

[Main Page](#)

[Classes](#)

[Files](#)

---

# **Box2D API Documentation**

**2.0.1**

## Getting Started

For tutorials please see <http://www.box2d.org/manual.html>

For discussion please visit <http://www.box2d.org/forum>

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# Box2D Class List

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

<b>b2AABB</b>	<i>An axis aligned bounding box</i>
<b>b2Body</b>	<i>A rigid body</i>
<b>b2BodyDef</b>	<i>A body definition holds all the data needed to construct a rigid body</i>
<b>b2BoundaryListener</b>	<i>This is called when a body's shape passes outside of the world boundary</i>
<b>b2CircleDef</b>	<i>This structure is used to build circle shapes</i>
<b>b2CircleShape</b>	<i>A circle shape</i>
<b>b2Color</b>	<i>Color for debug drawing. Each value has the range [0,1]</i>
<b>b2Contact</b>	<i>The class manages contact between two shapes</i>
<b>b2ContactEdge</b>	<i>A contact edge is used to connect bodies and contacts together in a contact graph where each body is a node and each contact is an edge</i>
<b>b2ContactFilter</b>	<i>Implement this class to provide collision filtering</i>
<b>b2ContactID</b>	<i>Contact ids to facilitate warm starting</i>
<b>b2ContactID::Features</b>	<i>The features that intersect to form the contact point</i>
<b>b2ContactListener</b>	<i>Implement this class to get collision results</i>
<b>b2ContactPoint</b>	<i>This structure is used to report contact points</i>
<b>b2ContactResult</b>	<i>This structure is used to report contact point results</i>
<b>b2DebugDraw</b>	<i>Implement and register this class with a <b>b2World</b> to provide debug drawing of physics entities in your game</i>
<b>b2DestructionListener</b>	<i>Joints and shapes are destroyed when their associated body is destroyed</i>
<b>b2DistanceJoint</b>	<i>A distance joint constrains two points on two bodies to remain at a fixed distance from each other</i>
<b>b2DistanceJointDef</b>	<i>Distance joint definition</i>
<b>b2FilterData</b>	<i>This holds contact filtering data</i>
<b>b2GearJoint</b>	<i>A gear joint is used to connect two joints together</i>
<b>b2GearJointDef</b>	<i>Gear joint definition</i>
<b>b2Joint</b>	<i>The base joint class</i>
<b>b2JointDef</b>	<i>Joint definitions are used to construct joints</i>
<b>b2JointEdge</b>	<i>A joint edge is used to connect bodies and joints together in a joint graph where each body is a node and each joint is an edge</i>
<b>b2Manifold</b>	<i>A manifold for two touching convex shapes</i>

<b>b2ManifoldPoint</b>	<i>A manifold point is a contact point belonging to a contact manifold</i>
<b>b2MassData</b>	<i>This holds the mass data computed for a shape</i>
<b>b2Mat22</b>	<i>A 2-by-2 matrix. Stored in column-major order</i>
<b>b2MouseJoint</b>	<i>A mouse joint is used to make a point on a body track a specified world point</i>
<b>b2MouseJointDef</b>	<i>Mouse joint definition</i>
<b>b2OBB</b>	<i>An oriented bounding box</i>
<b>b2PolygonDef</b>	<i>Convex polygon</i>
<b>b2PolygonShape</b>	<i>A convex polygon</i>
<b>b2PrismaticJoint</b>	<i>A prismatic joint</i>
<b>b2PrismaticJointDef</b>	<i>Prismatic joint definition</i>
<b>b2PulleyJoint</b>	<i>The pulley joint is connected to two bodies and two fixed ground points</i>
<b>b2PulleyJointDef</b>	<i>Pulley joint definition</i>
<b>b2RevoluteJoint</b>	<i>A revolute joint constrains two bodies to share a common point while they are free to rotate about the point</i>
<b>b2RevoluteJointDef</b>	<i>Revolute joint definition</i>
<b>b2Segment</b>	<i>A line segment</i>
<b>b2Shape</b>	<i>A shape is used for collision detection</i>
<b>b2ShapeDef</b>	<i>A shape definition is used to construct a shape</i>
<b>b2Sweep</b>	<i>This describes the motion of a body/shape for TOI computation</i>
<b>b2Vec2</b>	<i>A 2D column vector</i>
<b>b2Version</b>	<i>Version numbering scheme</i>
<b>b2World</b>	<i>The world class manages all physics entities, dynamic simulation, and asynchronous queries</i>
<b>b2XForm</b>	<i>A transform contains translation and rotation</i>

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2AABB Struct Reference

An axis aligned bounding box. [More...](#)

[List of all members.](#)

## Public Member Functions

bool	<b>IsValid</b> () const
------	-------------------------

*Verify that the bounds are sorted.*

## Public Attributes

<b>b2Vec2</b>	<b>lowerBound</b>
	<i>the lower vertex</i>
<b>b2Vec2</b>	<b>upperBound</b>
	<i>the upper vertex</i>

---

## Detailed Description

An axis aligned bounding box.

---

The documentation for this struct was generated from the following file:

- [b2Collision.h](#)
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2Body Class Reference

A rigid body. [More...](#)

[List of all members.](#)

## Public Member Functions

<b>b2Shape</b> *	<b>CreateShape</b> ( <b>b2ShapeDef</b> *shapeDef) <i>Creates a shape and attach it to this body.</i>
void	<b>DestroyShape</b> ( <b>b2Shape</b> *shape) <i>Destroy a shape.</i>
void	<b>SetMass</b> (const <b>b2MassData</b> *massData) <i>Set the mass properties.</i>
void	<b>SetMassFromShapes</b> () <i>Compute the mass properties from the attached shapes.</i>
bool	<b>SetXForm</b> (const <b>b2Vec2</b> &position, float32 angle) <i>Set the position of the body's origin and rotation (radians).</i>
const <b>b2XForm</b> &	<b>GetXForm</b> () const <i>Get the body transform for the body's origin.</i>
const <b>b2Vec2</b> &	<b>GetPosition</b> () const <i>Get the world body origin position.</i>
float32	<b>GetAngle</b> () const <i>Get the angle in radians.</i>
const <b>b2Vec2</b> &	<b>GetWorldCenter</b> () const <i>Get the world position of the center of mass.</i>
const <b>b2Vec2</b> &	<b>GetLocalCenter</b> () const <i>Get the local position of the center of mass.</i>
void	<b>SetLinearVelocity</b> (const <b>b2Vec2</b> &v) <i>Set the linear velocity of the center of mass.</i>
<b>b2Vec2</b>	<b>GetLinearVelocity</b> () const <i>Get the linear velocity of the center of mass.</i>
void	<b>SetAngularVelocity</b> (float32 omega) <i>Set the angular velocity.</i>
float32	<b>GetAngularVelocity</b> () const <i>Get the angular velocity.</i>
void	<b>ApplyForce</b> (const <b>b2Vec2</b> &force, const <b>b2Vec2</b> &point) <i>Apply a force at a world point.</i>
void	<b>ApplyTorque</b> (float32 torque) <i>Apply a torque.</i>
void	<b>ApplyImpulse</b> (const <b>b2Vec2</b> &impulse, const <b>b2Vec2</b> &point) <i>Apply an impulse at a point.</i>
float32	<b>GetMass</b> () const <i>Get the total mass of the body.</i>

float32	<b>GetInertia</b> () const <i>Get the central rotational inertia of the body.</i>
b2Vec2	<b>GetWorldPoint</b> (const b2Vec2 &localPoint) const <i>Get the world coordinates of a point given the local coordinates.</i>
b2Vec2	<b>GetWorldVector</b> (const b2Vec2 &localVector) const <i>Get the world coordinates of a vector given the local coordinates.</i>
b2Vec2	<b>GetLocalPoint</b> (const b2Vec2 &worldPoint) const <i>Gets a local point relative to the body's origin given a world point.</i>
b2Vec2	<b>GetLocalVector</b> (const b2Vec2 &worldVector) const <i>Gets a local vector given a world vector.</i>
b2Vec2	<b>GetLinearVelocityFromWorldPoint</b> (const b2Vec2 &worldPoint) const <i>Get the world linear velocity of a world point attached to this body.</i>
b2Vec2	<b>GetLinearVelocityFromLocalPoint</b> (const b2Vec2 &localPoint) const <i>Get the world velocity of a local point.</i>
bool	<b>IsBullet</b> () const <i>Is this body treated like a bullet for continuous collision detection?</i>
void	<b>SetBullet</b> (bool flag) <i>Should this body be treated like a bullet for continuous collision detection?</i>
bool	<b>IsStatic</b> () const <i>Is this body static (immovable)?</i>
bool	<b>IsDynamic</b> () const <i>Is this body dynamic (movable)?</i>
bool	<b>IsFrozen</b> () const <i>Is this body frozen?</i>
bool	<b>IsSleeping</b> () const <i>Is this body sleeping (not simulating).</i>
void	<b>AllowSleeping</b> (bool flag) <i>You can disable sleeping on this body.</i>
void	<b>WakeUp</b> () <i>Wake up this body so it will begin simulating.</i>
void	<b>PutToSleep</b> () <i>Put this body to sleep so it will stop simulating.</i>
b2Shape *	<b>GetShapeList</b> () <i>Get the list of all shapes attached to this body.</i>
b2JointEdge *	<b>GetJointList</b> () <i>Get the list of all joints attached to this body.</i>
b2Body *	<b>GetNext</b> () <i>Get the next body in the world's body list.</i>
void *	<b>GetUserData</b> () <i>Get the user data pointer that was provided in the body definition.</i>
void	<b>SetUserData</b> (void *data)

*Set the user data. Use this to store your application specific data.*

**b2World** \* [GetWorld \(\)](#)

*Get the parent world of this body.*

## Friends

class	<a href="#"><b>b2World</b></a>
class	<a href="#"><b>b2DistanceJoint</b></a>
class	<a href="#"><b>b2GearJoint</b></a>
class	<a href="#"><b>b2MouseJoint</b></a>
class	<a href="#"><b>b2PrismaticJoint</b></a>
class	<a href="#"><b>b2PulleyJoint</b></a>
class	<a href="#"><b>b2RevoluteJoint</b></a>

---

## **Detailed Description**

A rigid body.

---

## Member Function Documentation

```
b2Shape * b2Body::CreateShape ( b2ShapeDef * shapeDef )
```

Creates a shape and attach it to this body.

**Parameters:**

*shapeDef* the shape definition.

**Warning:**

This function is locked during callbacks.

```
void b2Body::DestroyShape ( b2Shape * shape )
```

Destroy a shape.

This removes the shape from the broad-phase and therefore destroys any contacts associated with this shape. All shapes attached to a body are implicitly destroyed when the body is destroyed.

**Parameters:**

*shape* the shape to be removed.

**Warning:**

This function is locked during callbacks.

```
void b2Body::SetMass ( const b2MassData * massData )
```

Set the mass properties.

Note that this changes the center of mass position. If you are not sure how to compute mass properties, use SetMassFromShapes. The inertia tensor is assumed to be relative to the center of mass.

**Parameters:**

*massData* the mass properties.

```
void b2Body::SetMassFromShapes( )
```

Compute the mass properties from the attached shapes.

You typically call this after adding all the shapes. If you add or remove shapes later, you may want to call this again. Note that this changes the center of mass position.

```
bool b2Body::SetXForm ( const b2Vec2 & position,  
                        float32           angle  
                      )
```

Set the position of the body's origin and rotation (radians).

This breaks any contacts and wakes the other bodies.

**Parameters:**

*position* the new world position of the body's origin (not necessarily the center of mass).  
*angle* the new world rotation angle of the body in radians.

**Returns:**

false if the movement put a shape outside the world. In this case the body is automatically frozen.

```
const b2XForm & b2Body::GetXForm ( ) const [inline]
```

Get the body transform for the body's origin.

**Returns:**

the world transform of the body's origin.

```
const b2Vec2 & b2Body::GetPosition ( ) const [inline]
```

Get the world body origin position.

**Returns:**

the world position of the body's origin.

```
float32 b2Body::GetAngle( ) const [inline]
```

Get the angle in radians.

**Returns:**

the current world rotation angle in radians.

```
void b2Body::SetLinearVelocity( const b2Vec2 & v ) [inline]
```

Set the linear velocity of the center of mass.

**Parameters:**

*v* the new linear velocity of the center of mass.

```
b2Vec2 b2Body::GetLinearVelocity( ) const [inline]
```

Get the linear velocity of the center of mass.

**Returns:**

the linear velocity of the center of mass.

```
void b2Body::SetAngularVelocity( float32 omega ) [inline]
```

Set the angular velocity.

**Parameters:**

*omega* the new angular velocity in radians/second.

```
float32 b2Body::GetAngularVelocity( ) const [inline]
```

Get the angular velocity.

**Returns:**

the angular velocity in radians/second.

```
void b2Body::ApplyForce ( const b2Vec2 & force,  
                         const b2Vec2 & point  
                     ) [inline]
```

Apply a force at a world point.

If the force is not applied at the center of mass, it will generate a torque and affect the angular velocity. This wakes up the body.

**Parameters:**

*force* the world force vector, usually in Newtons (N).  
*point* the world position of the point of application.

```
void b2Body::ApplyTorque ( float32 torque ) [inline]
```

Apply a torque.

This affects the angular velocity without affecting the linear velocity of the center of mass. This wakes up the body.

**Parameters:**

*torque* about the z-axis (out of the screen), usually in N-m.

```
void b2Body::ApplyImpulse ( const b2Vec2 & impulse,  
                           const b2Vec2 & point  
                         ) [inline]
```

Apply an impulse at a point.

This immediately modifies the velocity. It also modifies the angular velocity if the point of application is not at the center of mass. This

wakes up the body.

**Parameters:**

*impulse* the world impulse vector, usually in N-seconds or kg-m/s.  
*point* the world position of the point of application.

**float32 b2Body::GetMass ( ) const [inline]**

Get the total mass of the body.

**Returns:**

the mass, usually in kilograms (kg).

**float32 b2Body::GetInertia ( ) const [inline]**

Get the central rotational inertia of the body.

**Returns:**

the rotational inertia, usually in kg-m<sup>2</sup>.

**b2Vec2 b2Body::GetWorldPoint ( const b2Vec2 & *localPoint* ) const [inline]**

Get the world coordinates of a point given the local coordinates.

**Parameters:**

*localPoint* a point on the body measured relative the the body's origin.

**Returns:**

the same point expressed in world coordinates.

**b2Vec2 b2Body::GetWorldVector ( const b2Vec2 & *localVector* ) const [inline]**

Get the world coordinates of a vector given the local coordinates.

**Parameters:**

*localVector* a vector fixed in the body.

**Returns:**

the same vector expressed in world coordinates.

**b2Vec2** **b2Body::GetLocalPoint** ( const **b2Vec2** & *worldPoint* ) const [inline]

Gets a local point relative to the body's origin given a world point.

**Parameters:**

a point in world coordinates.

**Returns:**

the corresponding local point relative to the body's origin.

**b2Vec2** **b2Body::GetLocalVector** ( const **b2Vec2** & *worldVector* ) const [inline]

Gets a local vector given a world vector.

**Parameters:**

a vector in world coordinates.

**Returns:**

the corresponding local vector.

**b2Vec2** **b2Body::GetLinearVelocityFromWorldPoint** ( const **b2Vec2** & *worldPoint* ) const

Get the world linear velocity of a world point attached to this body.

