by the supplied Vector3.

bool [SetPerimeter](#AAAAAAAABA) (double area, double ratio=1.5, Vector3 moveTo=null)

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.

bool [SetPerimeter](#AAAAAAAABB) (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.

bool [SetPerimeter](#AAAAAAAABC) (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](#AAAAAAAABD) [get, set]

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

double [Area](#AAAAAAAABE) [get]

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

double [AreaVariance](#AAAAAAAABF) [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](#AAAAAAAABG) [get]

A Space created from [Room](#AAAAAAAAAC) characteristics. Adds properties to the Space recording Name TypeID as Type DesignArea as Design Area DesignX as Design Length DesignY as Design Width Perimeter.Area as Area Elevation Height

Color **Color** [get, set]

double **DesignArea** [get, set]

double [DesignLength](#AAAAAAAABJ) [get, set]

Desired x-axis dimension of this [Room](#AAAAAAAAAC).

double [DesignWidth](#AAAAAAAABK) [get, set]

Desired y-axis dimension of this [Room](#AAAAAAAAAC).

double **DesignRatio** [get, set]

bool [DesignSet](#AAAAAAAABM) [get]

Returns true if both DesignLength and DesignWidth are positive values.

Vector3 **DesignXYZ** [get, set]

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

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

double [Height](#AAAAAAAABP) [get, set]

Height of the [Room](#AAAAAAAAAC) prism. Set ignores non-positive values.

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

Arbitrary string identifier for this [Room](#AAAAAAAAAC) instance.

Polygon **Perimeter** [get, set]

bool [Placed](#AAAAAAAABS) [get, set]

Manual flag to record if the [Room](#AAAAAAAAAC) has been placed in its final position.

double [SizeX](#AAAAAAAABT) [get]

X dimensions of the [Room](#AAAAAAAAAC) Perimeter orthogonal bounding box.

double [SizeY](#AAAAAAAABU) [get]

X dimensions of the [Room](#AAAAAAAAAC) Perimeter orthogonal bounding box.

int [TypeID](#AAAAAAAABV) [get, set]

Arbitrary integer identifier of this instance..

string [UniqueID](#AAAAAAAABW) [get]

UUID for this instance, set on initialization.

### Detailed Description

A data structure recording room characteristics.

### Constructor & Destructor Documentation

#### RoomKit.Room.Room ()

Constructor setting all internal variables to default values to create a 1.0 x 1.0 x 1.0 white cube with no required adjacencies placed on the zero plane with an empty string, null perimeter, and an integer TypeID of -1.

### Member Function Documentation

#### Polygon RoomKit.Room.MoveFromTo (Vector3 from, Vector3 to)

Moves the [Room](#AAAAAAAAAC) along a 3D vector calculated between the supplied Vector3 points.

##### Parameters:

|  |  |
| --- | --- |
| from | Vector3 base point of the move. |
| to | Vector3 target point of the move. |

##### Returns:

A Polygon represeting the [Room](#AAAAAAAAAC)'s new Perimeter.

#### bool RoomKit.Room.Rotate (Vector3 pivot, double angle)

Rotates the [Room](#AAAAAAAAAC) Perimeter in the horizontal plane around the supplied pivot point.

##### Parameters:

|  |  |
| --- | --- |
| pivot | Vector3 point around which the [Room](#AAAAAAAAAC) Perimeter will be rotated. |
| angle | Angle in degrees to rotate the Perimeter. |

##### Returns:

True if the Perimeter is successfully rotated.

#### bool RoomKit.Room.SetDimensions (Vector3 xyz, Vector3 moveTo = null)

Creates and sets a rectangular [Room](#AAAAAAAAAC) Perimeter, Height, and southwest corner location with a supplied vectors. Sets the DesignX and DesignY properties.

##### Parameters:

|  |  |
| --- | --- |
| xyz | Vector3 dimensions of a new Polygon Perimeter. If xy.Z is > 0.0, sets the height of the [Room](#AAAAAAAAAC). |
| moveTo | Vector3 location of the new Polygon's southwest corner. |

##### Returns:

True if the Perimeter is successfully set.

#### bool RoomKit.Room.SetPerimeter (Vector3 moveTo = null)

Creates and sets a rectangular [Room](#AAAAAAAAAC) Perimeter with dimensions derived from [Room](#AAAAAAAAAC) characteristics with its southwest corner at the origin or at the 2D location implied by the supplied Vector3.

##### Returns:

True if the Perimeter is successfully set.

#### bool RoomKit.Room.SetPerimeter (double area, double ratio = 1.5, Vector3 moveTo = null)

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:

|  |  |
| --- | --- |
| area | Area override for the new [Room](#AAAAAAAAAC) Perimeter. If zero, defaults to the value of DesignArea. |
| ratio | Desired ratio of X to Y [Room](#AAAAAAAAAC) dimensions. |
| moveTo | Vector3 location of the new Polygon's southwest corner. |

##### Returns:

True if the Perimeter is successfully set.

#### bool RoomKit.Room.SetPerimeter (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:

True if the Perimeter is successfully set.

#### bool RoomKit.Room.SetPerimeter (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:

True if the Perimeter is successfully set.

### 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. Adds properties to the Space recording Name TypeID as Type DesignArea as Design Area DesignX as Design Length DesignY as Design Width Perimeter.Area as Area Elevation Height

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

Desired x-axis dimension of this [Room](#AAAAAAAAAC).

