

# Class List

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

[detail level 1 2]

## ▼ N Cubiquity

### C ByteArray

Provides a simple array of bytes with direct access to each element

### C ColoredCubesVolume

Allows environments to be built from millions of colored cubes

### C ColoredCubesVolumeCollider

Causes the colored cubes volume to have a collision mesh and allows it to participate in collisions

### C ColoredCubesVolumeRenderer

Controls some visual aspects of the colored cubes volume and allows it to be rendered

### C CubiquityException

Thrown to indicate an error has occurred inside the Cubiquity native code library

### C MaterialSet

Represents the combination of materials which a given voxel is composed of

### C Paths

Defines a number of commonly used paths

### C PickVoxelResult

Stores the result of picking a voxel

### C PickSurfaceResult

Stores the result of picking a point on a volume surface

### C Picking

Contains methods for picking directly against the volume data (rather than the mesh representation)

### C QuantizedColor

Stores an *approximate* color value with a limited bit-depth

### C Region

Denotes a region of 3D space, typically representing the bounds for a volume

### C TerrainVolume

Allows the creation of dynamic terrains featuring caves and overhangs

### C TerrainVolumeCollider

Causes the terrain volume to have a collision mesh and allows it to participate in collisions

### C TerrainVolumeRenderer

Controls some visual aspects of the terrain volume and allows it to be rendered

### C Vector3i

A three-dimensional vector type with integer components

### C Volume

Base class representing behaviour common to all volumes

### C VolumeCollider

Causes the volume to have a collision mesh and allows it to participate in collisions

### C VolumeRenderer

Controls some visual aspects of the volume and allows it to be rendered

### C ColoredCubesVolumeData

An implementation of **VolumeData** which stores a

 **TerrainVolumeData**

QuantizedColor for each voxel

An implementation of **VolumeData** which stores a MaterialSet for each voxel

 **VolumeData**

Base class representing the actual 3D grid of voxel values

---