**Parameters:**

a point in world coordinates.

**Returns:**

the world velocity of a point.

```
b2Vec2 b2Body::GetLinearVelocityFromLocalPoint ( const b2Vec2 & localPoint ) const [
```

Get the world velocity of a local point.

**Parameters:**

a point in local coordinates.

**Returns:**

the world velocity of a point.

```
void b2Body::PutToSleep ( ) [inline]
```

Put this body to sleep so it will stop simulating.

This also sets the velocity to zero.

---

The documentation for this class was generated from the following files:

- b2Body.h
- b2Body.cpp

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## **b2BodyDef Struct Reference**

A body definition holds all the data needed to construct a rigid body.

[More...](#)

[List of all members.](#)

## Public Member Functions

---

### **b2BodyDef ()**

*This constructor sets the body definition default values.*

## Public Attributes

<b>b2MassData</b>	<b>massData</b>
	<i>You can use this to initialized the mass properties of the body.</i>
void *	<b>userData</b>
	<i>Use this to store application specific body data.</i>
<b>b2Vec2</b>	<b>position</b>
	<i>The world position of the body.</i>
float32	<b>angle</b>
	<i>The world angle of the body in radians.</i>
float32	<b>linearDamping</b>
	<i>Linear damping is use to reduce the linear velocity.</i>
float32	<b>angularDamping</b>
	<i>Angular damping is use to reduce the angular velocity.</i>
bool	<b>allowSleep</b>
	<i>Set this flag to false if this body should never fall asleep.</i>
bool	<b>isSleeping</b>
	<i>Is this body initially sleeping?</i>
bool	<b>fixedRotation</b>
	<i>Should this body be prevented from rotating? Useful for characters.</i>
bool	<b>isBullet</b>
	<i>Is this a fast moving body that should be prevented from tunneling through other moving bodies? Note that all bodies are prevented from tunneling through static bodies.</i>

## **Detailed Description**

A body definition holds all the data needed to construct a rigid body.

You can safely re-use body definitions.

---

## Member Data Documentation

### **b2MassData b2BodyDef::massData**

You can use this to initialized the mass properties of the body.

If you prefer, you can set the mass properties after the shapes have been added using **b2Body::SetMassFromShapes**.

### **b2Vec2 b2BodyDef::position**

The world position of the body.

Avoid creating bodies at the origin since this can lead to many overlapping shapes.

### **float32 b2BodyDef::linearDamping**

Linear damping is use to reduce the linear velocity.

The damping parameter can be larger than 1.0f but the damping effect becomes sensitive to the time step when the damping parameter is large.

### **float32 b2BodyDef::angularDamping**

Angular damping is use to reduce the angular velocity.

The damping parameter can be larger than 1.0f but the damping effect becomes sensitive to the time step when the damping parameter is large.

### **bool b2BodyDef::allowSleep**

Set this flag to false if this body should never fall asleep.

Note that this increases CPU usage.

---

**bool b2BodyDef::isBullet**

Is this a fast moving body that should be prevented from tunneling through other moving bodies? Note that all bodies are prevented from tunneling through static bodies.

**Warning:**

You should use this flag sparingly since it increases processing time.

---

The documentation for this struct was generated from the following file:

- b2Body.h

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## **b2BoundaryListener Class Reference**

This is called when a body's shape passes outside of the world boundary.

[More...](#)

[List of all members.](#)

## Public Member Functions

virtual void **Violation** (**b2Body** \*body)=0

*This is called for each body that leaves the world boundary.*

---

## **Detailed Description**

This is called when a body's shape passes outside of the world boundary.

---

## Member Function Documentation

```
virtual void b2BoundaryListener::Violation ( b2Body * body ) [pure virtual]
```

This is called for each body that leaves the world boundary.

**Warning:**

you can't modify the world inside this callback.

---

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

- b2WorldCallbacks.h
- 

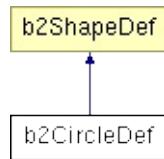
*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2CircleDef Struct Reference

This structure is used to build circle shapes. [More...](#)

Inheritance diagram for b2CircleDef:



[List of all members.](#)

---

## Detailed Description

This structure is used to build circle shapes.

---

The documentation for this struct was generated from the following file:

- b2CircleShape.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

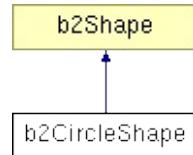
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2CircleShape Class Reference

A circle shape. [More...](#)

Inheritance diagram for b2CircleShape:



[List of all members.](#)

## Public Member Functions

bool	<b>TestPoint</b> (const <b>b2XForm</b> &transform, const <b>b2Vec2</b> &p) const  <b>See also:</b> <i><a href="#">b2Shape::TestPoint</a></i>
bool	<b>TestSegment</b> (const <b>b2XForm</b> &transform, float32 *lambda, <b>b2Vec2</b> *normal, const <b>b2Segment</b> &segment, float32 maxLambda) const  <b>See also:</b> <i><a href="#">b2Shape::TestSegment</a></i>
void	<b>ComputeAABB</b> ( <b>b2AABB</b> *aabb, const <b>b2XForm</b> &transform) const  <b>See also:</b> <i><a href="#">b2Shape::ComputeAABB</a></i>
void	<b>ComputeSweptAABB</b> ( <b>b2AABB</b> *aabb, const <b>b2XForm</b> &transform1, const <b>b2XForm</b> &transform2) const  <b>See also:</b> <i><a href="#">b2Shape::ComputeSweptAABB</a></i>
void	<b>ComputeMass</b> ( <b>b2MassData</b> *massData) const  <b>See also:</b> <i><a href="#">b2Shape::ComputeMass</a></i>
const <b>b2Vec2</b> &	<b>GetLocalPosition</b> () const <i>Get the local position of this circle in its parent body.</i>
float32	<b>GetRadius</b> () const <i>Get the radius of this circle.</i>

## Friends

class [b2Shape](#)

---

## Detailed Description

A circle shape.

---

The documentation for this class was generated from the following files:

- b2CircleShape.h
  - b2CircleShape.cpp
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Color Struct Reference

Color for debug drawing. Each value has the range [0,1]. [More...](#)

[List of all members.](#)

---

## Detailed Description

Color for debug drawing. Each value has the range [0,1].

---

The documentation for this struct was generated from the following file:

- b2WorldCallbacks.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Contact Class Reference

The class manages contact between two shapes. [More...](#)

Inherited by b2CircleContact, b2NullContact, b2PolyAndCircleContact, and b2PolygonContact.

[List of all members.](#)

## Public Member Functions

virtual <b>b2Manifold</b> *	<b>GetManifolds</b> ()=0 <i>Get the manifold array.</i>
int32	<b>GetManifoldCount</b> () const <i>Get the number of manifolds.</i>
bool	<b>IsSolid</b> () const <i>Is this contact solid?</i>
<b>b2Contact</b> *	<b>GetNext</b> () <i>Get the next contact in the world's contact list.</i>
<b>b2Shape</b> *	<b>GetShape1</b> () <i>Get the first shape in this contact.</i>
<b>b2Shape</b> *	<b>GetShape2</b> () <i>Get the second shape in this contact.</i>

---

## **Detailed Description**

The class manages contact between two shapes.

A contact exists for each overlapping AABB in the broad-phase (except if filtered). Therefore a contact object may exist that has no contact points.

---

## Member Function Documentation

```
int32 b2Contact::GetManifoldCount( ) const [inline]
```

Get the number of manifolds.

This is 0 or 1 between convex shapes. This may be greater than 1 for convex-vs-concave shapes. Each manifold holds up to two contact points with a shared contact normal.

```
bool b2Contact::IsSolid( ) const [inline]
```

Is this contact solid?

**Returns:**

true if this contact should generate a response.

---

The documentation for this class was generated from the following files:

- b2Contact.h
- b2Contact.cpp

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ContactEdge Struct Reference

A contact edge is used to connect bodies and contacts together in a contact graph where each body is a node and each contact is an edge.

[More...](#)

[List of all members.](#)

## Public Attributes

<b>b2Body</b> *	<b>other</b> <i>provides quick access to the other body attached.</i>
<b>b2Contact</b> *	<b>contact</b> <i>the contact</i>
<b>b2ContactEdge</b> *	<b>prev</b> <i>the previous contact edge in the body's contact list</i>
<b>b2ContactEdge</b> *	<b>next</b> <i>the next contact edge in the body's contact list</i>

---

## Detailed Description

A contact edge is used to connect bodies and contacts together in a contact graph where each body is a node and each contact is an edge.

A contact edge belongs to a doubly linked list maintained in each attached body. Each contact has two contact nodes, one for each attached body.

---

The documentation for this struct was generated from the following file:

- b2Contact.h

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2ContactFilter Class Reference

Implement this class to provide collision filtering. [More...](#)

[List of all members.](#)

## Public Member Functions

virtual bool	<b>ShouldCollide</b> ( <b>b2Shape</b> *shape1, <b>b2Shape</b> *shape2)
--------------	--

*Return true if contact calculations should be performed between these two shapes.*

---

## **Detailed Description**

Implement this class to provide collision filtering.

In other words, you can implement this class if you want finer control over contact creation.

---

## Member Function Documentation

```
bool b2ContactFilter::ShouldCollide ( b2Shape * shape1,  
                                     b2Shape * shape2  
                                 ) [virtual]
```

Return true if contact calculations should be performed between these two shapes.

**Warning:**

for performance reasons this is only called when the AABBs begin to overlap.

---

The documentation for this class was generated from the following files:

- b2WorldCallbacks.h
- b2WorldCallbacks.cpp

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2ContactID Union Reference

Contact ids to facilitate warm starting. [More...](#)

[List of all members.](#)

## Public Attributes

uint32 **key**

*Used to quickly compare contact ids.*

## Classes

struct	<b>Features</b>
	<i>The features that intersect to form the contact point.</i> <a href="#">More...</a>

---

## Detailed Description

Contact ids to facilitate warm starting.

---

The documentation for this union was generated from the following file:

- [b2Collision.h](#)
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ContactID::Features

## b2ContactID::Features Struct Reference

The features that intersect to form the contact point. [More...](#)

[List of all members.](#)

## Public Attributes

uint8	<b>referenceEdge</b>
	<i>The edge that defines the outward contact normal.</i>
uint8	<b>incidentEdge</b>
	<i>The edge most anti-parallel to the reference edge.</i>
uint8	<b>incidentVertex</b>
	<i>The vertex (0 or 1) on the incident edge that was clipped.</i>
uint8	<b>flip</b>
	<i>A value of 1 indicates that the reference edge is on shape2.</i>

---

## Detailed Description

The features that intersect to form the contact point.

---

The documentation for this struct was generated from the following file:

- [b2Collision.h](#)

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2ContactListener Class Reference

Implement this class to get collision results. [More...](#)

[List of all members.](#)

## Public Member Functions

virtual void	<b>Add</b> (const <b>b2ContactPoint</b> *point) <i>Called when a contact point is added.</i>
virtual void	<b>Persist</b> (const <b>b2ContactPoint</b> *point) <i>Called when a contact point persists.</i>
virtual void	<b>Remove</b> (const <b>b2ContactPoint</b> *point) <i>Called when a contact point is removed.</i>
virtual void	<b>Result</b> (const <b>b2ContactResult</b> *point) <i>Called after a contact point is solved.</i>

---

## Detailed Description

Implement this class to get collision results.

You can use these results for things like sounds and game logic. You can also get contact results by traversing the contact lists after the time step. However, you might miss some contacts because continuous physics leads to sub-stepping. Additionally you may receive multiple callbacks for the same contact in a single time step. You should strive to make your callbacks efficient because there may be many callbacks per time step.

### **Warning:**

The contact separation is the last computed value.

You cannot create/destroy Box2D entities inside these callbacks.

---

## Member Function Documentation

```
virtual void b2ContactListener::Add ( const b2ContactPoint * point ) [inline, virtual]
```

Called when a contact point is added.

This includes the geometry and the forces.

```
virtual void b2ContactListener::Persist ( const b2ContactPoint * point ) [inline, virtual]
```

Called when a contact point persists.

This includes the geometry and the forces.

```
virtual void b2ContactListener::Remove ( const b2ContactPoint * point ) [inline, virtual]
```

Called when a contact point is removed.

This includes the last computed geometry and forces.

---

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

- b2WorldCallbacks.h

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ContactPoint Struct Reference

This structure is used to report contact points. [More...](#)

[List of all members.](#)

## Public Attributes

<b>b2Shape</b> *	<b>shape1</b> <i>the first shape</i>
<b>b2Shape</b> *	<b>shape2</b> <i>the second shape</i>
<b>b2Vec2</b>	<b>position</b> <i>position in world coordinates</i>
<b>b2Vec2</b>	<b>velocity</b> <i>velocity of point on body2 relative to point on body1 (pre-solver)</i>
<b>b2Vec2</b>	<b>normal</b> <i>points from shape1 to shape2</i>
float32	<b>separation</b> <i>the separation is negative when shapes are touching</i>
float32	<b>friction</b> <i>the combined friction coefficient</i>
float32	<b>restitution</b> <i>the combined restitution coefficient</i>
<b>b2ContactID</b>	<b>id</b> <i>the contact id identifies the features in contact</i>

---

## Detailed Description

This structure is used to report contact points.

---

The documentation for this struct was generated from the following file:

- b2Contact.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2ContactResult Struct Reference

This structure is used to report contact point results. [More...](#)

[List of all members.](#)

## Public Attributes

<b>b2Shape</b> *	<b>shape1</b> <i>the first shape</i>
<b>b2Shape</b> *	<b>shape2</b> <i>the second shape</i>
<b>b2Vec2</b>	<b>position</b> <i>position in world coordinates</i>
<b>b2Vec2</b>	<b>normal</b> <i>points from shape1 to shape2</i>
float32	<b>normalImpulse</b> <i>the normal impulse applied to body2</i>
float32	<b>tangentImpulse</b> <i>the tangent impulse applied to body2</i>
<b>b2ContactID</b>	<b>id</b> <i>the contact id identifies the features in contact</i>

---

## Detailed Description

This structure is used to report contact point results.

---

The documentation for this struct was generated from the following file:

- b2Contact.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2DebugDraw Class Reference

Implement and register this class with a [b2World](#) to provide debug drawing of physics entities in your game. [More...](#)

[List of all members.](#)

## Public Types

enum	{ <b>e_shapeBit</b> = 0x0001, <b>e_jointBit</b> = 0x0002, <b>e_coreShapeBit</b> = 0x0004, <b>e_aabbBit</b> = 0x0008, <b>e_obbBit</b> = 0x0010, <b>e_pairBit</b> = 0x0020, <b>e_centerOfMassBit</b> = 0x0040 }
------	---

## Public Member Functions

void	<b>SetFlags</b> (uint32 flags) <i>Set the drawing flags.</i>
uint32	<b>GetFlags</b> () const <i>Get the drawing flags.</i>
void	<b>AppendFlags</b> (uint32 flags) <i>Append flags to the current flags.</i>
void	<b>ClearFlags</b> (uint32 flags) <i>Clear flags from the current flags.</i>
virtual void	<b>DrawPolygon</b> (const <b>b2Vec2</b> *vertices, int32 vertexCount, const <b>b2Color</b> &color)=0 <i>Draw a closed polygon provided in CCW order.</i>
virtual void	<b>DrawSolidPolygon</b> (const <b>b2Vec2</b> *vertices, int32 vertexCount, const <b>b2Color</b> &color)=0 <i>Draw a solid closed polygon provided in CCW order.</i>
virtual void	<b>DrawCircle</b> (const <b>b2Vec2</b> &center, float32 radius, const <b>b2Color</b> &color)=0 <i>Draw a circle.</i>
virtual void	<b>DrawSolidCircle</b> (const <b>b2Vec2</b> &center, float32 radius, const <b>b2Vec2</b> &axis, const <b>b2Color</b> &color)=0 <i>Draw a solid circle.</i>
virtual void	<b>DrawSegment</b> (const <b>b2Vec2</b> &p1, const <b>b2Vec2</b> &p2, const <b>b2Color</b> &color)=0 <i>Draw a line segment.</i>
virtual void	<b>DrawXForm</b> (const <b>b2XForm</b> &xf)=0 <i>Draw a transform.</i>

---

## Detailed Description

Implement and register this class with a **b2World** to provide debug drawing of physics entities in your game.