#### bool RoomKit.Room.DesignSet[get]

Returns true if both DesignLength and DesignWidth are positive values.

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

Desired y-axis dimension of this [Room](#AAAAAAAAAC).

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

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

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

Height of the [Room](#AAAAAAAAAC) prism. Set ignores non-positive values.

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

Arbitrary string identifier for this [Room](#AAAAAAAAAC) instance.

#### bool RoomKit.Room.Placed[get], [set]

Manual flag to record if the [Room](#AAAAAAAAAC) has been placed in its final position.

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

X dimensions of the [Room](#AAAAAAAAAC) Perimeter orthogonal bounding box.

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

X dimensions of the [Room](#AAAAAAAAAC) Perimeter orthogonal bounding box.

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

Arbitrary integer identifier of this instance..

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

UUID for this 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](#AAAAAAAABX) ()

Creates an empty group of Rooms.

void [MoveFromTo](#AAAAAAAABY) (Vector3 from, Vector3 to)

Moves all Rooms in the [RoomGroup](#AAAAAAAAAD) and the [RoomGroup](#AAAAAAAAAD) Perimeter along a 3D vector calculated between the supplied Vector3 points.

void [Rotate](#AAAAAAAABZ) (Vector3 pivot, double angle)

Rotates the [RoomGroup](#AAAAAAAAAD) Perimeter and Rooms in the horizontal plane around the supplied pivot point.

void [SetColor](#AAAAAAAACA) (Color color)

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

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

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

bool [RoomsByDivision](#AAAAAAAACC) (int xRooms=1, int yRooms=1, double height=3.0, string name="")

Clears the current Rooms list and creates new Rooms defined by orthogonal x- and y-axis divisions of the [RoomGroup](#AAAAAAAAAD) Perimeter.

### Properties

double [AreaAvailable](#AAAAAAAACD) [get]

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

double [AreaPlaced](#AAAAAAAACE) [get]

Area allocated within the [RoomGroup](#AAAAAAAAAD).

double **Elevation** [get, set]

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

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

Polygon **Perimeter** [get, set]

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

List of Rooms placed within the Perimeter.

List< Polygon > [RoomsAsPolygons](#AAAAAAAACJ) [get]

List of all [Room](#AAAAAAAAAC) perimeters as Polygons.

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

List of all Rooms as Spaces.

double [SizeX](#AAAAAAAACL) [get]

X dimension of the Perimeter orthogonal bounding box.

double [SizeY](#AAAAAAAACM) [get]

Y dimension of the Perimeter orthogonal bounding box.

int [TypeID](#AAAAAAAACN) [get, set]

Arbitrary integer identifier of this instance..

string [UniqueID](#AAAAAAAACO) [get]

UUID for this [RoomGroup](#AAAAAAAAAD) instance, set on initialization.

### Detailed Description

Creates and manages Rooms within a perimeter.

### Constructor & Destructor Documentation

#### RoomKit.RoomGroup.RoomGroup ()

Creates an empty group of Rooms.

##### Returns:

A new [RoomGroup](#AAAAAAAAAD).

### Member Function Documentation

#### void RoomKit.RoomGroup.MoveFromTo (Vector3 from, Vector3 to)

Moves all Rooms in the [RoomGroup](#AAAAAAAAAD) and the [RoomGroup](#AAAAAAAAAD) Perimeter along a 3D vector calculated between the supplied Vector3 points.

##### Parameters:

|  |  |
| --- | --- |
| from | Vector3 base point of the move. |
| to | Vector3 target point of the move. |

##### Returns:

None.

#### bool RoomKit.RoomGroup.RoomsByDivision (int xRooms = 1, int yRooms = 1, double height = 3.0, string name = "")

Clears the current Rooms list and creates new Rooms defined by orthogonal x- and y-axis divisions of the [RoomGroup](#AAAAAAAAAD) Perimeter.

##### Parameters:

|  |  |
| --- | --- |
| xRooms | The quantity of Rooms along orthogonal x-axis. Must be positive. |
| yRooms | The quantity of Rooms along orthogonal y-axis. Must be positive. |

##### Returns:

True if the Rooms are created.

#### void RoomKit.RoomGroup.Rotate (Vector3 pivot, double angle)

Rotates the [RoomGroup](#AAAAAAAAAD) Perimeter and Rooms in the horizontal plane around the supplied pivot point.

##### Parameters:

|  |  |
| --- | --- |
| pivot | Vector3 point around which the [Room](#AAAAAAAAAC) Perimeter will be rotated. |
| angle | Angle in degrees to rotate the Perimeter. |

##### Returns:

None.

#### void RoomKit.RoomGroup.SetColor (Color color)

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

##### Parameters:

|  |  |
| --- | --- |
| color | The new color 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.AreaAvailable[get]

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

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

Area allocated within the [RoomGroup](#AAAAAAAAAD).

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

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

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

List of Rooms placed within the Perimeter.

#### List<Polygon> RoomKit.RoomGroup.RoomsAsPolygons[get]

List of all [Room](#AAAAAAAAAC) perimeters as Polygons.

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

List of all Rooms as Spaces.

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

X dimension of the Perimeter orthogonal bounding box.

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

Y dimension of the Perimeter orthogonal bounding box.

