# Better Physics Layers 1.0.0

## API Reference

---

### PhysicsMask.cs

---

public void **SetBelongsTo**(int **layer,** bool **doesBelongTo**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Sets whether this object belongs to the specified layer or not, and updates the physics system to match.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetBelongsTo**(string **layerName**, bool **doesBelongTo**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Sets whether this object belongs to the specified layer or not, and updates the physics system to match.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetBelongsToBitmask**(ulong **bitmask**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Directly sets the belongsTo bitmask, and updates the physics system to match.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetCollidesWith**(int **layer**, bool **doesCollideWith**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Sets whether this object collides with the specified layer or not, and updates the physics system to match.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetCollidesWith**(string **layerName**, bool **doesCollideWith**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Sets whether this object collides with the specified layer or not, and updates the physics system to match.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetCollidesWithBitmask**(ulong **bitmask**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Directly sets the collidesWith bitmask, and updates the physics system to match.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetLayer**(int **layer**, bool **existsInLayer**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Sets whether this object both belongs to the specified layer, and collides with other objects in the same layer.
Combines of SetBelongsTo and SetCollidesWith, for convenience and to allow a simpler usage of the PhysicsMask system.
The regular rules still apply, however, so if this object is added to layer 1 but another object is not set to collide with layer 1 (even if the other object also belongs to layer 1), the objects won't collide.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetLayer**(string **layerName**, bool **existsInLayer**, bool **autoUpdateIgnores** = DEFAULT_AUTO_UPDATE_IGNORES)

Sets whether this object both belongs to the specified layer, and collides with other objects in the same layer.
Combines of SetBelongsTo and SetCollidesWith, for convenience and to allow a simpler usage of the PhysicsMask system.
The regular rules still apply, however, so if this object is added to layer 1 but another object is not set to collide with layer 1 (even if the other object also belongs to layer 1), the objects won't collide.
In most cases there should never be a reason to set autoUpdateIgnores to false.

---

public void **SetColliders**(List<Collider> **colliders**)

Sets the list of colliders that are affected by this component.

---

public bool **GetBelongsTo**(int **layer**)

Returns whether or not this object belongs to the specified layer.

---

public bool **GetBelongsTo**(string **layerName**)

Returns whether or not this object belongs to the specified layer.

---

public ulong **GetBelongsToBitmask**()

Returns the belongsTo bitmask.

---

public bool **GetCollidesWith**(int **layer**)

Returns whether or not this object collides with the specified layer.

---

public bool **GetCollidesWith**(string **layerName**)

Returns whether or not this object collides with the specified layer.

---

public ulong **GetCollidesWithBitmask**()

Returns the collidesWith bitmask.

---

public bool **GetLayer**(int **layer**)

Returns whether or not this object both belongs to and collides with the specified layer.

---

public bool **GetLayer**(string **layerName**)

Returns whether or not this object both belongs to and collides with the specified layer.

---

public List<Collider> **GetColliders**()

Returns the list of colliders that are affected by this component.

---

public static ulong **BitmaskFromLayers**(params int[] **layers**)

Convenience function. Generates a bitmask from a supplied array of layers

---

public static RaycastHit[] **FilterRaycastHits**(RaycastHit[] **hits**, ulong **rayBelongsToBitmask**, ulong **rayCollidesWithBitmask**, bool **requireMask** = true)

Since there's no way to specify which colliders can and can't be hit by raycasts except regular Unity layers, this function can be used to filter out hits that don't match the supplied bitmasks.
Setting requireMask to false causes hits on objects without a PhysicsMask component to be returned

---

## PhysicsMaskController

---

public void **SetLayerName**(int **layer**, string **name**)

Sets a layer name by index.

---

public int **GetLayerByName**(string **name**)

Gets a layer index by name.

---

public List<string> **GetLayerNames**()

Returns the list of all layer names.

---

public static void **UpdateCollisionIgnores**()

Updates the physics system with the latest belongsTo and collidesWith bitmasks on all PhysicsMask components.
It's much faster to call UpdateCollisionIgnores(mask), so avoid this non-specific version if possible.

---

public static void **UpdateCollisionIgnores**(PhysicsMask **mask**)

Updates the physics system with the latest belongsTo and collidesWith bitmasks between the specified PhysicsMask and all others.