---

# Member Enumeration Documentation

## anonymous enum

### Enumerator:

<code>e_shapeBit</code>	draw shapes
<code>e_jointBit</code>	draw joint connections
<code>e_coreShapeBit</code>	draw core (TOI) shapes
<code>e_aabbBit</code>	draw axis aligned bounding boxes
<code>e_obbBit</code>	draw oriented bounding boxes
<code>e_pairBit</code>	draw broad-phase pairs
<code>e_centerOfMassBit</code>	draw center of mass frame

---

## Member Function Documentation

```
virtual void b2DebugDraw::DrawXForm ( const b2XForm & xf ) [pure virtual]
```

Draw a transform.

Choose your own length scale.

**Parameters:**

*xf* a transform.

---

The documentation for this class was generated from the following files:

- b2WorldCallbacks.h
- b2WorldCallbacks.cpp

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2DestructionListener Class Reference

Joints and shapes are destroyed when their associated body is destroyed. [More...](#)

[List of all members.](#)

## Public Member Functions

virtual void	<b>SayGoodbye (b2Joint *joint)=0</b> <i>Called when any joint is about to be destroyed due to the destruction of one of its attached bodies.</i>
virtual void	<b>SayGoodbye (b2Shape *shape)=0</b> <i>Called when any shape is about to be destroyed due to the destruction of its parent body.</i>

---

## **Detailed Description**

Joints and shapes are destroyed when their associated body is destroyed.

Implement this listener so that you may nullify references to these joints and shapes.

---

## Member Function Documentation

```
virtual void b2DestructionListener::SayGoodbye ( b2Joint * joint ) [pure virtual]
```

Called when any joint is about to be destroyed due to the destruction of one of its attached bodies.

```
virtual void b2DestructionListener::SayGoodbye ( b2Shape * shape ) [pure virtual]
```

Called when any shape is about to be destroyed due to the destruction of its parent body.

---

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

- b2WorldCallbacks.h

[Main Page](#) [Classes](#) [Files](#)

---

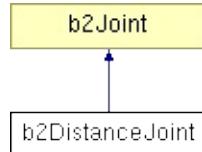
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2DistanceJoint Class Reference

A distance joint constrains two points on two bodies to remain at a fixed distance from each other. [More...](#)

Inheritance diagram for b2DistanceJoint:



[List of all members.](#)

## Public Member Functions

b2Vec2	<b>GetAnchor1</b> () const <i>Get the anchor point on body1 in world coordinates.</i>
b2Vec2	<b>GetAnchor2</b> () const <i>Get the anchor point on body2 in world coordinates.</i>
b2Vec2	<b>GetReactionForce</b> () const <i>Get the reaction force on body2 at the joint anchor.</i>
float32	<b>GetReactionTorque</b> () const <i>Get the reaction torque on body2.</i>

---

## Detailed Description

A distance joint constrains two points on two bodies to remain at a fixed distance from each other.

You can view this as a massless, rigid rod.

---

The documentation for this class was generated from the following files:

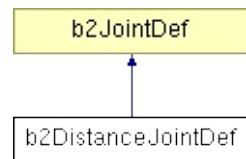
- b2DistanceJoint.h
- b2DistanceJoint.cpp

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2DistanceJointDef Struct Reference

Distance joint definition. [More...](#)

Inheritance diagram for b2DistanceJointDef:



[List of all members.](#)

## Public Member Functions

void	<b>Initialize</b> ( <b>b2Body</b> * <b>body1</b> , <b>b2Body</b> * <b>body2</b> , const <b>b2Vec2</b> & <b>anchor1</b> , const <b>b2Vec2</b> & <b>anchor2</b> ) <i>Initialize the bodies, anchors, and length using the world anchors.</i>
------	---

## Public Attributes

b2Vec2	<b>localAnchor1</b>
	<i>The local anchor point relative to body1's origin.</i>
b2Vec2	<b>localAnchor2</b>
	<i>The local anchor point relative to body2's origin.</i>
float32	<b>length</b>
	<i>The equilibrium length between the anchor points.</i>
float32	<b>frequencyHz</b>
	<i>The response speed.</i>
float32	<b>dampingRatio</b>
	<i>The damping ratio. 0 = no damping, 1 = critical damping.</i>

---

## Detailed Description

Distance joint definition.

This requires defining an anchor point on both bodies and the non-zero length of the distance joint. The definition uses local anchor points so that the initial configuration can violate the constraint slightly. This helps when saving and loading a game.

**Warning:**

Do not use a zero or short length.

---

## Member Function Documentation

```
void b2DistanceJointDef::Initialize ( b2Body *      body1,
                                      b2Body *      body2,
                                      const b2Vec2 & anchor1,
                                      const b2Vec2 & anchor2
                                    )
```

Initialize the bodies, anchors, and length using the world anchors.

---

The documentation for this struct was generated from the following files:

- b2DistanceJoint.h
  - b2DistanceJoint.cpp
-

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2FilterData Struct Reference

This holds contact filtering data. [More...](#)

[List of all members.](#)

## Public Attributes

uint16	<b>categoryBits</b>
	<i>The collision category bits. Normally you would just set one bit.</i>
uint16	<b>maskBits</b>
	<i>The collision mask bits.</i>
int16	<b>groupIndex</b>
	<i>Collision groups allow a certain group of objects to never collide (negative) or always collide (positive).</i>

---

## **Detailed Description**

This holds contact filtering data.

---

## Member Data Documentation

**uint16 b2FilterData::maskBits**

The collision mask bits.

This states the categories that this shape would accept for collision.

**int16 b2FilterData::groupIndex**

Collision groups allow a certain group of objects to never collide (negative) or always collide (positive).

Zero means no collision group. Non-zero group filtering always wins against the mask bits.

---

The documentation for this struct was generated from the following file:

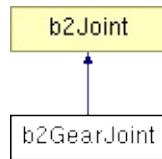
- b2Shape.h
-

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2GearJoint Class Reference

A gear joint is used to connect two joints together. [More...](#)

Inheritance diagram for b2GearJoint:



[List of all members.](#)

## Public Member Functions

b2Vec2	<b>GetAnchor1</b> () const <i>Get the anchor point on body1 in world coordinates.</i>
b2Vec2	<b>GetAnchor2</b> () const <i>Get the anchor point on body2 in world coordinates.</i>
b2Vec2	<b>GetReactionForce</b> () const <i>Get the reaction force on body2 at the joint anchor.</i>
float32	<b>GetReactionTorque</b> () const <i>Get the reaction torque on body2.</i>
float32	<b>GetRatio</b> () const <i>Get the gear ratio.</i>

---

## Detailed Description

A gear joint is used to connect two joints together.

Either joint can be a revolute or prismatic joint. You specify a gear ratio to bind the motions together: coordinate1 + ratio \* coordinate2 = constant  
The ratio can be negative or positive. If one joint is a revolute joint and the other joint is a prismatic joint, then the ratio will have units of length or units of 1/length.

### Warning:

The revolute and prismatic joints must be attached to fixed bodies (which must be body1 on those joints).

---

The documentation for this class was generated from the following files:

- b2GearJoint.h
- b2GearJoint.cpp

[Main Page](#) [Classes](#) [Files](#)

---

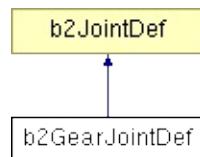
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2GearJointDef Struct Reference

Gear joint definition. [More...](#)

Inheritance diagram for b2GearJointDef:



[List of all members.](#)

## Public Attributes

<b>b2Joint *</b>	<b>joint1</b> <i>The first revolute/prismatic joint attached to the gear joint.</i>
<b>b2Joint *</b>	<b>joint2</b> <i>The second revolute/prismatic joint attached to the gear joint.</i>
float32	<b>ratio</b> <i>The gear ratio.</i>

---

## **Detailed Description**

Gear joint definition.

This definition requires two existing revolute or prismatic joints (any combination will work). The provided joints must attach a dynamic body to a static body.

---

## Member Data Documentation

float32 **b2GearJointDef::ratio**

The gear ratio.

**See also:**

[b2GearJoint](#) for explanation.

---

The documentation for this struct was generated from the following file:

- b2GearJoint.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

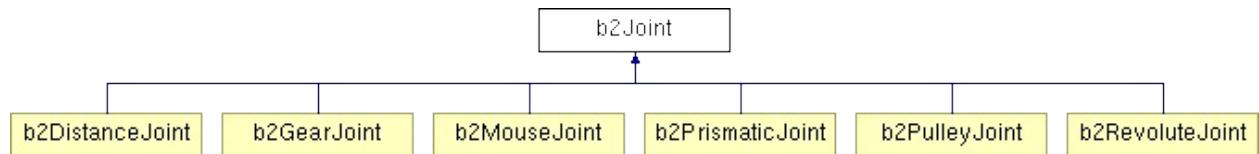
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2Joint Class Reference

The base joint class. [More...](#)

Inheritance diagram for b2Joint:



[List of all members.](#)

## Public Member Functions

b2JointType	<b>GetType () const</b> <i>Get the type of the concrete joint.</i>
b2Body *	<b>GetBody1 ()</b> <i>Get the first body attached to this joint.</i>
b2Body *	<b>GetBody2 ()</b> <i>Get the second body attached to this joint.</i>
virtual b2Vec2	<b>GetAnchor1 () const =0</b> <i>Get the anchor point on body1 in world coordinates.</i>
virtual b2Vec2	<b>GetAnchor2 () const =0</b> <i>Get the anchor point on body2 in world coordinates.</i>
virtual b2Vec2	<b>GetReactionForce () const =0</b> <i>Get the reaction force on body2 at the joint anchor.</i>
virtual float32	<b>GetReactionTorque () const =0</b> <i>Get the reaction torque on body2.</i>
b2Joint *	<b>GetNext ()</b> <i>Get the next joint the world joint list.</i>
void *	<b>GetUserData ()</b> <i>Get the user data pointer.</i>
void	<b>SetUserData (void *data)</b> <i>Set the user data pointer.</i>

## Friends

class [b2World](#)

class [b2Body](#)

---

## Detailed Description

The base joint class.

Joints are used to constraint two bodies together in various fashions. Some joints also feature limits and motors.

---

The documentation for this class was generated from the following files:

- b2Joint.h
  - b2Joint.cpp
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

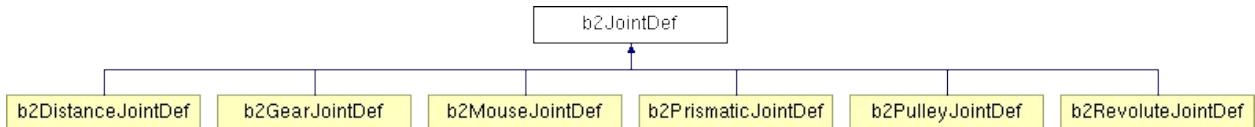
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2JointDef Struct Reference

Joint definitions are used to construct joints. [More...](#)

Inheritance diagram for b2JointDef:



[List of all members.](#)

## Public Attributes

b2JointType	<b>type</b> <i>The joint type is set automatically for concrete joint types.</i>
void *	<b>userData</b> <i>Use this to attach application specific data to your joints.</i>
b2Body *	<b>body1</b> <i>The first attached body.</i>
b2Body *	<b>body2</b> <i>The second attached body.</i>
bool	<b>collideConnected</b> <i>Set this flag to true if the attached bodies should collide.</i>

---

## Detailed Description

Joint definitions are used to construct joints.

---

The documentation for this struct was generated from the following file:

- b2Joint.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2JointEdge Struct Reference

A joint edge is used to connect bodies and joints together in a joint graph where each body is a node and each joint is an edge. [More...](#)

[List of all members.](#)

## Public Attributes

<b>b2Body</b> *	<b>other</b> <i>provides quick access to the other body attached.</i>
<b>b2Joint</b> *	<b>joint</b> <i>the joint</i>
<b>b2JointEdge</b> *	<b>prev</b> <i>the previous joint edge in the body's joint list</i>
<b>b2JointEdge</b> *	<b>next</b> <i>the next joint edge in the body's joint list</i>

---

## Detailed Description

A joint edge is used to connect bodies and joints together in a joint graph where each body is a node and each joint is an edge.

A joint edge belongs to a doubly linked list maintained in each attached body. Each joint has two joint nodes, one for each attached body.

---

The documentation for this struct was generated from the following file:

- b2Joint.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2Manifold Struct Reference

A manifold for two touching convex shapes. [More...](#)

[List of all members.](#)

## Public Attributes

<b>b2ManifoldPoint</b>	<b>points</b> [b2_maxManifoldPoints] <i>the points of contact</i>
<b>b2Vec2</b>	<b>normal</b> <i>the shared unit normal vector</i>
int32	<b>pointCount</b> <i>the number of manifold points</i>

---

## Detailed Description

A manifold for two touching convex shapes.

---

The documentation for this struct was generated from the following file:

- [b2Collision.h](#)
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ManifoldPoint Struct Reference

A manifold point is a contact point belonging to a contact manifold.

[More...](#)

[List of all members.](#)

## Public Attributes

b2Vec2	<b>localPoint1</b> <i>local position of the contact point in body1</i>
b2Vec2	<b>localPoint2</b> <i>local position of the contact point in body2</i>
float32	<b>separation</b> <i>the separation of the shapes along the normal vector</i>
float32	<b>normalImpulse</b> <i>the non-penetration impulse</i>
float32	<b>tangentImpulse</b> <i>the friction impulse</i>
b2ContactID	<b>id</b> <i>uniquely identifies a contact point between two shapes</i>

---

## Detailed Description

A manifold point is a contact point belonging to a contact manifold.

It holds details related to the geometry and dynamics of the contact points. The point is stored in local coordinates because CCD requires sub-stepping in which the separation is stale.

---

The documentation for this struct was generated from the following file:

- [b2Collision.h](#)
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2MassData Struct Reference

This holds the mass data computed for a shape. [More...](#)

[List of all members.](#)

## Public Attributes

float32	<b>mass</b>
	<i>The mass of the shape, usually in kilograms.</i>
b2Vec2	<b>center</b>
	<i>The position of the shape's centroid relative to the shape's origin.</i>
float32	<b>I</b>
	<i>The rotational inertia of the shape.</i>

---

## Detailed Description

This holds the mass data computed for a shape.