#### int RoomKit.RoomGroup.TypeID[get], [set]

Arbitrary integer identifier of this instance..

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

UUID for this [RoomGroup](#AAAAAAAAAD) instance, set on initialization.

#### 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](#AAAAAAAACP) (Line row)

Constructor initializes the [RoomRow](#AAAAAAAAAE) with a new Line.

[RoomRow](#AAAAAAAACQ) (Vector3 start, Vector3 end)

Constructor initializes the [RoomRow](#AAAAAAAAAE) with line endpoints.

bool [AddRoom](#AAAAAAAACR) ([Room](#AAAAAAAAAC) room, Polygon within=null, IList< Polygon > among=null)

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 with which it cannot intersect.

void [MoveFromTo](#AAAAAAAACS) (Vector3 from, Vector3 to)

Moves all Rooms in the [RoomRow](#AAAAAAAAAE) and the [RoomRow](#AAAAAAAAAE) Row along a 3D vector calculated between the supplied Vector3 points.

void [Rotate](#AAAAAAAACT) (Vector3 pivot, double angle)

Rotates the [RoomRow](#AAAAAAAAAE) Row and Rooms in the horizontal plane around the supplied pivot point.

void [SetColor](#AAAAAAAACU) (Color color)

Uniformly sets the color of all Rooms in the [RoomRow](#AAAAAAAAAE).

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

Uniformly sets the height of all Rooms in the [RoomRow](#AAAAAAAAAE).

### Properties

double [AreaPlaced](#AAAAAAAACW) [get]

Aggregate area of the Rooms placed on this Row.

double [AvailableLength](#AAAAAAAACX) [get]

Unallocated length of the [RoomRow](#AAAAAAAAAE).

Polygon [Circulation](#AAAAAAAACY) [get]

Circulation envelope around the row.

double **CirculationWidth** [get, set]

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

Depth of the deepest room along the Row.

double **Elevation** [get, set]

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

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

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

List of Rooms placed along the Row.

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

List of all [Room](#AAAAAAAAAC) perimeters as Polygons.

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

List of all Rooms as Spaces.

Line [Row](#AAAAAAAADG) [get]

Line along which Rooms can be placed.

double [SizeX](#AAAAAAAADH) [get]

X dimension of the Circulation orthogonal bounding box.

double [SizeY](#AAAAAAAADI) [get]

Y dimension of the Circulation orthogonal bounding box.

int [TypeID](#AAAAAAAADJ) [get, set]

Arbitrary integer identifier of this instance..

string [UniqueID](#AAAAAAAADK) [get]

UUID for this [RoomRow](#AAAAAAAAAE) instance, set on initialization.

### Detailed Description

Creates and manages Rooms placed along a line.

### Constructor & Destructor Documentation

#### RoomKit.RoomRow.RoomRow (Line row)

Constructor initializes the [RoomRow](#AAAAAAAAAE) with a new Line.

#### RoomKit.RoomRow.RoomRow (Vector3 start, Vector3 end)

Constructor initializes the [RoomRow](#AAAAAAAAAE) with line endpoints.

### Member Function Documentation

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

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 with which it cannot intersect.

##### Parameters:

|  |  |
| --- | --- |
| room | [Room](#AAAAAAAAAC) from which to derive the Polygon to place. |
| within | Polygon perimeter within which a new [Room](#AAAAAAAAAC) must fit. |
| among | List of Polygon perimeters the new [Room](#AAAAAAAAAC) cannot intersect. |

##### Returns:

True if the room was successfully placed.

#### void RoomKit.RoomRow.MoveFromTo (Vector3 from, Vector3 to)

Moves all Rooms in the [RoomRow](#AAAAAAAAAE) and the [RoomRow](#AAAAAAAAAE) Row along a 3D vector calculated between the supplied Vector3 points.

##### Parameters:

|  |  |
| --- | --- |
| from | Vector3 base point of the move. |
| to | Vector3 target point of the move. |

##### Returns:

None.

#### void RoomKit.RoomRow.Rotate (Vector3 pivot, double angle)

Rotates the [RoomRow](#AAAAAAAAAE) Row and Rooms in the horizontal plane around the supplied pivot point.

##### Parameters:

|  |  |
| --- | --- |
| pivot | Vector3 point around which the [Room](#AAAAAAAAAC) Perimeter will be rotated. |
| angle | Angle in degrees to rotate the Perimeter. |

##### Returns:

None.

#### void RoomKit.RoomRow.SetColor (Color color)

Uniformly sets the color of all Rooms in the [RoomRow](#AAAAAAAAAE).

##### Parameters:

|  |  |
| --- | --- |
| color | New color of the Rooms. |

##### Returns:

None.

#### void RoomKit.RoomRow.SetHeight (double height)

Uniformly sets the height of all Rooms in the [RoomRow](#AAAAAAAAAE).

##### Parameters:

|  |  |
| --- | --- |
| elevation | New height of the Rooms. |

##### Returns:

None.

### Property Documentation

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

Aggregate area of the Rooms placed on this Row.

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

Unallocated length of the [RoomRow](#AAAAAAAAAE).

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

Circulation envelope around the row.

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

Depth of the deepest room along the Row.

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

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

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

List of Rooms placed along the Row.

#### List<Polygon> RoomKit.RoomRow.RoomsAsPolygons[get]

List of all [Room](#AAAAAAAAAC) perimeters as Polygons.

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

List of all Rooms as Spaces.

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

Line along which Rooms can be placed.

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

X dimension of the Circulation orthogonal bounding box.

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

Y dimension of the Circulation orthogonal bounding box.

#### int RoomKit.RoomRow.TypeID[get], [set]

Arbitrary integer identifier of this instance..

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

UUID for this [RoomRow](#AAAAAAAAAE) instance, set on initialization.

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

RoomKit/RoomRow.cs

## RoomKit.Scope Class Reference

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

### Public Member Functions

[Scope](#AAAAAAAADL) ()

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

[Room](#AAAAAAAAAC) [FindByDesignArea](#AAAAAAAADM) (double area, bool unplaced=true)

Finds the first Occupant [Room](#AAAAAAAAAC) with the DesignArea value closest to the supplied area. C

[Room](#AAAAAAAAAC) [FindByDesignXY](#AAAAAAAADN) (double designLength, double designWidth, bool unplaced=true)

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

[Room](#AAAAAAAAAC) [FindByTypeID](#AAAAAAAADO) (int typeID, bool unplaced=true)

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

### Properties

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

List of Rooms designated as circulation.

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

List of Rooms designated for occupation, rather than circulation.

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

List of Rooms designated for building services.

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

List of Rooms intended as a series of tenant space containers of other Rooms.

List< Polygon > [AllocatedAsPolygons](#AAAAAAAADT) [get]

List of allocated Circulation, Occupant, and Service [Room](#AAAAAAAAAC) Perimeters as Polygons.

double [AreaDesignAvailable](#AAAAAAAADU) [get]

Area available for horizontal circulation.

double [AreaDesignCirculation](#AAAAAAAADV) [get]

Intended aggregate area of all Occupant Rooms.

double [AreaDesignOccupant](#AAAAAAAADW) [get]

Intended aggregate area of all Occupant Rooms.

double [AreaCirculation](#AAAAAAAADX) [get]

Allocated aggregate area of all placed Circulation Rooms.

double [AreaOccupant](#AAAAAAAADY) [get]

Allocated aggregate area of all placed Occupant Rooms.

double [AreaService](#AAAAAAAADZ) [get]

Aggregate area of all Services Rooms.

double [AreaTenant](#AAAAAAAAEA) [get]

Aggregate area of all occupiable Tenant Rooms.

List< Polygon > [CirculationAsPolygons](#AAAAAAAAEB) [get]

List of all Circulation [Room](#AAAAAAAAAC) Perimeters as Polygons.

List< Polygon > [OccupantAsPolygons](#AAAAAAAAEC) [get]

List of all Occupant [Room](#AAAAAAAAAC) Perimeters as Polygons.

List< Polygon > [ServiceAsPolygons](#AAAAAAAAED) [get]

List of all Service [Room](#AAAAAAAAAC) Perimeters as Polygons.

List< Polygon > [TenantAsPolygons](#AAAAAAAAEE) [get]

List of all Tenant [Room](#AAAAAAAAAC) Perimeter Polygons.

List< [Room](#AAAAAAAAAC) > [Placed](#AAAAAAAAEF) [get]

List of all Rooms marked as Placed.

bool [PlacedAll](#AAAAAAAAEG) [get]

Returns whether all Occupant Rooms have been Placed.

double [PlacedQuantity](#AAAAAAAAEH) [get]

The quantity of placed Rooms.

double [RatioCirculation](#AAAAAAAAEI) [get]

Returns the ratio of the aggregate area of all Occupant Rooms against the Circulation area.

double [RatioDesignCirculation](#AAAAAAAAEJ) [get]

Returns the ratio of the aggregate area of all designed Occupant Rooms against the designed Circulation area.

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

Returns all Rooms with Placed = false.

double [UnplacedQuantity](#AAAAAAAAEL) [get]

The quantity of unplaced Rooms.

### Detailed Description

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.FindByDesignArea (double area, bool unplaced = true)

Finds the first Occupant [Room](#AAAAAAAAAC) with the DesignArea value closest to the supplied area. C

##### Parameters:

|  |  |
| --- | --- |
| area | Area to match from the list of all Occupant [Room](#AAAAAAAAAC) definitions. |

##### Returns:

A [Room](#AAAAAAAAAC).

#### [Room](#AAAAAAAAAC) RoomKit.Scope.FindByDesignXY (double designLength, double designWidth, bool unplaced = true)

Finds the first Occupant [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:

A [Room](#AAAAAAAAAC).

#### [Room](#AAAAAAAAAC) RoomKit.Scope.FindByTypeID (int typeID, bool unplaced = true)

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

##### Parameters:

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

##### Returns:

A [Room](#AAAAAAAAAC).

### Property Documentation

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

List of allocated Circulation, Occupant, and Service [Room](#AAAAAAAAAC) Perimeters as Polygons.

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

Allocated aggregate area of all placed Circulation Rooms.

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

Area available for horizontal circulation.

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

Intended aggregate area of all Occupant Rooms.

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

Intended aggregate area of all Occupant Rooms.

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

Allocated aggregate area of all placed Occupant Rooms.

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

Aggregate area of all Services Rooms.

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

Aggregate area of all occupiable Tenant Rooms.

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

List of Rooms designated as circulation.

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

List of all Circulation [Room](#AAAAAAAAAC) Perimeters as Polygons.