---

The documentation for this struct was generated from the following file:

- b2Shape.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Mat22 Struct Reference

A 2-by-2 matrix. Stored in column-major order. [More...](#)

[List of all members.](#)

## Public Member Functions

	<b>b2Mat22 ()</b> <i>The default constructor does nothing (for performance).</i>
	<b>b2Mat22 (const b2Vec2 &amp;c1, const b2Vec2 &amp;c2)</b> <i>Construct this matrix using columns.</i>
	<b>b2Mat22 (float32 a11, float32 a12, float32 a21, float32 a22)</b> <i>Construct this matrix using scalars.</i>
	<b>b2Mat22 (float32 angle)</b> <i>Construct this matrix using an angle.</i>
void	<b>Set (const b2Vec2 &amp;c1, const b2Vec2 &amp;c2)</b> <i>Initialize this matrix using columns.</i>
void	<b>Set (float32 angle)</b> <i>Initialize this matrix using an angle.</i>
void	<b>SetIdentity ()</b> <i>Set this to the identity matrix.</i>
void	<b>SetZero ()</b> <i>Set this matrix to all zeros.</i>
float32	<b>GetAngle () const</b> <i>Extract the angle from this matrix (assumed to be a rotation matrix).</i>
b2Vec2	<b>Solve (const b2Vec2 &amp;b) const</b> <i>Solve <math>A * x = b</math>, where <math>b</math> is a column vector.</i>

---

## **Detailed Description**

A 2-by-2 matrix. Stored in column-major order.

---

## Constructor & Destructor Documentation

```
b2Mat22::b2Mat22 ( float32 angle ) [inline, explicit]
```

Construct this matrix using an angle.

This matrix becomes an orthonormal rotation matrix.

---

## Member Function Documentation

```
void b2Mat22::Set ( float32 angle ) [inline]
```

Initialize this matrix using an angle.

This matrix becomes an orthonormal rotation matrix.

```
float32 b2Mat22::GetAngle ( ) const [inline]
```

Extract the angle from this matrix (assumed to be a rotation matrix).

```
b2Vec2 b2Mat22::Solve ( const b2Vec2 & b ) const [inline]
```

Solve  $A * x = b$ , where  $b$  is a column vector.

This is more efficient than computing the inverse in one-shot cases.

---

The documentation for this struct was generated from the following file:

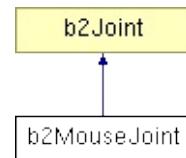
- b2Math.h

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2MouseJoint Class Reference

A mouse joint is used to make a point on a body track a specified world point. [More...](#)

Inheritance diagram for b2MouseJoint:



[List of all members.](#)

## Public Member Functions

<b>b2Vec2</b>	<b>GetAnchor1</b> () const <i>Implements b2Joint.</i>
<b>b2Vec2</b>	<b>GetAnchor2</b> () const <i>Implements b2Joint.</i>
<b>b2Vec2</b>	<b>GetReactionForce</b> () const <i>Implements b2Joint.</i>
float32	<b>GetReactionTorque</b> () const <i>Implements b2Joint.</i>
void	<b>SetTarget</b> (const <b>b2Vec2</b> &target) <i>Use this to update the target point.</i>

---

## Detailed Description

A mouse joint is used to make a point on a body track a specified world point.

This a soft constraint with a maximum force. This allows the constraint to stretch and without applying huge forces.

---

The documentation for this class was generated from the following files:

- b2MouseJoint.h
  - b2MouseJoint.cpp
-

[Main Page](#) [Classes](#) [Files](#)

---

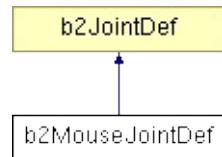
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2MouseJointDef Struct Reference

Mouse joint definition. [More...](#)

Inheritance diagram for b2MouseJointDef:



[List of all members.](#)

## Public Attributes

b2Vec2	<b>target</b>
	<i>The initial world target point.</i>
float32	<b>maxForce</b>
	<i>The maximum constraint force that can be exerted to move the candidate body.</i>
float32	<b>frequencyHz</b>
	<i>The response speed.</i>
float32	<b>dampingRatio</b>
	<i>The damping ratio. 0 = no damping, 1 = critical damping.</i>
float32	<b>timeStep</b>
	<i>The time step used in the simulation.</i>

---

## **Detailed Description**

Mouse joint definition.

This requires a world target point, tuning parameters, and the time step.

---

## Member Data Documentation

### **b2Vec2 b2MouseJointDef::target**

The initial world target point.

This is assumed to coincide with the body anchor initially.

### **float32 b2MouseJointDef::maxForce**

The maximum constraint force that can be exerted to move the candidate body.

Usually you will express as some multiple of the weight (multiplier \* mass \* gravity).

---

The documentation for this struct was generated from the following file:

- b2MouseJoint.h
-

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2OBB Struct Reference

An oriented bounding box. [More...](#)

[List of all members.](#)

## Public Attributes

<b>b2Mat22</b>	<b>R</b> <i>the rotation matrix</i>
<b>b2Vec2</b>	<b>center</b> <i>the local centroid</i>
<b>b2Vec2</b>	<b>extents</b> <i>the half-widths</i>

---

## Detailed Description

An oriented bounding box.

---

The documentation for this struct was generated from the following file:

- [b2Collision.h](#)
- 

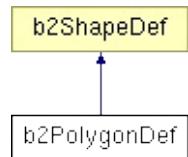
*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2PolygonDef Struct Reference

Convex polygon. [More...](#)

Inheritance diagram for b2PolygonDef:



[List of all members.](#)

## Public Member Functions

void	<a href="#">SetAsBox</a> (float32 hx, float32 hy) <i>Build vertices to represent an axis-aligned box.</i>
void	<a href="#">SetAsBox</a> (float32 hx, float32 hy, const <b>b2Vec2</b> &center, float32 angle) <i>Build vertices to represent an oriented box.</i>

## Public Attributes

b2Vec2	<b>vertices</b> [b2_maxPolygonVertices]
<i>The polygon vertices in local coordinates.</i>	
int32	<b>vertexCount</b>
<i>The number of polygon vertices.</i>	

---

## **Detailed Description**

Convex polygon.

The vertices must be in CCW order for a right-handed coordinate system with the z-axis coming out of the screen.

---

## Member Function Documentation

```
void b2PolygonDef::SetAsBox ( float32 hx,  
                           float32 hy  
                         )
```

Build vertices to represent an axis-aligned box.

### Parameters:

*hx* the half-width.

*hy* the half-height.

```
void b2PolygonDef::SetAsBox ( float32      hx,  
                           float32      hy,  
                           const b2Vec2 & center,  
                           float32      angle  
                         )
```

Build vertices to represent an oriented box.

### Parameters:

*hx* the half-width.

*hy* the half-height.

*center* the center of the box in local coordinates.

*angle* the rotation of the box in local coordinates.

---

The documentation for this struct was generated from the following files:

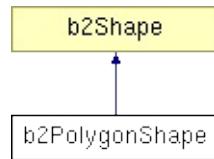
- b2PolygonShape.h
- b2PolygonShape.cpp

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2PolygonShape Class Reference

A convex polygon. [More...](#)

Inheritance diagram for b2PolygonShape:



[List of all members.](#)

## Public Member Functions

bool	<b>TestPoint</b> (const <b>b2XForm</b> &transform, const <b>b2Vec2</b> &p) const  <b>See also:</b> <i><a href="#">b2Shape::TestPoint</a></i>
bool	<b>TestSegment</b> (const <b>b2XForm</b> &transform, float32 *lambda, <b>b2Vec2</b> *normal, const <b>b2Segment</b> &segment, float32 maxLambda) const  <b>See also:</b> <i><a href="#">b2Shape::TestSegment</a></i>
void	<b>ComputeAABB</b> ( <b>b2AABB</b> *aabb, const <b>b2XForm</b> &transform) const  <b>See also:</b> <i><a href="#">b2Shape::ComputeAABB</a></i>
void	<b>ComputeSweptAABB</b> ( <b>b2AABB</b> *aabb, const <b>b2XForm</b> &transform1, const <b>b2XForm</b> &transform2) const  <b>See also:</b> <i><a href="#">b2Shape::ComputeSweptAABB</a></i>
void	<b>ComputeMass</b> ( <b>b2MassData</b> *massData) const  <b>See also:</b> <i><a href="#">b2Shape::ComputeMass</a></i>
const <b>b2OBB</b> &	<b>GetOBB</b> () const <i>Get the oriented bounding box relative to the parent body.</i>
const <b>b2Vec2</b> &	<b>GetCentroid</b> () const <i>Get local centroid relative to the parent body.</i>
int32	<b>GetVertexCount</b> () const <i>Get the vertex count.</i>
const <b>b2Vec2</b> *	<b>GetVertices</b> () const <i>Get the vertices in local coordinates.</i>
const <b>b2Vec2</b> *	<b>GetCoreVertices</b> () const <i>Get the core vertices in local coordinates.</i>

const <b>b2Vec2</b> *	<b>GetNormals</b> () const <i>Get the edge normal vectors. There is one for each vertex.</i>
<b>b2Vec2</b>	<b>GetFirstVertex</b> (const <b>b2XForm</b> &xf) const <i>Get the first vertex and apply the supplied transform.</i>
<b>b2Vec2</b>	<b>Centroid</b> (const <b>b2XForm</b> &xf) const <i>Get the centroid and apply the supplied transform.</i>
<b>b2Vec2</b>	<b>Support</b> (const <b>b2XForm</b> &xf, const <b>b2Vec2</b> &d) const <i>Get the support point in the given world direction.</i>

## Friends

class [b2Shape](#)

---

## **Detailed Description**

A convex polygon.

---

## Member Function Documentation

```
const b2Vec2 * b2PolygonShape::GetCoreVertices( ) const [inline]
```

Get the core vertices in local coordinates.

These vertices represent a smaller polygon that is used for time of impact computations.

```
b2Vec2 b2PolygonShape::Support( const b2XForm & xf,  
                                const b2Vec2 & d  
                                ) const
```

Get the support point in the given world direction.

Use the supplied transform.

---

The documentation for this class was generated from the following files:

- b2PolygonShape.h
- b2PolygonShape.cpp

[Main Page](#) [Classes](#) [Files](#)

---

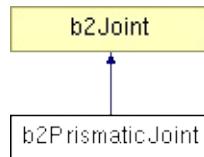
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2PrismaticJoint Class Reference

A prismatic joint. [More...](#)

Inheritance diagram for b2PrismaticJoint:



[List of all members.](#)

## Public Member Functions

b2Vec2	<b>GetAnchor1</b> () const <i>Get the anchor point on body1 in world coordinates.</i>
b2Vec2	<b>GetAnchor2</b> () const <i>Get the anchor point on body2 in world coordinates.</i>
b2Vec2	<b>GetReactionForce</b> () const <i>Get the reaction force on body2 at the joint anchor.</i>
float32	<b>GetReactionTorque</b> () const <i>Get the reaction torque on body2.</i>
float32	<b>GetJointTranslation</b> () const <i>Get the current joint translation, usually in meters.</i>
float32	<b>GetJointSpeed</b> () const <i>Get the current joint translation speed, usually in meters per second.</i>
bool	<b>IsLimitEnabled</b> () const <i>Is the joint limit enabled?</i>
void	<b>EnableLimit</b> (bool flag) <i>Enable/disable the joint limit.</i>
float32	<b>GetLowerLimit</b> () const <i>Get the lower joint limit, usually in meters.</i>
float32	<b>GetUpperLimit</b> () const <i>Get the upper joint limit, usually in meters.</i>
void	<b>SetLimits</b> (float32 lower, float32 upper) <i>Set the joint limits, usually in meters.</i>
bool	<b>IsMotorEnabled</b> () const <i>Is the joint motor enabled?</i>
void	<b>EnableMotor</b> (bool flag) <i>Enable/disable the joint motor.</i>
void	<b>SetMotorSpeed</b> (float32 speed) <i>Set the motor speed, usually in meters per second.</i>
float32	<b>GetMotorSpeed</b> () const <i>Get the motor speed, usually in meters per second.</i>
void	<b>SetMaxMotorForce</b> (float32 force) <i>Set the maximum motor force, usually in N.</i>
float32	<b>GetMotorForce</b> () const <i>Get the current motor force, usually in N.</i>

---

## Detailed Description

A prismatic joint.

This joint provides one degree of freedom: translation along an axis fixed in body1. Relative rotation is prevented. You can use a joint limit to restrict the range of motion and a joint motor to drive the motion or to model joint friction.

---

The documentation for this class was generated from the following files:

- b2PrismaticJoint.h
- b2PrismaticJoint.cpp

[Main Page](#) [Classes](#) [Files](#)

---

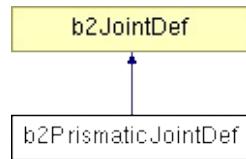
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2PrismaticJointDef Struct Reference

Prismatic joint definition. [More...](#)

Inheritance diagram for b2PrismaticJointDef:



[List of all members.](#)

## Public Member Functions

void	<b>Initialize</b> ( <b>b2Body</b> * <b>body1</b> , <b>b2Body</b> * <b>body2</b> , const <b>b2Vec2</b> & <b>anchor</b> , const <b>b2Vec2</b> & <b>axis</b> )
------	---

*Initialize the bodies, anchors, axis, and reference angle using the world anchor and world axis.*

## Public Attributes

b2Vec2	<b>localAnchor1</b>
	<i>The local anchor point relative to body1's origin.</i>
b2Vec2	<b>localAnchor2</b>
	<i>The local anchor point relative to body2's origin.</i>
b2Vec2	<b>localAxis1</b>
	<i>The local translation axis in body1.</i>
float32	<b>referenceAngle</b>
	<i>The constrained angle between the bodies: body2_angle - body1_angle.</i>
bool	<b>enableLimit</b>
	<i>Enable/disable the joint limit.</i>
float32	<b>lowerTranslation</b>
	<i>The lower translation limit, usually in meters.</i>
float32	<b>upperTranslation</b>
	<i>The upper translation limit, usually in meters.</i>
bool	<b>enableMotor</b>
	<i>Enable/disable the joint motor.</i>
float32	<b>maxMotorForce</b>
	<i>The maximum motor torque, usually in N-m.</i>
float32	<b>motorSpeed</b>
	<i>The desired motor speed in radians per second.</i>

---

## Detailed Description

Prismatic joint definition.

This requires defining a line of motion using an axis and an anchor point. The definition uses local anchor points and a local axis so that the initial configuration can violate the constraint slightly. The joint translation is zero when the local anchor points coincide in world space. Using local anchors and a local axis helps when saving and loading a game.

---

## Member Function Documentation

```
void b2PrismaticJointDef::Initialize ( b2Body *      body1,  
                                      b2Body *      body2,  
                                      const b2Vec2 & anchor,  
                                      const b2Vec2 & axis  
                                     )
```

Initialize the bodies, anchors, axis, and reference angle using the world anchor and world axis.

---

The documentation for this struct was generated from the following files:

- b2PrismaticJoint.h
  - b2PrismaticJoint.cpp
-

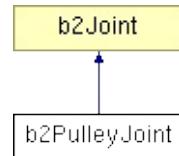
[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2PulleyJoint Class Reference

The pulley joint is connected to two bodies and two fixed ground points.

[More...](#)

Inheritance diagram for b2PulleyJoint:



[List of all members.](#)

## Public Member Functions

<b>b2Vec2</b>	<b>GetAnchor1 () const</b> <i>Get the anchor point on body1 in world coordinates.</i>
<b>b2Vec2</b>	<b>GetAnchor2 () const</b> <i>Get the anchor point on body2 in world coordinates.</i>
<b>b2Vec2</b>	<b>GetReactionForce () const</b> <i>Get the reaction force on body2 at the joint anchor.</i>
float32	<b>GetReactionTorque () const</b> <i>Get the reaction torque on body2.</i>
<b>b2Vec2</b>	<b>GetGroundAnchor1 () const</b> <i>Get the first ground anchor.</i>
<b>b2Vec2</b>	<b>GetGroundAnchor2 () const</b> <i>Get the second ground anchor.</i>
float32	<b>GetLength1 () const</b> <i>Get the current length of the segment attached to body1.</i>
float32	<b>GetLength2 () const</b> <i>Get the current length of the segment attached to body2.</i>
float32	<b>GetRatio () const</b> <i>Get the pulley ratio.</i>

## Detailed Description

The pulley joint is connected to two bodies and two fixed ground points.

The pulley supports a ratio such that: length1 + ratio \* length2 <= constant Yes, the force transmitted is scaled by the ratio. The pulley also enforces a maximum length limit on both sides. This is useful to prevent one side of the pulley hitting the top.

---

The documentation for this class was generated from the following files:

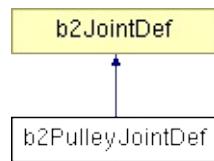
- b2PulleyJoint.h
  - b2PulleyJoint.cpp
-

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2PulleyJointDef Struct Reference

Pulley joint definition. [More...](#)

Inheritance diagram for b2PulleyJointDef:



[List of all members.](#)

## Public Member Functions

void	<b>Initialize</b> ( <b>b2Body</b> * <b>body1</b> , <b>b2Body</b> * <b>body2</b> , const <b>b2Vec2</b> & <b>groundAnchor1</b> , const <b>b2Vec2</b> & <b>groundAnchor2</b> , const <b>b2Vec2</b> & <b>anchor1</b> , const <b>b2Vec2</b> & <b>anchor2</b> , float32 <b>ratio</b> ) <i>Initialize the bodies, anchors, lengths, max lengths, and ratio using the world anchors.</i>
------	---

## Public Attributes

b2Vec2	<b>groundAnchor1</b>
	<i>The first ground anchor in world coordinates. This point never moves.</i>
b2Vec2	<b>groundAnchor2</b>
	<i>The second ground anchor in world coordinates. This point never moves.</i>
b2Vec2	<b>localAnchor1</b>
	<i>The local anchor point relative to body1's origin.</i>
b2Vec2	<b>localAnchor2</b>
	<i>The local anchor point relative to body2's origin.</i>
float32	<b>length1</b>
	<i>The a reference length for the segment attached to body1.</i>
float32	<b>maxLength1</b>
	<i>The maximum length of the segment attached to body1.</i>
float32	<b>length2</b>
	<i>The a reference length for the segment attached to body2.</i>
float32	<b>maxLength2</b>
	<i>The maximum length of the segment attached to body2.</i>
float32	<b>ratio</b>
	<i>The pulley ratio, used to simulate a block-and-tackle.</i>

---

## Detailed Description

Pulley joint definition.

This requires two ground anchors, two dynamic body anchor points, max lengths for each side, and a pulley ratio.

---

The documentation for this struct was generated from the following files:

- b2PulleyJoint.h
  - b2PulleyJoint.cpp
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

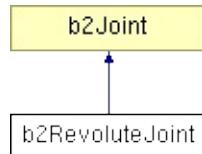
[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2RevoluteJoint Class Reference

A revolute joint constrains two bodies to share a common point while they are free to rotate about the point. [More...](#)

Inheritance diagram for b2RevoluteJoint:



[List of all members.](#)

## Public Member Functions

b2Vec2	<b>GetAnchor1</b> () const <i>Get the anchor point on body1 in world coordinates.</i>
b2Vec2	<b>GetAnchor2</b> () const <i>Get the anchor point on body2 in world coordinates.</i>
b2Vec2	<b>GetReactionForce</b> () const <i>Get the reaction force on body2 at the joint anchor.</i>
float32	<b>GetReactionTorque</b> () const <i>Get the reaction torque on body2.</i>
float32	<b>GetJointAngle</b> () const <i>Get the current joint angle in radians.</i>
float32	<b>GetJointSpeed</b> () const <i>Get the current joint angle speed in radians per second.</i>
bool	<b>IsLimitEnabled</b> () const <i>Is the joint limit enabled?</i>
void	<b>EnableLimit</b> (bool flag) <i>Enable/disable the joint limit.</i>
float32	<b>GetLowerLimit</b> () const <i>Get the lower joint limit in radians.</i>
float32	<b>GetUpperLimit</b> () const <i>Get the upper joint limit in radians.</i>
void	<b>SetLimits</b> (float32 lower, float32 upper) <i>Set the joint limits in radians.</i>
bool	<b>IsMotorEnabled</b> () const <i>Is the joint motor enabled?</i>
void	<b>EnableMotor</b> (bool flag) <i>Enable/disable the joint motor.</i>
void	<b>SetMotorSpeed</b> (float32 speed) <i>Set the motor speed in radians per second.</i>
float32	<b>GetMotorSpeed</b> () const <i>Get the motor speed in radians per second.</i>
void	<b>SetMaxMotorTorque</b> (float32 torque) <i>Set the maximum motor torque, usually in N-m.</i>
float32	<b>GetMotorTorque</b> () const <i>Get the current motor torque, usually in N-m.</i>

---

## Detailed Description

A revolute joint constrains two bodies to share a common point while they are free to rotate about the point.

The relative rotation about the shared point is the joint angle. You can limit the relative rotation with a joint limit that specifies a lower and upper angle. You can use a motor to drive the relative rotation about the shared point. A maximum motor torque is provided so that infinite forces are not generated.

---

The documentation for this class was generated from the following files:

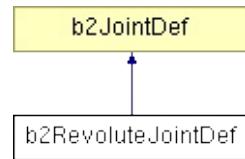
- b2RevoluteJoint.h
- b2RevoluteJoint.cpp

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2RevoluteJointDef Struct Reference

Revolute joint definition. [More...](#)

Inheritance diagram for b2RevoluteJointDef:



[List of all members.](#)

## Public Member Functions

void	<b>Initialize</b> ( <b>b2Body</b> * <b>body1</b> , <b>b2Body</b> * <b>body2</b> , const <b>b2Vec2</b> & <b>anchor</b> ) <i>Initialize the bodies, anchors, and reference angle using the world anchor.</i>
------	---

## Public Attributes

b2Vec2	<b>localAnchor1</b>
	<i>The local anchor point relative to body1's origin.</i>
b2Vec2	<b>localAnchor2</b>
	<i>The local anchor point relative to body2's origin.</i>
float32	<b>referenceAngle</b>
	<i>The body2 angle minus body1 angle in the reference state (radians).</i>
bool	<b>enableLimit</b>
	<i>A flag to enable joint limits.</i>
float32	<b>lowerAngle</b>
	<i>The lower angle for the joint limit (radians).</i>
float32	<b>upperAngle</b>
	<i>The upper angle for the joint limit (radians).</i>
bool	<b>enableMotor</b>
	<i>A flag to enable the joint motor.</i>
float32	<b>motorSpeed</b>
	<i>The desired motor speed. Usually in radians per second.</i>
float32	<b>maxMotorTorque</b>
	<i>The maximum motor torque used to achieve the desired motor speed.</i>

---

## Detailed Description

Revolute joint definition.

This requires defining an anchor point where the bodies are joined. The definition uses local anchor points so that the initial configuration can violate the constraint slightly. You also need to specify the initial relative angle for joint limits. This helps when saving and loading a game. The local anchor points are measured from the body's origin rather than the center of mass because: 1. you might not know where the center of mass will be. 2. if you add/remove shapes from a body and recompute the mass, the joints will be broken.

---

## Member Function Documentation

```
void b2RevoluteJointDef::Initialize ( b2Body *      body1,
                                      b2Body *      body2,
                                      const b2Vec2 & anchor
                                    )
```

Initialize the bodies, anchors, and reference angle using the world anchor.

---

## Member Data Documentation

**float32 b2RevoluteJointDef::maxMotorTorque**

The maximum motor torque used to achieve the desired motor speed.

Usually in N-m.

---

The documentation for this struct was generated from the following files:

- b2RevoluteJoint.h
- b2RevoluteJoint.cpp

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2Segment Struct Reference

A line segment. [More...](#)

[List of all members.](#)

## Public Member Functions

bool	<b>TestSegment</b> (float32 *lambda, <b>b2Vec2</b> *normal, const <b>b2Segment</b> &segment, float32 maxLambda) const <i>Ray cast against this segment with another segment.</i>
------	---

## Public Attributes

<b>b2Vec2</b>	<b>p1</b>
	<i>the starting point</i>
<b>b2Vec2</b>	<b>p2</b>
	<i>the ending point</i>

---

## Detailed Description

A line segment.

---

The documentation for this struct was generated from the following files:

- [b2Collision.h](#)
  - [b2Collision.cpp](#)
- 

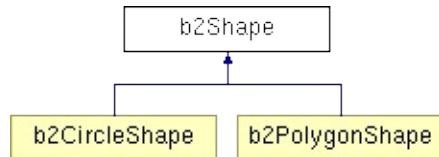
*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2Shape Class Reference

A shape is used for collision detection. [More...](#)

Inheritance diagram for b2Shape:



[List of all members.](#)

## Public Member Functions

b2ShapeType	<b>GetType</b> () const <i>Get the type of this shape.</i>
bool	<b>IsSensor</b> () const <i>Is this shape a sensor (non-solid)?</i>
void	<b>SetFilterData</b> (const <b>b2FilterData</b> &filter) <i>Set the contact filtering data.</i>
const <b>b2FilterData</b> &	<b>GetFilterData</b> () const <i>Get the contact filtering data.</i>
<b>b2Body</b> *	<b>GetBody</b> () <i>Get the parent body of this shape.</i>
<b>b2Shape</b> *	<b>GetNext</b> () <i>Get the next shape in the parent body's shape list.</i>
void *	<b>GetUserData</b> () <i>Get the user data that was assigned in the shape definition.</i>
void	<b>SetUserData</b> (void *data) <i>Set the user data. Use this to store your application specific data.</i>
virtual bool	<b>TestPoint</b> (const <b>b2XForm</b> &xf, const <b>b2Vec2</b> &p) const =0 <i>Test a point for containment in this shape.</i>
virtual bool	<b>TestSegment</b> (const <b>b2XForm</b> &xf, float32 *lambda, <b>b2Vec2</b> *normal, const <b>b2Segment</b> &segment, float32 maxLambda) const =0 <i>Perform a ray cast against this shape.</i>
virtual void	<b>ComputeAABB</b> ( <b>b2AABB</b> *aabb, const <b>b2XForm</b> &xf) const =0 <i>Given a transform, compute the associated axis aligned bounding box for this shape.</i>
virtual void	<b>ComputeSweptAABB</b> ( <b>b2AABB</b> *aabb, const <b>b2XForm</b> &xf1, const <b>b2XForm</b> &xf2) const =0 <i>Given two transforms, compute the associated swept axis aligned bounding box for this shape.</i>
virtual void	<b>ComputeMass</b> ( <b>b2MassData</b> *massData) const =0 <i>Compute the mass properties of this shape using its dimensions and density.</i>
float32	<b>GetSweepRadius</b> () const <i>Get the maximum radius about the parent body's center of mass.</i>
float32	<b>GetFriction</b> () const <i>Get the coefficient of friction.</i>
float32	<b>GetRestitution</b> () const <i>Get the coefficient of restitution.</i>

## Friends

class [b2Body](#)

class [b2World](#)

---

## Detailed Description

A shape is used for collision detection.

Shapes are created in **b2World**. You can use shape for collision detection before they are attached to the world.

### **Warning:**

you cannot reuse shapes.

---

## Member Function Documentation

```
b2ShapeType b2Shape::GetType( ) const [inline]
```

Get the type of this shape.

You can use this to down cast to the concrete shape.

**Returns:**

the shape type.

```
bool b2Shape::IsSensor( ) const [inline]
```

Is this shape a sensor (non-solid)?

**Returns:**

the true if the shape is a sensor.

```
void b2Shape::SetFilterData( const b2FilterData & filter ) [inline]
```

Set the contact filtering data.

You must call **b2World::Refilter** to correct existing contacts/non-contacts.

```
b2Body * b2Shape::GetBody( ) [inline]
```

Get the parent body of this shape.

This is NULL if the shape is not attached.

**Returns:**

the parent body.

```
b2Shape * b2Shape::GetNext( ) [inline]
```

Get the next shape in the parent body's shape list.

**Returns:**

the next shape.

```
void * b2Shape::GetUserData( ) [inline]
```

Get the user data that was assigned in the shape definition.

Use this to store your application specific data.

```
virtual bool b2Shape::TestPoint ( const b2XForm & xf,  
                                const b2Vec2 & p  
                                ) const [pure virtual]
```

Test a point for containment in this shape.

This only works for convex shapes.

**Parameters:**

*xf* the shape world transform.  
*p* a point in world coordinates.

Implemented in **b2CircleShape**, and **b2PolygonShape**.

```
virtual bool b2Shape::TestSegment ( const b2XForm & xf,  
                                    float32 * lambda,  
                                    b2Vec2 * normal,  
                                    const b2Segment & segment,  
                                    float32 maxLambda  
                                    ) const [pure virtual]
```

Perform a ray cast against this shape.

**Parameters:**

<i>xf</i>	the shape world transform.
<i>lambda</i>	returns the hit fraction. You can use this to compute the contact point $p = (1 - \lambda) * segment.p1 + \lambda * segment.p2$ .
<i>normal</i>	returns the normal at the contact point. If there is no intersection, the normal is not set.
<i>segment</i>	defines the begin and end point of the ray cast.
<i>maxLambda</i>	a number typically in the range [0,1].

**Returns:**

true if there was an intersection.

Implemented in **b2CircleShape**, and **b2PolygonShape**.

```
virtual void b2Shape::ComputeAABB ( b2AABB * aabb,
                                    const b2XForm & xf
                                )
                                    const [pure virtual]
```

Given a transform, compute the associated axis aligned bounding box for this shape.

**Parameters:**

*aabb* returns the axis aligned box.  
*xf* the world transform of the shape.

Implemented in **b2CircleShape**, and **b2PolygonShape**.

```
virtual void b2Shape::ComputeSweptAABB ( b2AABB * aabb,
                                         const b2XForm & xf1,
                                         const b2XForm & xf2
                                       )
                                         const [pure virtual]
```

Given two transforms, compute the associated swept axis aligned bounding box for this shape.

**Parameters:**

*aabb* returns the axis aligned box.  
*xf1* the starting shape world transform.  
*xf2* the ending shape world transform.

Implemented in **b2CircleShape**, and **b2PolygonShape**.

```
virtual void b2Shape::ComputeMass ( b2MassData * massData ) const [pure virtual]
```

Compute the mass properties of this shape using its dimensions and density.

The inertia tensor is computed about the local origin, not the centroid.

**Parameters:**

*massData* returns the mass data for this shape.

Implemented in **b2CircleShape**, and **b2PolygonShape**.