##### Returns:

A list of Polygons.

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

List of Rooms designated for occupation, rather than circulation.

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

List of all Occupant [Room](#AAAAAAAAAC) Perimeters as Polygons.

##### Returns:

A list of Polygons.

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

List of all Rooms marked as Placed.

##### Returns:

A list of Rooms.

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

Returns whether all Occupant Rooms have been Placed.

##### Returns:

Returns true if each [Room](#AAAAAAAAAC) in Occupant has been marked with [Room.Placed](#AAAAAAAABS) = true.

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

The quantity of placed Rooms.

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

Returns the ratio of the aggregate area of all Occupant Rooms against the Circulation area.

##### Returns:

A list of Rooms.

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

Returns the ratio of the aggregate area of all designed Occupant Rooms against the designed Circulation area.

##### Returns:

A list of Rooms.

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

List of Rooms designated for building services.

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

List of all Service [Room](#AAAAAAAAAC) Perimeters as Polygons.

##### Returns:

A list of Polygons.

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

List of Rooms intended as a series of tenant space containers of other Rooms.

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

List of all Tenant [Room](#AAAAAAAAAC) Perimeter Polygons.

##### Returns:

A list of Polygons.

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

Returns all Rooms with Placed = false.

##### Returns:

A list of Rooms.

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

The quantity of unplaced 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](#AAAAAAAAEM) ()

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.

bool [AddCorridor](#AAAAAAAAEN) ([Room](#AAAAAAAAAC) room, bool fit=true)

Adds a [Room](#AAAAAAAAAC) to the Corridors list.

bool [AddExclusion](#AAAAAAAAEO) ([Room](#AAAAAAAAAC) room, bool fit=true)

Adds a [Room](#AAAAAAAAAC) to the Exclusions list.

bool [AddRoom](#AAAAAAAAEP) ([Room](#AAAAAAAAAC) room, bool fit=true)

Adds a [Room](#AAAAAAAAAC) to the Rooms list.

bool [AddService](#AAAAAAAAEQ) ([Room](#AAAAAAAAAC) room, bool fit=true)

Adds a [Room](#AAAAAAAAAC) to the Services list.

double [AreaByName](#AAAAAAAAER) (string name)

Returns the aggregate area of all Rooms with a supplied name.

void [MoveFromTo](#AAAAAAAAES) (Vector3 from, Vector3 to)

Moves all Rooms in the [Story](#AAAAAAAAAG) and the [Story](#AAAAAAAAAG) Envelope along a 3D vector calculated between the supplied Vector3 points.

bool [RoomsByDivision](#AAAAAAAAET) (int xRooms=1, int yRooms=1, double height=3.0, double setback=0.0, string name="", Color color=null, bool fit=true)

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.

List< [Room](#AAAAAAAAAC) > [RoomsByName](#AAAAAAAAEU) (string name)

Returns a list of Rooms with a specific name.

void [Rotate](#AAAAAAAAEV) (Vector3 pivot, double angle)

Rotates the [Story](#AAAAAAAAAG) Perimeter and Rooms in the horizontal plane around the supplied pivot point.

### Properties

double [Area](#AAAAAAAAEW) [get]

Area of the perimeter.

double [AreaAvailable](#AAAAAAAAEX) [get]

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

double [AreaPlaced](#AAAAAAAAEY) [get]

Area allocated to Corridors, Rooms, and Services.

Color **Color** [get, set]

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

List of Rooms designated as cooridors.

List< Polygon > [CorridorsAsPolygons](#AAAAAAAAFB) [get]

Polygons representing Corridors. Rooms Perimeters in the [Story](#AAAAAAAAAG) conform to Corridor Perimeters.

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

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

Color [CorridorsColor](#AAAAAAAAFD) [set]

Sets the Corridors color.

double **Elevation** [get, set]

[Room](#AAAAAAAAAC) [Envelope](#AAAAAAAAFF) [get]

[Room](#AAAAAAAAAC) representing the [Story](#AAAAAAAAAG) envelope.

Polygon [EnvelopeAsPolygon](#AAAAAAAAFG) [get]

Polygon representation of the [Story](#AAAAAAAAAG) Perimeter.

Space [EnvelopeAsSpace](#AAAAAAAAFH) [get]

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

List< [Room](#AAAAAAAAAC) > [Exclusions](#AAAAAAAAFI) [get]

Rooms representing areas that must not be intersected, but which will not be available as Spaces. All other [Room](#AAAAAAAAAC) Perimeters in the [Story](#AAAAAAAAAG) conform to Exclusion [Room](#AAAAAAAAAC) Perimeters.

List< Polygon > [ExclusionsAsPolygons](#AAAAAAAAFJ) [get]

Polygons representing areas that must not be intersected. All other [Room](#AAAAAAAAAC) Perimeters in the [Story](#AAAAAAAAAG) conform to Exclusion [Room](#AAAAAAAAAC) Perimeters.

double **Height** [get, set]

double [HeightInteriors](#AAAAAAAAFL) [set]

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

IList< Polygon > [InteriorsAsPolygons](#AAAAAAAAFM) [get]

Returns all Corridors, Exclusions, Rooms, and Services as Polygons.

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

Returns all Corridors, Rooms, and Services as Spaces.

bool [IsBasement](#AAAAAAAAFO) [get, set]

Identifies whether this story represents a base ment level.