---

The documentation for this class was generated from the following files:

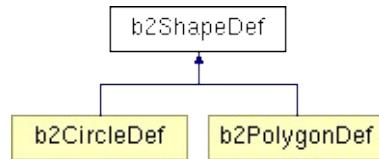
- b2Shape.h
- b2Shape.cpp

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2ShapeDef Struct Reference

A shape definition is used to construct a shape. [More...](#)

Inheritance diagram for b2ShapeDef:



[List of all members.](#)

## Public Member Functions

### **b2ShapeDef ()**

*The constructor sets the default shape definition values.*

## Public Attributes

b2ShapeType	<b>type</b> <i>Holds the shape type for down-casting.</i>
void *	<b>userData</b> <i>Use this to store application specify shape data.</i>
float32	<b>friction</b> <i>The shape's friction coefficient, usually in the range [0,1].</i>
float32	<b>restitution</b> <i>The shape's restitution (elasticity) usually in the range [0,1].</i>
float32	<b>density</b> <i>The shape's density, usually in kg/m^2.</i>
bool	<b>isSensor</b> <i>A sensor shape collects contact information but never generates a collision response.</i>
<b>b2FilterData</b>	<b>filter</b> <i>Contact filtering data.</i>

## **Detailed Description**

A shape definition is used to construct a shape.

This class defines an abstract shape definition. You can reuse shape definitions safely.

---

## Member Data Documentation

`bool b2ShapeDef::isSensor`

A sensor shape collects contact information but never generates a collision response.

---

The documentation for this struct was generated from the following file:

- `b2Shape.h`
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Sweep Struct Reference

This describes the motion of a body/shape for TOI computation. [More...](#)

[List of all members.](#)

## Public Member Functions

void	<b>GetXForm</b> ( <b>b2XForm</b> *xf, float32 t) const <i>Get the interpolated transform at a specific time.</i>
void	<b>Advance</b> (float32 t) <i>Advance the sweep forward, yielding a new initial state.</i>

## Public Attributes

b2Vec2	<b>localCenter</b>
	<i>local center of mass position</i>
b2Vec2	<b>c</b>
	<i>center world positions</i>
float32	<b>a</b>
	<i>world angles</i>
float32	<b>t0</b>
	<i>time interval = [t0,1], where t0 is in [0,1]</i>

---

## **Detailed Description**

This describes the motion of a body/shape for TOI computation.

Shapes are defined with respect to the body origin, which may no coincide with the center of mass. However, to support dynamics we must interpolate the center of mass position.

---

## Member Function Documentation

```
void b2Sweep::GetXForm ( b2XForm * xf,  
                         float32      t  
                     ) const
```

Get the interpolated transform at a specific time.

### Parameters:

*t* the normalized time in [0,1].

```
void b2Sweep::Advance ( float32 t )
```

Advance the sweep forward, yielding a new initial state.

### Parameters:

*t* the new initial time.

---

The documentation for this struct was generated from the following files:

- b2Math.h
- b2Math.cpp

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Vec2 Struct Reference

A 2D column vector. [More...](#)

[List of all members.](#)

## Public Member Functions

	<b>b2Vec2 ()</b> <i>Default constructor does nothing (for performance).</i>
	<b>b2Vec2 (float32 x, float32 y)</b> <i>Construct using coordinates.</i>
void	<b>SetZero ()</b> <i>Set this vector to all zeros.</i>
void	<b>Set (float32 x_, float32 y_)</b> <i>Set this vector to some specified coordinates.</i>
<b>b2Vec2</b>	<b>operator- () const</b> <i>Negate this vector.</i>
void	<b>operator+= (const b2Vec2 &amp;v)</b> <i>Add a vector to this vector.</i>
void	<b>operator-= (const b2Vec2 &amp;v)</b> <i>Subtract a vector from this vector.</i>
void	<b>operator *= (float32 a)</b> <i>Multiply this vector by a scalar.</i>
float32	<b>Length () const</b> <i>Get the length of this vector (the norm).</i>
float32	<b>LengthSquared () const</b> <i>Get the length squared.</i>
float32	<b>Normalize ()</b> <i>Convert this vector into a unit vector. Returns the length.</i>
bool	<b>IsValid () const</b> <i>Does this vector contain finite coordinates?</i>

## **Detailed Description**

A 2D column vector.

---

## Member Function Documentation

```
float32 b2Vec2::LengthSquared( ) const [inline]
```

Get the length squared.

For performance, use this instead of **b2Vec2::Length** (if possible).

---

The documentation for this struct was generated from the following file:

- b2Math.h

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Version Struct Reference

Version numbering scheme. [More...](#)

[List of all members.](#)

## Public Attributes

int32	<b>major</b> <i>significant changes</i>
int32	<b>minor</b> <i>incremental changes</i>
int32	<b>revision</b> <i>bug fixes</i>

---

## Detailed Description

Version numbering scheme.

See [http://en.wikipedia.org/wiki/Software\\_versioning](http://en.wikipedia.org/wiki/Software_versioning)

---

The documentation for this struct was generated from the following file:

- [b2Settings.h](#)

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2World Class Reference

The world class manages all physics entities, dynamic simulation, and asynchronous queries. [More...](#)

[List of all members.](#)

## Public Member Functions

	<b>b2World</b> (const <b>b2AABB</b> &worldAABB, const <b>b2Vec2</b> &gravity, bool doSleep) <i>Construct a world object.</i>
	<b>~b2World ()</b> <i>Destruct the world. All physics entities are destroyed and all heap memory is released.</i>
void	<b>SetDestructionListener</b> ( <b>b2DestructionListener</b> *listener) <i>Register a destruction listener.</i>
void	<b>SetBoundaryListener</b> ( <b>b2BoundaryListener</b> *listener) <i>Register a broad-phase boundary listener.</i>
void	<b>SetContactFilter</b> ( <b>b2ContactFilter</b> *filter) <i>Register a contact filter to provide specific control over collision.</i>
void	<b>SetContactListener</b> ( <b>b2ContactListener</b> *listener) <i>Register a contact event listener.</i>
void	<b>SetDebugDraw</b> ( <b>b2DebugDraw</b> *debugDraw) <i>Register a routine for debug drawing.</i>
<b>b2Body</b> *	<b>CreateBody</b> (const <b>b2BodyDef</b> *def) <i>Create a rigid body given a definition.</i>
void	<b>DestroyBody</b> ( <b>b2Body</b> *body) <i>Destroy a rigid body given a definition.</i>
<b>b2Joint</b> *	<b>CreateJoint</b> (const <b>b2JointDef</b> *def) <i>Create a joint to constrain bodies together.</i>
void	<b>DestroyJoint</b> ( <b>b2Joint</b> *joint) <i>Destroy a joint.</i>
<b>b2Body</b> *	<b>GetGroundBody</b> () <i>The world provides a single static ground body with no collision shapes.</i>
void	<b>Step</b> (float32 timeStep, int32 iterations) <i>Take a time step.</i>
int32	<b>Query</b> (const <b>b2AABB</b> &aabb, <b>b2Shape</b> **shapes, int32 maxCount) <i>Query the world for all shapes that potentially overlap the provided AABB.</i>
<b>b2Body</b> *	<b>GetBodyList</b> () <i>Get the world body list.</i>
<b>b2Joint</b> *	<b>GetJointList</b> () <i>Get the world joint list.</i>
void	<b>Refilter</b> ( <b>b2Shape</b> *shape) <i>Re-filter a shape. This re-runs contact filtering on a shape.</i>
void	<b>SetWarmStarting</b> (bool flag) <i>Enable/disable warm starting. For testing.</i>

void	<b>SetPositionCorrection</b> (bool flag) <i>Enable/disable position correction. For testing.</i>
void	<b>SetContinuousPhysics</b> (bool flag) <i>Enable/disable continuous physics. For testing.</i>
void	<b>Validate</b> () <i>Perform validation of internal data structures.</i>
int32	<b>GetProxyCount</b> () const <i>Get the number of broad-phase proxies.</i>
int32	<b>GetPairCount</b> () const <i>Get the number of broad-phase pairs.</i>
int32	<b>GetBodyCount</b> () const <i>Get the number of bodies.</i>
int32	<b>GetJointCount</b> () const <i>Get the number joints.</i>
int32	<b>GetContactCount</b> () const <i>Get the number of contacts (each may have 0 or more contact points).</i>
void	<b>SetGravity</b> (const <b>b2Vec2</b> &gravity) <i>Change the global gravity vector.</i>

## Friends

class **b2Body**

---

## **Detailed Description**

The world class manages all physics entities, dynamic simulation, and asynchronous queries.

The world also contains efficient memory management facilities.

---

## Constructor & Destructor Documentation

```
b2World::b2World ( const b2AABB & worldAABB,  
                    const b2Vec2 & gravity,  
                    bool doSleep  
                )
```

Construct a world object.

### Parameters:

*worldAABB* a bounding box that completely encompasses all your shapes.  
*gravity* the world gravity vector.  
*doSleep* improve performance by not simulating inactive bodies.

---

## Member Function Documentation

```
void b2World::SetContactFilter ( b2ContactFilter * filter )
```

Register a contact filter to provide specific control over collision.

Otherwise the default filter is used (b2\_defaultFilter).

```
void b2World::SetDebugDraw ( b2DebugDraw * debugDraw )
```

Register a routine for debug drawing.

The debug draw functions are called inside the [b2World::Step](#) method, so make sure your renderer is ready to consume draw commands when you call [Step\(\)](#).

```
b2Body * b2World::CreateBody ( const b2BodyDef * def )
```

Create a rigid body given a definition.

No reference to the definition is retained.

**Warning:**

This function is locked during callbacks.

```
void b2World::DestroyBody ( b2Body * body )
```

Destroy a rigid body given a definition.

No reference to the definition is retained. This function is locked during callbacks.

**Warning:**

This automatically deletes all associated shapes and joints.

This function is locked during callbacks.

```
b2Joint * b2World::CreateJoint ( const b2JointDef * def )
```

Create a joint to constrain bodies together.

No reference to the definition is retained. This may cause the connected bodies to cease colliding.

**Warning:**

This function is locked during callbacks.

```
void b2World::DestroyJoint ( b2Joint * joint )
```

Destroy a joint.

This may cause the connected bodies to begin colliding.

**Warning:**

This function is locked during callbacks.

```
b2Body * b2World::GetGroundBody ( ) [inline]
```

The world provides a single static ground body with no collision shapes.

You can use this to simplify the creation of joints and static shapes.

```
void b2World::Step ( float32 timeStep,
                     int32 iterations
                   )
```

Take a time step.

This performs collision detection, integration, and constraint solution.

## Parameters:

*timeStep* the amount of time to simulate, this should not vary.  
*iterations* the number of iterations to be used by the constraint solver.

```
int32 b2World::Query ( const b2AABB & aabb,  
                      b2Shape **      shapes,  
                      int32            maxCount  
)
```

Query the world for all shapes that potentially overlap the provided AABB.

You provide a shape pointer buffer of specified size. The number of shapes found is returned.

## Parameters:

*aabb* the query box.  
*shapes* a user allocated shape pointer array of size *maxCount* (or greater).  
*maxCount* the capacity of the shapes array.

## Returns:

the number of shapes found in *aabb*.

```
b2Body * b2World::GetBodyList ( ) [inline]
```

Get the world body list.

With the returned body, use **b2Body::GetNext** to get the next body in the world list. A NULL body indicates the end of the list.

## Returns:

the head of the world body list.

```
b2Joint * b2World::GetJointList ( ) [inline]
```

Get the world joint list.

With the returned joint, use **b2Joint::GetNext** to get the next joint in the world list. A NULL joint indicates the end of the list.

**Returns:**

the head of the world joint list.

---

The documentation for this class was generated from the following files:

- b2World.h
- b2World.cpp

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2XForm Struct Reference

A transform contains translation and rotation. [More...](#)

[List of all members.](#)

## Public Member Functions

	<b>b2XForm ()</b> <i>The default constructor does nothing (for performance).</i>
	<b>b2XForm (const b2Vec2 &amp;position, const b2Mat22 &amp;R)</b> <i>Initialize using a position vector and a rotation matrix.</i>
void	<b>SetIdentity ()</b> <i>Set this to the identity transform.</i>

---

## Detailed Description

A transform contains translation and rotation.

It is used to represent the position and orientation of rigid frames.

---

The documentation for this struct was generated from the following file:

- b2Math.h
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# Box2D Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

- [b2AABB](#)
- [b2Body](#)
- [b2BodyDef](#)
- [b2BoundaryListener](#)
- [b2Color](#)
- [b2Contact](#)
- [b2ContactEdge](#)
- [b2ContactFilter](#)
- [b2ContactID](#)
- [b2ContactID::Features](#)
- [b2ContactListener](#)
- [b2ContactPoint](#)
- [b2ContactResult](#)
- [b2DebugDraw](#)
- [b2DestructionListener](#)
- [b2FilterData](#)
- [b2Joint](#)
  - [b2DistanceJoint](#)
  - [b2GearJoint](#)
  - [b2MouseJoint](#)
  - [b2PrismaticJoint](#)
  - [b2PulleyJoint](#)
  - [b2RevoluteJoint](#)
- [b2JointDef](#)
  - [b2DistanceJointDef](#)
  - [b2GearJointDef](#)
  - [b2MouseJointDef](#)
  - [b2PrismaticJointDef](#)
  - [b2PulleyJointDef](#)
  - [b2RevoluteJointDef](#)
- [b2JointEdge](#)
- [b2Manifold](#)
- [b2ManifoldPoint](#)
- [b2MassData](#)

- **b2Mat22**
  - **b2OBB**
  - **b2Segment**
  - **b2Shape**
    - **b2CircleShape**
    - **b2PolygonShape**
  - **b2ShapeDef**
    - **b2CircleDef**
    - **b2PolygonDef**
  - **b2Sweep**
  - **b2Vec2**
  - **b2Version**
  - **b2World**
  - **b2XForm**
- 

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

The screenshot shows a top navigation bar with several tabs: 'Main Page', 'Classes', 'Files', 'Alphabetical List', 'Class List', 'Class Hierarchy', and 'Class Members'. Below this is a secondary navigation bar with tabs: 'All', 'Functions', 'Variables', and 'Enumerator'. At the bottom is a horizontal search bar containing letters from 'a' to 'x' and a tilde (~), with the letter 'a' currently highlighted.

Here is a list of all documented class members with links to the class documentation for each member:

- a -

- a : [b2Sweep](#)
- Add() : [b2ContactListener](#)
- Advance() : [b2Sweep](#)
- allowSleep : [b2BodyDef](#)
- AllowSleeping() : [b2Body](#)
- angle : [b2BodyDef](#)
- angularDamping : [b2BodyDef](#)
- AppendFlags() : [b2DebugDraw](#)
- ApplyForce() : [b2Body](#)
- ApplyImpulse() : [b2Body](#)
- ApplyTorque() : [b2Body](#)

[Main Page](#)[Classes](#)[Files](#)[File List](#)[File Members](#)

# Box2D File List

Here is a list of all documented files with brief descriptions:

<a href="#">b2Collision.h</a>	<i>Structures and functions used for computing contact points, distance queries, and TOI queries</i>
<a href="#">b2Settings.h</a>	<i>Global tuning constants based on meters-kilograms-seconds (MKS) units</i>

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#)

[Classes](#)

[Files](#)

---

## b2Collision.h File Reference

Structures and functions used for computing contact points, distance queries, and TOI queries. [More...](#)

## Classes

union	<b>b2ContactID</b> <i>Contact ids to facilitate warm starting.</i> <a href="#">More...</a>
struct	<b>b2ContactID::Features</b> <i>The features that intersect to form the contact point.</i> <a href="#">More...</a>
struct	<b>b2ManifoldPoint</b> <i>A manifold point is a contact point belonging to a contact manifold.</i> <a href="#">More...</a>
struct	<b>b2Manifold</b> <i>A manifold for two touching convex shapes.</i> <a href="#">More...</a>
struct	<b>b2Segment</b> <i>A line segment.</i> <a href="#">More...</a>
struct	<b>b2AABB</b> <i>An axis aligned bounding box.</i> <a href="#">More...</a>
struct	<b>b2OBB</b> <i>An oriented bounding box.</i> <a href="#">More...</a>

## Functions

void	<b>b2CollideCircles</b> ( <b>b2Manifold</b> *manifold, const <b>b2CircleShape</b> *circle1, const <b>b2XForm</b> &xf1, const <b>b2CircleShape</b> *circle2, const <b>b2XForm</b> &xf2) <i>Compute the collision manifold between two circles.</i>
void	<b>b2CollidePolygonAndCircle</b> ( <b>b2Manifold</b> *manifold, const <b>b2PolygonShape</b> *polygon, const <b>b2XForm</b> &xf1, const <b>b2CircleShape</b> *circle, const <b>b2XForm</b> &xf2) <i>Compute the collision manifold between a polygon and a circle.</i>
void	<b>b2CollidePolygons</b> ( <b>b2Manifold</b> *manifold, const <b>b2PolygonShape</b> *polygon1, const <b>b2XForm</b> &xf1, const <b>b2PolygonShape</b> *polygon2, const <b>b2XForm</b> &xf2) <i>Compute the collision manifold between two circles.</i>
float32	<b>b2Distance</b> ( <b>b2Vec2</b> *x1, <b>b2Vec2</b> *x2, const <b>b2Shape</b> *shape1, const <b>b2XForm</b> &xf1, const <b>b2Shape</b> *shape2, const <b>b2XForm</b> &xf2) <i>Compute the distance between two shapes and the closest points.</i>
float32	<b>b2TimeOfImpact</b> (const <b>b2Shape</b> *shape1, const <b>b2Sweep</b> &sweep1, const <b>b2Shape</b> *shape2, const <b>b2Sweep</b> &sweep2) <i>Compute the time when two shapes begin to touch or touch at a closer distance.</i>

## **Detailed Description**

Structures and functions used for computing contact points, distance queries, and TOI queries.

---

## Function Documentation

```
float32 b2Distance ( b2Vec2 *      x1,
                     b2Vec2 *      x2,
                     const b2Shape * shape1,
                     const b2XForm & xf1,
                     const b2Shape * shape2,
                     const b2XForm & xf2
)
```

Compute the distance between two shapes and the closest points.

**Returns:**

the distance between the shapes or zero if they are overlapped/touching.

```
float32 b2TimeOfImpact ( const b2Shape * shape1,
                         const b2Sweep & sweep1,
                         const b2Shape * shape2,
                         const b2Sweep & sweep2
)
```

Compute the time when two shapes begin to touch or touch at a closer distance.

**Warning:**

the sweeps must have the same time interval.

**Returns:**

the fraction between [0,1] in which the shapes first touch.  
fraction=0 means the shapes begin touching/overlapped, and  
fraction=1 means the shapes don't touch.

[Main Page](#)

[Classes](#)

[Files](#)

---

## b2Settings.h File Reference

Global tuning constants based on meters-kilograms-seconds (MKS) units. [More...](#)

## Classes

struct **b2Version**

*Version numbering scheme. [More...](#)*

## Functions

void *	<b>b2Alloc</b> (int32 size) <i>Implement this function to use your own memory allocator.</i>
void	<b>b2Free</b> (void *mem) <i>If you implement b2Alloc, you should also implement this function.</i>
float32	<b>b2MixFriction</b> (float32 friction1, float32 friction2) <i>Friction mixing law. Feel free to customize this.</i>
float32	<b>b2MixRestitution</b> (float32 restitution1, float32 restitution2) <i>Restitution mixing law. Feel free to customize this.</i>

## Variables

const float32	<b>b2_linearSlop</b> = 0.005f <i>A small length used as a collision and constraint tolerance.</i>
const float32	<b>b2_angularSlop</b> = 2.0f / 180.0f * b2_pi <i>A small angle used as a collision and constraint tolerance.</i>
const float32	<b>b2_toiSlop</b> = 8.0f * b2_linearSlop <i>Continuous collision detection (CCD) works with core, shrunken shapes.</i>
const int32	<b>b2_maxTOIContactsPerIsland</b> = 32 <i>Maximum number of contacts to be handled to solve a TOI island.</i>
const float32	<b>b2_velocityThreshold</b> = 1.0f <i>A velocity threshold for elastic collisions.</i>
const float32	<b>b2_maxLinearCorrection</b> = 0.2f <i>The maximum linear position correction used when solving constraints.</i>
const float32	<b>b2_maxAngularCorrection</b> = 8.0f / 180.0f * b2_pi <i>The maximum angular position correction used when solving constraints.</i>
const float32	<b>b2_maxLinearVelocity</b> = 200.0f <i>The maximum linear velocity of a body.</i>
const float32	<b>b2_maxAngularVelocity</b> = 250.0f <i>The maximum angular velocity of a body.</i>
const float32	<b>b2_contactBaumgarte</b> = 0.2f <i>This scale factor controls how fast overlap is resolved.</i>
const float32	<b>b2_timeToSleep</b> = 0.5f <i>The time that a body must be still before it will go to sleep.</i>
const float32	<b>b2_linearSleepTolerance</b> = 0.01f <i>A body cannot sleep if its linear velocity is above this tolerance.</i>
const float32	<b>b2_angularSleepTolerance</b> = 2.0f / 180.0f <i>A body cannot sleep if its angular velocity is above this tolerance.</i>
int32	<b>b2_byteCount</b> <i>The current number of bytes allocated through b2Alloc.</i>
<b>b2Version</b>	<b>b2_version</b> <i>Current version.</i>

## **Detailed Description**

Global tuning constants based on meters-kilograms-seconds (MKS) units.

---

## Variable Documentation

```
const float32 b2_angularSlop = 2.0f / 180.0f * b2_pi
```

A small angle used as a collision and constraint tolerance.

Usually it is chosen to be numerically significant, but visually insignificant.

```
const float32 b2_contactBaumgarte = 0.2f
```

This scale factor controls how fast overlap is resolved.

Ideally this would be 1 so that overlap is removed in one time step. However using values close to 1 often lead to overshoot.

```
const float32 b2_linearSlop = 0.005f
```

A small length used as a collision and constraint tolerance.

Usually it is chosen to be numerically significant, but visually insignificant.

```
const float32 b2_maxAngularCorrection = 8.0f / 180.0f * b2_pi
```

The maximum angular position correction used when solving constraints.

This helps to prevent overshoot.

```
const float32 b2_maxAngularVelocity = 250.0f
```

The maximum angular velocity of a body.

This limit is very large and is used to prevent numerical problems. You shouldn't need to adjust this.

```
const float32 b2_maxLinearCorrection = 0.2f
```

The maximum linear position correction used when solving constraints.

This helps to prevent overshoot.

```
const float32 b2_maxLinearVelocity = 200.0f
```

The maximum linear velocity of a body.

This limit is very large and is used to prevent numerical problems. You shouldn't need to adjust this.

```
const float32 b2_toiSlop = 8.0f * b2_linearSlop
```

Continuous collision detection (CCD) works with core, shrunken shapes.

This is the amount by which shapes are automatically shrunk to work with CCD. This must be larger than b2\_linearSlop.

```
const float32 b2_velocityThreshold = 1.0f
```

A velocity threshold for elastic collisions.

Any collision with a relative linear velocity below this threshold will be treated as inelastic.

Here is a list of all documented file members with links to the documentation:

- b2-angularSleepTolerance : [b2Settings.h](#)
  - b2-angularSlop : [b2Settings.h](#)
  - b2-byteCount : [b2Settings.h](#)
  - b2-contactBaumgarte : [b2Settings.h](#)
  - b2-linearSleepTolerance : [b2Settings.h](#)
  - b2-linearSlop : [b2Settings.h](#)
  - b2-maxAngularCorrection : [b2Settings.h](#)
  - b2-maxAngularVelocity : [b2Settings.h](#)
  - b2-maxLinearCorrection : [b2Settings.h](#)
  - b2-maxLinearVelocity : [b2Settings.h](#)
  - b2-maxTOIContactsPerIsland : [b2Settings.h](#)
  - b2-timeToSleep : [b2Settings.h](#)
  - b2-toiSlop : [b2Settings.h](#)
  - b2-velocityThreshold : [b2Settings.h](#)
  - b2-version : [b2Settings.h](#)
  - b2Alloc() : [b2Settings.h](#)
  - b2CollideCircles() : [b2Collision.h](#)
  - b2CollidePolygonAndCircle() : [b2Collision.h](#)
  - b2CollidePolygons() : [b2Collision.h](#)
  - b2Distance() : [b2Collision.h](#)
  - b2Free() : [b2Settings.h](#)
  - b2MixFriction() : [b2Settings.h](#)
  - b2MixRestitution() : [b2Settings.h](#)
  - b2TimeOfImpact() : [b2Collision.h](#)
-

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# Box2D Class Index

[A](#) | [B](#) | [C](#) | [D](#) | [F](#) | [G](#) | [J](#) | [M](#) | [O](#) | [P](#) | [R](#) | [S](#) | [V](#) | [W](#) | [X](#)

A	b2ContactID	b2GearJoint	O
<a href="#">b2AABB</a>	<a href="#">b2ContactID::Features</a>	<a href="#">b2GearJointDef</a>	<a href="#">b2OBB</a>
B	b2ContactListener	J	P
<a href="#">b2Body</a>	<a href="#">b2ContactPoint</a>	<a href="#">b2Joint</a>	<a href="#">b2PolygonDef</a>
<a href="#">b2BodyDef</a>	<a href="#">b2ContactResult</a>	<a href="#">b2JointDef</a>	<a href="#">b2PolygonShape</a>
<a href="#">b2BoundaryListener</a>	D	<a href="#">b2JointEdge</a>	<a href="#">b2PrismaticJoint</a>
C	b2DebugDraw	M	<a href="#">b2PrismaticJointDef</a>
<a href="#">b2CircleDef</a>	<a href="#">b2DestructionListener</a>	<a href="#">b2Manifold</a>	<a href="#">b2PulleyJoint</a>
<a href="#">b2CircleShape</a>	<a href="#">b2DistanceJoint</a>	<a href="#">b2ManifoldPoint</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">b2Color</a>	<a href="#">b2DistanceJointDef</a>	<a href="#">b2MassData</a>	R
<a href="#">b2Contact</a>	F	<a href="#">b2Mat22</a>	<a href="#">b2RevoluteJoint</a>
<a href="#">b2ContactEdge</a>	<a href="#">b2FilterData</a>	<a href="#">b2MouseJoint</a>	<a href="#">b2RevoluteJointDef</a>
<a href="#">b2ContactFilter</a>	G	<a href="#">b2MouseJointDef</a>	

[A](#) | [B](#) | [C](#) | [D](#) | [F](#) | [G](#) | [J](#) | [M](#) | [O](#) | [P](#) | [R](#) | [S](#) | [V](#) | [W](#) | [X](#)

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2AABB Member List

This is the complete list of members for **b2AABB**, including all inherited members.

<a href="#">IsValid()</a>	const	<b>b2AABB</b>	[inline]
<a href="#">lowerBound</a>		<b>b2AABB</b>	
<a href="#">upperBound</a>		<b>b2AABB</b>	

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  **doxygen** 1.5.4

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2Body Member List

This is the complete list of members for **b2Body**, including all inherited members.

<a href="#">AllowSleeping(bool flag)</a>	<b>b2Body</b> [inline]
<a href="#">ApplyForce(const b2Vec2 &amp;force, const b2Vec2 &amp;point)</a>	<b>b2Body</b> [inline]
<a href="#">ApplyImpulse(const b2Vec2 &amp;impulse, const b2Vec2 &amp;point)</a>	<b>b2Body</b> [inline]
<a href="#">ApplyTorque(float32 torque)</a>	<b>b2Body</b> [inline]
<a href="#">CreateShape(b2ShapeDef *shapeDef)</a>	<b>b2Body</b>
<a href="#">DestroyShape(b2Shape *shape)</a>	<b>b2Body</b>
<a href="#">GetAngle() const</a>	<b>b2Body</b> [inline]
<a href="#">GetAngularVelocity() const</a>	<b>b2Body</b> [inline]
<a href="#">GetInertia() const</a>	<b>b2Body</b> [inline]
<a href="#">GetJointList()</a>	<b>b2Body</b> [inline]
<a href="#">GetLinearVelocity() const</a>	<b>b2Body</b> [inline]
<a href="#">GetLinearVelocityFromLocalPoint(const b2Vec2 &amp;localPoint) const</a>	<b>b2Body</b> [inline]
<a href="#">GetLinearVelocityFromWorldPoint(const b2Vec2 &amp;worldPoint) const</a>	<b>b2Body</b> [inline]
<a href="#">GetLocalCenter() const</a>	<b>b2Body</b> [inline]
<a href="#">GetLocalPoint(const b2Vec2 &amp;worldPoint) const</a>	<b>b2Body</b> [inline]
<a href="#">GetLocalVector(const b2Vec2 &amp;worldVector) const</a>	<b>b2Body</b> [inline]
<a href="#">GetMass() const</a>	<b>b2Body</b> [inline]
<a href="#">GetNext()</a>	<b>b2Body</b> [inline]
<a href="#">GetPosition() const</a>	<b>b2Body</b> [inline]
<a href="#">GetShapeList()</a>	<b>b2Body</b> [inline]
<a href="#">GetUserData()</a>	<b>b2Body</b> [inline]
<a href="#">GetWorld()</a>	<b>b2Body</b> [inline]
<a href="#">GetWorldCenter() const</a>	<b>b2Body</b> [inline]
<a href="#">GetWorldPoint(const b2Vec2 &amp;localPoint) const</a>	<b>b2Body</b> [inline]
<a href="#">GetWorldVector(const b2Vec2 &amp;localVector) const</a>	<b>b2Body</b> [inline]
<a href="#">GetXForm() const</a>	<b>b2Body</b> [inline]
<a href="#">IsBullet() const</a>	<b>b2Body</b> [inline]
<a href="#">IsDynamic() const</a>	<b>b2Body</b> [inline]
<a href="#">IsFrozen() const</a>	<b>b2Body</b> [inline]
<a href="#">IsSleeping() const</a>	<b>b2Body</b> [inline]
<a href="#">IsStatic() const</a>	<b>b2Body</b> [inline]
<a href="#">PutToSleep()</a>	<b>b2Body</b> [inline]
<a href="#">SetAngularVelocity(float32 omega)</a>	<b>b2Body</b> [inline]
<a href="#">SetBullet(bool flag)</a>	<b>b2Body</b> [inline]
<a href="#">SetLinearVelocity(const b2Vec2 &amp;v)</a>	<b>b2Body</b> [inline]
<a href="#">SetMass(const b2MassData *massData)</a>	<b>b2Body</b>

<a href="#">SetMassFromShapes()</a>	<b>b2Body</b>
<a href="#">SetUserData(void *data)</a>	<b>b2Body</b> [inline]
<a href="#">SetXForm(const b2Vec2 &amp;position, float32 angle)</a>	<b>b2Body</b>
<a href="#">WakeUp()</a>	<b>b2Body</b> [inline]

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2BodyDef Member List

This is the complete list of members for **b2BodyDef**, including all inherited members.

allowSleep	b2BodyDef
angle	b2BodyDef
angularDamping	b2BodyDef
b2BodyDef()	b2BodyDef [inline]
fixedRotation	b2BodyDef
isBullet	b2BodyDef
isSleeping	b2BodyDef
linearDamping	b2BodyDef
massData	b2BodyDef
position	b2BodyDef
userData	b2BodyDef

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2BoundaryListener Member List

This is the complete list of members for **b2BoundaryListener**, including all inherited members.