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

Arbitrary string identifier.

Polygon **Perimeter** [get, set]

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

List of Rooms designated as occupiable rooms.

List< Polygon > [RoomsAsPolygons](#AAAAAAAAFS) [get]

Polygons representing Services. Corridors and Rooms Perimeters in the [Story](#AAAAAAAAAG) conform to Service [Room](#AAAAAAAAAC) Perimeters.

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

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

Color [RoomsColor](#AAAAAAAAFU) [set]

Sets the Rooms rendering color.

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

A list of Rooms designated as building services.

List< Polygon > [ServicesAsPolygons](#AAAAAAAAFW) [get]

Polygons representing Services. Corridors and Rooms Perimeters in the [Story](#AAAAAAAAAG) conform to Service [Room](#AAAAAAAAAC) Perimeters.

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

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

Color [ServicesColor](#AAAAAAAAFY) [set]

Sets the Services Space rendering color.

Floor [Slab](#AAAAAAAAFZ) [get]

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

double **SlabThickness** [get, set]

int [TypeID](#AAAAAAAAGB) [get, set]

Arbitrary integer identifier of this instance..

string [UniqueID](#AAAAAAAAGC) [get]

UUID for this instance, set on initialization.

### 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.

##### Returns:

A new [Story](#AAAAAAAAAG).

### Member Function Documentation

#### bool RoomKit.Story.AddCorridor ([Room](#AAAAAAAAAC) room, bool fit = true)

Adds a [Room](#AAAAAAAAAC) to the Corridors list.

##### Parameters:

|  |  |
| --- | --- |
| room | [Room](#AAAAAAAAAC) to add. |
| fit | Indicates whether the new room should mutually fit to other [Story](#AAAAAAAAAG) features. Default is true. |

##### Returns:

True if one or more rooms were added to the [Story](#AAAAAAAAAG).

#### bool RoomKit.Story.AddExclusion ([Room](#AAAAAAAAAC) room, bool fit = true)

Adds a [Room](#AAAAAAAAAC) to the Exclusions list.

##### Parameters:

|  |  |
| --- | --- |
| room | [Room](#AAAAAAAAAC) to add. |
| fit | Indicates whether the new room should mutually fit to other [Story](#AAAAAAAAAG) features. Default is true. |

##### Returns:

True if one or more rooms were added to the [Story](#AAAAAAAAAG).

#### bool RoomKit.Story.AddRoom ([Room](#AAAAAAAAAC) room, bool fit = true)

Adds a [Room](#AAAAAAAAAC) to the Rooms list.

##### Parameters:

|  |  |
| --- | --- |
| room | [Room](#AAAAAAAAAC) to add. |
| fit | Indicates whether the new [Room](#AAAAAAAAAC) should mutually fit to other [Story](#AAAAAAAAAG) features. Default is true. |

##### Returns:

True if one or more Rooms were added to the [Story](#AAAAAAAAAG).

#### bool RoomKit.Story.AddService ([Room](#AAAAAAAAAC) room, bool fit = true)

Adds a [Room](#AAAAAAAAAC) to the Services list.

##### Parameters:

|  |  |
| --- | --- |
| room | [Room](#AAAAAAAAAC) to add. |
| fit | Indicates whether the new [Room](#AAAAAAAAAC) should mutually fit to other [Story](#AAAAAAAAAG) features. Default is true. |

##### Returns:

True if one or more Rooms were added to the [Story](#AAAAAAAAAG).

#### double RoomKit.Story.AreaByName (string name)

Returns the aggregate area of all Rooms with a supplied name.

##### Parameters:

|  |  |
| --- | --- |
| name | Name of the Rooms to retrieve. |

##### Returns:

None.

///

#### void RoomKit.Story.MoveFromTo (Vector3 from, Vector3 to)

Moves all Rooms in the [Story](#AAAAAAAAAG) and the [Story](#AAAAAAAAAG) Envelope along a 3D vector calculated between the supplied Vector3 points.

##### Parameters:

|  |  |
| --- | --- |
| from | Vector3 base point of the move. |
| to | Vector3 target point of the move. |

##### Returns:

None.

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

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.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Story.RoomsByName (string name)

Returns a list of Rooms with a specific name.

##### Parameters:

|  |  |
| --- | --- |
| name | Name of the rooms to retrieve. |

##### Returns:

None.

///

#### void RoomKit.Story.Rotate (Vector3 pivot, double angle)

Rotates the [Story](#AAAAAAAAAG) Perimeter and Rooms in the horizontal plane around the supplied pivot point.

##### Parameters:

|  |  |
| --- | --- |
| pivot | Vector3 point around which the [Room](#AAAAAAAAAC) Perimeter will be rotated. |
| angle | Angle in degrees to rotate the Perimeter. |

##### 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<Polygon> RoomKit.Story.CorridorsAsPolygons[get]

Polygons representing Corridors. Rooms Perimeters in the [Story](#AAAAAAAAAG) conform to Corridor Perimeters.

#### 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.

#### [Room](#AAAAAAAAAC) RoomKit.Story.Envelope[get]

[Room](#AAAAAAAAAC) representing the [Story](#AAAAAAAAAG) envelope.

#### Polygon RoomKit.Story.EnvelopeAsPolygon[get]

Polygon representation of the [Story](#AAAAAAAAAG) Perimeter.

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

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

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

Rooms representing areas that must not be intersected, but which will not be available as Spaces. All other [Room](#AAAAAAAAAC) Perimeters in the [Story](#AAAAAAAAAG) conform to Exclusion [Room](#AAAAAAAAAC) Perimeters.

#### List<Polygon> RoomKit.Story.ExclusionsAsPolygons[get]

Polygons representing areas that must not be intersected. All other [Room](#AAAAAAAAAC) Perimeters in the [Story](#AAAAAAAAAG) conform to Exclusion [Room](#AAAAAAAAAC) Perimeters.

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

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

#### IList<Polygon> RoomKit.Story.InteriorsAsPolygons[get]

Returns all Corridors, Exclusions, Rooms, and Services as Polygons.

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

Returns all Corridors, Rooms, and Services as Spaces.

#### bool RoomKit.Story.IsBasement[get], [set]

Identifies whether this story represents a base ment level.

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

Arbitrary string identifier.

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

List of Rooms designated as occupiable rooms.

#### List<Polygon> RoomKit.Story.RoomsAsPolygons[get]

Polygons representing Services. Corridors and Rooms Perimeters in the [Story](#AAAAAAAAAG) conform to Service [Room](#AAAAAAAAAC) Perimeters.

#### 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 rendering color.

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

A list of Rooms designated as building services.

#### List<Polygon> RoomKit.Story.ServicesAsPolygons[get]

Polygons representing Services. Corridors and Rooms Perimeters in the [Story](#AAAAAAAAAG) conform to Service [Room](#AAAAAAAAAC) Perimeters.

#### 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.

#### int RoomKit.Story.TypeID[get], [set]

Arbitrary integer identifier of this instance..

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

UUID for this instance, set on initialization.

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

RoomKit/Story.cs

## RoomKit.Tower Class Reference

### Public Member Functions

bool [AddCore](#AAAAAAAAGD) (Polygon perimeter, int baseStory=0, double addHeight=0.0, Color color=null)

Adds a new service Core to the [Tower](#AAAAAAAAAH).

double [AreaByName](#AAAAAAAAGE) (string name)

Returns the aggregate area of all Rooms with a supplied name.

void [MoveFromTo](#AAAAAAAAGF) (Vector3 from, Vector3 to)

Moves all Cores and Stories in the [Tower](#AAAAAAAAAH) along a 3D vector calculated between the supplied Vector3 points.

List< [Room](#AAAAAAAAAC) > [RoomsByName](#AAAAAAAAGG) (string name)

Returns a list of Rooms with a specific name.

void [Rotate](#AAAAAAAAGH) (Vector3 pivot, double angle)

Rotates the [Tower](#AAAAAAAAAH) Perimeter and Stories in the horizontal plane around the supplied pivot point.

bool [Stack](#AAAAAAAAGI) ()

Creates the [Tower](#AAAAAAAAAH) by stacking a series of [Story](#AAAAAAAAAG) instances from the [Tower](#AAAAAAAAAH) Elevation.

bool [SetStoryHeight](#AAAAAAAAGJ) (int story, double height, bool interiors=true, bool upward=true)

Sets the height of an index-specified [Story](#AAAAAAAAAG) and relocates Stories above to accommodate the [Story](#AAAAAAAAAG)'s new height.

### Public Attributes

List< [Story](#AAAAAAAAAG) > [Stories](#AAAAAAAAGK) = null

List of all Stories in the [Tower](#AAAAAAAAAH).

### Properties

double [Area](#AAAAAAAAGL) [get]

Returns the aggregate area of all Stories in the [Tower](#AAAAAAAAAH).

Color **Color** [get, set]

List< [Room](#AAAAAAAAAC) > [Cores](#AAAAAAAAGN) [get]

List of all service Cores in the [Tower](#AAAAAAAAAH).

double **Elevation** [get, set]

[Room](#AAAAAAAAAC) [Envelope](#AAAAAAAAGP) [get]

[Room](#AAAAAAAAAC) representing the [Tower](#AAAAAAAAAH) envelope.

Polygon [EnvelopeAsPolygon](#AAAAAAAAGQ) [get]

Polygon representation of the [Story](#AAAAAAAAAG) Perimeter.

Space [EnvelopeAsSpace](#AAAAAAAAGR) [get]

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

int **Floors** [get, set]

double [Height](#AAAAAAAAGT) [get]

Highest point of the highest tower story above the zero plane.

double [HeightLimit](#AAAAAAAAGU) [get, set]

Desired typical [Story](#AAAAAAAAAG) height in the [Tower](#AAAAAAAAAH).

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

Arbitrary string identifier for this [Tower](#AAAAAAAAAH) instance.

Polygon **Perimeter** [get, set]

List< Floor > [Slabs](#AAAAAAAAGX) [get]

List of all Slabs from every [Story](#AAAAAAAAAG) in the [Tower](#AAAAAAAAAH).

List< Space > [Spaces](#AAAAAAAAGY) [get]

List of all Spaces from every [Story](#AAAAAAAAAG) in the [Tower](#AAAAAAAAAH).

double **StoryHeight** [get, set]

double **TargetArea** [get, set]

int [TypeID](#AAAAAAAAHB) [get, set]

Arbitrary integer identifier of this instance..

string [UniqueID](#AAAAAAAAHC) [get]

UUID for this instance, set on initialization.

### Member Function Documentation

#### bool RoomKit.Tower.AddCore (Polygon perimeter, int baseStory = 0, double addHeight = 0.0, Color color = null)

Adds a new service Core to the [Tower](#AAAAAAAAAH).

##### Parameters:

|  |  |
| --- | --- |
| perimeter | Polygon perimeter defining the footprint of the service Core. |
| baseStory | Index of the lowest [Story](#AAAAAAAAAG) whose elevation will serve as the lowest level of the Core. |
| addHeight | Additional height of the Core above the highest [Story](#AAAAAAAAAG). |
| color | Color of the Core when it it is accessed as a Space. |

##### Returns:

True if the Core is successfully added.

#### double RoomKit.Tower.AreaByName (string name)

Returns the aggregate area of all Rooms with a supplied name.

##### Parameters:

|  |  |
| --- | --- |
| name | Name of the Rooms to retrieve. |

##### Returns:

None.

///

#### void RoomKit.Tower.MoveFromTo (Vector3 from, Vector3 to)

Moves all Cores and Stories in the [Tower](#AAAAAAAAAH) along a 3D vector calculated between the supplied Vector3 points.

##### Parameters:

|  |  |
| --- | --- |
| from | Vector3 base point of the move. |
| to | Vector3 target point of the move. |

##### Returns:

None.

#### List<[Room](#AAAAAAAAAC)> RoomKit.Tower.RoomsByName (string name)

Returns a list of Rooms with a specific name.

##### Parameters:

|  |  |
| --- | --- |
| name | Name of the rooms to retrieve. |

##### Returns:

None.

///

#### void RoomKit.Tower.Rotate (Vector3 pivot, double angle)

Rotates the [Tower](#AAAAAAAAAH) Perimeter and Stories in the horizontal plane around the supplied pivot point.

##### Parameters:

|  |  |
| --- | --- |
| pivot | Vector3 point around which the [Room](#AAAAAAAAAC) Perimeter will be rotated. |
| angle | Angle in degrees to rotate the Perimeter. |

##### Returns:

None.

#### bool RoomKit.Tower.SetStoryHeight (int story, double height, bool interiors = true, bool upward = true)

Sets the height of an index-specified [Story](#AAAAAAAAAG) and relocates Stories above to accommodate the [Story](#AAAAAAAAAG)'s new height.

##### Parameters:

|  |  |
| --- | --- |
| story | Index of the [Story](#AAAAAAAAAG) to affect. |
| height | Desired new height of the specified [Story](#AAAAAAAAAG). |
| interiors | If true also sets any Corridors and Rooms in the [Story](#AAAAAAAAAG) to the new Height. |

##### Returns:

True if the [Tower](#AAAAAAAAAH) is successfully stacked.

#### bool RoomKit.Tower.Stack ()

Creates the [Tower](#AAAAAAAAAH) by stacking a series of [Story](#AAAAAAAAAG) instances from the [Tower](#AAAAAAAAAH) Elevation.

##### Parameters:

|  |  |
| --- | --- |
| floors | Desired quantity of stacked Stories to form the [Tower](#AAAAAAAAAH). If greater than zero, overrides and resets the current Floors property. |
| storyHeight | Desired typical [Story](#AAAAAAAAAG) height of the [Tower](#AAAAAAAAAH). If greater than zero, overrides and resets the current StoryHeight property. |
| basement | Whether to consider the lowest floor a basement. |

##### Returns:

True if the [Tower](#AAAAAAAAAH) is successfully stacked.

### Member Data Documentation

#### List<[Story](#AAAAAAAAAG)> RoomKit.Tower.Stories = null

List of all Stories in the [Tower](#AAAAAAAAAH).

### Property Documentation

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

Returns the aggregate area of all Stories in the [Tower](#AAAAAAAAAH).

#### List<[Room](#AAAAAAAAAC)> RoomKit.Tower.Cores[get]

List of all service Cores in the [Tower](#AAAAAAAAAH).

#### [Room](#AAAAAAAAAC) RoomKit.Tower.Envelope[get]

[Room](#AAAAAAAAAC) representing the [Tower](#AAAAAAAAAH) envelope.

#### Polygon RoomKit.Tower.EnvelopeAsPolygon[get]

Polygon representation of the [Story](#AAAAAAAAAG) Perimeter.

#### Space RoomKit.Tower.EnvelopeAsSpace[get]

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

#### double RoomKit.Tower.Height[get]

Highest point of the highest tower story above the zero plane.

#### double RoomKit.Tower.HeightLimit[get], [set]

Desired typical [Story](#AAAAAAAAAG) height in the [Tower](#AAAAAAAAAH).

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

Arbitrary string identifier for this [Tower](#AAAAAAAAAH) instance.

#### List<Floor> RoomKit.Tower.Slabs[get]

List of all Slabs from every [Story](#AAAAAAAAAG) in the [Tower](#AAAAAAAAAH).

#### List<Space> RoomKit.Tower.Spaces[get]

List of all Spaces from every [Story](#AAAAAAAAAG) in the [Tower](#AAAAAAAAAH).

#### int RoomKit.Tower.TypeID[get], [set]

Arbitrary integer identifier of this instance..

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

UUID for this instance, set on initialization.

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

RoomKit/Tower.cs

# Index

INDEX