**Violation(b2Body \*body)=0** **b2BoundaryListener** [pure virtual]

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2CircleDef Member List

This is the complete list of members for **b2CircleDef**, including all inherited members.

<a href="#">b2ShapeDef()</a>	<a href="#">b2ShapeDef</a>	[inline]
<a href="#">density</a>	<a href="#">b2ShapeDef</a>	
<a href="#">filter</a>	<a href="#">b2ShapeDef</a>	
<a href="#">friction</a>	<a href="#">b2ShapeDef</a>	
<a href="#">isSensor</a>	<a href="#">b2ShapeDef</a>	
<a href="#">restitution</a>	<a href="#">b2ShapeDef</a>	
<a href="#">type</a>	<a href="#">b2ShapeDef</a>	
<a href="#">userData</a>	<a href="#">b2ShapeDef</a>	

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2CircleShape Member List

This is the complete list of members for **b2CircleShape**, including all inherited members.

<b>ComputeAABB</b> (b2AABB *aabb, const b2XForm &transform) const	<b>b2CircleShape</b>	[virtual]
<b>ComputeMass</b> (b2MassData *massData) const	<b>b2CircleShape</b>	[virtual]
<b>ComputeSweptAABB</b> (b2AABB *aabb, const b2XForm &transform1, const b2XForm &transform2) const	<b>b2CircleShape</b>	[virtual]
<b>GetBody()</b>	<b>b2Shape</b>	[inline]
<b>GetFilterData()</b> const	<b>b2Shape</b>	[inline]
<b>GetFriction()</b> const	<b>b2Shape</b>	[inline]
<b>GetLocalPosition()</b> const	<b>b2CircleShape</b>	[inline]
<b>GetNext()</b>	<b>b2Shape</b>	[inline]
<b>GetRadius()</b> const	<b>b2CircleShape</b>	[inline]
<b>GetRestitution()</b> const	<b>b2Shape</b>	[inline]
<b>GetSweepRadius()</b> const	<b>b2Shape</b>	[inline]
<b>GetType()</b> const	<b>b2Shape</b>	[inline]
<b>GetUserData()</b>	<b>b2Shape</b>	[inline]
<b>IsSensor()</b> const	<b>b2Shape</b>	[inline]
<b>SetFilterData</b> (const b2FilterData &filter)	<b>b2Shape</b>	[inline]
<b>SetUserData</b> (void *data)	<b>b2Shape</b>	[inline]
<b>TestPoint</b> (const b2XForm &transform, const b2Vec2 &p) const	<b>b2CircleShape</b>	[virtual]
<b>TestSegment</b> (const b2XForm &transform, float32 *lambda, b2Vec2 *normal, const b2Segment &segment, float32 maxLambda) const	<b>b2CircleShape</b>	[virtual]

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2Color Member List

This is the complete list of members for **b2Color**, including all inherited members.

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Contact Member List

This is the complete list of members for **b2Contact**, including all inherited members.

<b>GetManifoldCount()</b> const	<b>b2Contact</b>	[inline]
<b>GetManifolds()=0</b>	<b>b2Contact</b>	[pure virtual]
<b>GetNext()</b>	<b>b2Contact</b>	[inline]
<b>GetShape1()</b>	<b>b2Contact</b>	[inline]
<b>GetShape2()</b>	<b>b2Contact</b>	[inline]
<b>IsSolid()</b> const	<b>b2Contact</b>	[inline]

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2ContactEdge Member List

This is the complete list of members for **b2ContactEdge**, including all inherited members.

contact	b2ContactEdge
next	b2ContactEdge
other	b2ContactEdge
prev	b2ContactEdge

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2ContactFilter Member List

This is the complete list of members for **b2ContactFilter**, including all inherited members.

**ShouldCollide**(b2Shape \*shape1, b2Shape \*shape2) **b2ContactFilter** [virtual]

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ContactID Member List

This is the complete list of members for **b2ContactID**, including all inherited members.

[key b2ContactID](#)

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ContactID::Features Member List

This is the complete list of members for **b2ContactID::Features**, including all inherited members.

<a href="#">flip</a>	b2ContactID::Features
<a href="#">incidentEdge</a>	b2ContactID::Features
<a href="#">incidentVertex</a>	b2ContactID::Features
<a href="#">referenceEdge</a>	b2ContactID::Features

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2ContactListener Member List

This is the complete list of members for **b2ContactListener**, including all inherited members.

<b>Add</b> (const b2ContactPoint *point)	<b>b2ContactListener</b>	[inline, virtual]
<b>Persist</b> (const b2ContactPoint *point)	<b>b2ContactListener</b>	[inline, virtual]
<b>Remove</b> (const b2ContactPoint *point)	<b>b2ContactListener</b>	[inline, virtual]
<b>Result</b> (const b2ContactResult *point)	<b>b2ContactListener</b>	[inline, virtual]

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ContactPoint Member List

This is the complete list of members for **b2ContactPoint**, including all inherited members.

<a href="#">friction</a>	b2ContactPoint
<a href="#">id</a>	b2ContactPoint
<a href="#">normal</a>	b2ContactPoint
<a href="#">position</a>	b2ContactPoint
<a href="#">restitution</a>	b2ContactPoint
<a href="#">separation</a>	b2ContactPoint
<a href="#">shape1</a>	b2ContactPoint
<a href="#">shape2</a>	b2ContactPoint
<a href="#">velocity</a>	b2ContactPoint

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ContactResult Member List

This is the complete list of members for **b2ContactResult**, including all inherited members.

<b>id</b>	b2ContactResult
<b>normal</b>	b2ContactResult
<b>normalImpulse</b>	b2ContactResult
<b>position</b>	b2ContactResult
<b>shape1</b>	b2ContactResult
<b>shape2</b>	b2ContactResult
<b>tangentImpulse</b>	b2ContactResult

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2DebugDraw Member List

This is the complete list of members for **b2DebugDraw**, including all inherited members.

<b>AppendFlags</b> (uint32 flags)	<b>b2DebugDraw</b>
<b>ClearFlags</b> (uint32 flags)	<b>b2DebugDraw</b>
<b>DrawCircle</b> (const b2Vec2 &center, float32 radius, const b2Color &color)=0	<b>b2DebugDraw</b> [pure virtual]
<b>DrawPolygon</b> (const b2Vec2 *vertices, int32 vertexCount, const b2Color &color)=0	<b>b2DebugDraw</b> [pure virtual]
<b>DrawSegment</b> (const b2Vec2 &p1, const b2Vec2 &p2, const b2Color &color)=0	<b>b2DebugDraw</b> [pure virtual]
<b>DrawSolidCircle</b> (const b2Vec2 &center, float32 radius, const b2Vec2 &axis, const b2Color &color)=0	<b>b2DebugDraw</b> [pure virtual]
<b>DrawSolidPolygon</b> (const b2Vec2 *vertices, int32 vertexCount, const b2Color &color)=0	<b>b2DebugDraw</b> [pure virtual]
<b>DrawXForm</b> (const b2XForm &xf)=0	<b>b2DebugDraw</b> [pure virtual]
<b>e_aabbBit</b> enum value	<b>b2DebugDraw</b>
<b>e_centerOfMassBit</b> enum value	<b>b2DebugDraw</b>
<b>e_coreShapeBit</b> enum value	<b>b2DebugDraw</b>
<b>e_jointBit</b> enum value	<b>b2DebugDraw</b>
<b>e_obbBit</b> enum value	<b>b2DebugDraw</b>
<b>e_pairBit</b> enum value	<b>b2DebugDraw</b>
<b>e_shapeBit</b> enum value	<b>b2DebugDraw</b>
<b>GetFlags()</b> const	<b>b2DebugDraw</b>
<b>SetFlags</b> (uint32 flags)	<b>b2DebugDraw</b>

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2DestructionListener Member List

This is the complete list of members for **b2DestructionListener**, including all inherited members.

<b>SayGoodbye</b> (b2Joint *joint)=0	<b>b2DestructionListener</b> [pure virtual]
<b>SayGoodbye</b> (b2Shape *shape)=0	<b>b2DestructionListener</b> [pure virtual]

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2DistanceJoint Member List

This is the complete list of members for **b2DistanceJoint**, including all inherited members.

<a href="#">GetAnchor1()</a> const	<b>b2DistanceJoint</b>	[virtual]
<a href="#">GetAnchor2()</a> const	<b>b2DistanceJoint</b>	[virtual]
<a href="#">GetBody1()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetBody2()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetNext()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetReactionForce()</a> const	<b>b2DistanceJoint</b>	[virtual]
<a href="#">GetReactionTorque()</a> const	<b>b2DistanceJoint</b>	[virtual]
<a href="#">GetType()</a> const	<b>b2Joint</b>	[inline]
<a href="#">GetUserData()</a>	<b>b2Joint</b>	[inline]
<a href="#">SetUserData(void *data)</a>	<b>b2Joint</b>	[inline]

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2DistanceJointDef Member List

This is the complete list of members for **b2DistanceJointDef**, including all inherited members.

<a href="#">body1</a>	<a href="#">b2JointDef</a>
<a href="#">body2</a>	<a href="#">b2JointDef</a>
<a href="#">collideConnected</a>	<a href="#">b2JointDef</a>
<a href="#">dampingRatio</a>	<a href="#">b2DistanceJointDef</a>
<a href="#">frequencyHz</a>	<a href="#">b2DistanceJointDef</a>
<a href="#">Initialize(b2Body *body1, b2Body *body2, const b2Vec2 &amp;anchor1, const b2Vec2 &amp;anchor2)</a>	<a href="#">b2DistanceJointDef</a>
<a href="#">length</a>	<a href="#">b2DistanceJointDef</a>
<a href="#">localAnchor1</a>	<a href="#">b2DistanceJointDef</a>
<a href="#">localAnchor2</a>	<a href="#">b2DistanceJointDef</a>
<a href="#">type</a>	<a href="#">b2JointDef</a>
<a href="#">userData</a>	<a href="#">b2JointDef</a>

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2FilterData Member List

This is the complete list of members for **b2FilterData**, including all inherited members.

categoryBits	b2FilterData
groupIndex	b2FilterData
maskBits	b2FilterData

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2GearJoint Member List

This is the complete list of members for **b2GearJoint**, including all inherited members.

<a href="#">GetAnchor1()</a> const	<b>b2GearJoint</b>	[virtual]
<a href="#">GetAnchor2()</a> const	<b>b2GearJoint</b>	[virtual]
<a href="#">GetBody1()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetBody2()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetNext()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetRatio()</a> const	<b>b2GearJoint</b>	
<a href="#">GetReactionForce()</a> const	<b>b2GearJoint</b>	[virtual]
<a href="#">GetReactionTorque()</a> const	<b>b2GearJoint</b>	[virtual]
<a href="#">GetType()</a> const	<b>b2Joint</b>	[inline]
<a href="#">GetUserData()</a>	<b>b2Joint</b>	[inline]
<a href="#">SetUserData(void *data)</a>	<b>b2Joint</b>	[inline]

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2GearJointDef Member List

This is the complete list of members for **b2GearJointDef**, including all inherited members.

<b>body1</b>	b2JointDef
<b>body2</b>	b2JointDef
<b>collideConnected</b>	b2JointDef
<b>joint1</b>	b2GearJointDef
<b>joint2</b>	b2GearJointDef
<b>ratio</b>	b2GearJointDef
<b>type</b>	b2JointDef
<b>userData</b>	b2JointDef

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Joint Member List

This is the complete list of members for **b2Joint**, including all inherited members.

<a href="#">GetAnchor1()</a> const =0	<b>b2Joint</b> [pure virtual]
<a href="#">GetAnchor2()</a> const =0	<b>b2Joint</b> [pure virtual]
<a href="#">GetBody1()</a>	<b>b2Joint</b> [inline]
<a href="#">GetBody2()</a>	<b>b2Joint</b> [inline]
<a href="#">GetNext()</a>	<b>b2Joint</b> [inline]
<a href="#">GetReactionForce()</a> const =0	<b>b2Joint</b> [pure virtual]
<a href="#">GetReactionTorque()</a> const =0	<b>b2Joint</b> [pure virtual]
<a href="#">GetType()</a> const	<b>b2Joint</b> [inline]
<a href="#">GetUserData()</a>	<b>b2Joint</b> [inline]
<a href="#">SetUserData(void *data)</a>	<b>b2Joint</b> [inline]

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2JointDef Member List

This is the complete list of members for **b2JointDef**, including all inherited members.

<b>body1</b>	b2JointDef
<b>body2</b>	b2JointDef
<b>collideConnected</b>	b2JointDef
<b>type</b>	b2JointDef
<b>userData</b>	b2JointDef

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2JointEdge Member List

This is the complete list of members for **b2JointEdge**, including all inherited members.

<code>joint</code>	<code>b2JointEdge</code>
<code>next</code>	<code>b2JointEdge</code>
<code>other</code>	<code>b2JointEdge</code>
<code>prev</code>	<code>b2JointEdge</code>

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2Manifold Member List

This is the complete list of members for **b2Manifold**, including all inherited members.

normal	b2Manifold
pointCount	b2Manifold
points	b2Manifold

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2ManifoldPoint Member List

This is the complete list of members for **b2ManifoldPoint**, including all inherited members.

<b>id</b>	<a href="#">b2ManifoldPoint</a>
<b>localPoint1</b>	<a href="#">b2ManifoldPoint</a>
<b>localPoint2</b>	<a href="#">b2ManifoldPoint</a>
<b>normalImpulse</b>	<a href="#">b2ManifoldPoint</a>
<b>separation</b>	<a href="#">b2ManifoldPoint</a>
<b>tangentImpulse</b>	<a href="#">b2ManifoldPoint</a>

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2MassData Member List

This is the complete list of members for **b2MassData**, including all inherited members.

center	b2MassData
I	b2MassData
mass	b2MassData

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Mat22 Member List

This is the complete list of members for **b2Mat22**, including all inherited members.

<a href="#">b2Mat22()</a>	<b>b2Mat22</b> [inline]
<a href="#">b2Mat22(const b2Vec2 &amp;c1, const b2Vec2 &amp;c2)</a>	<b>b2Mat22</b> [inline]
<a href="#">b2Mat22(float32 a11, float32 a12, float32 a21, float32 a22)</a>	<b>b2Mat22</b> [inline]
<a href="#">b2Mat22(float32 angle)</a>	<b>b2Mat22</b> [inline, explicit]
<a href="#">GetAngle() const</a>	<b>b2Mat22</b> [inline]
<a href="#">Set(const b2Vec2 &amp;c1, const b2Vec2 &amp;c2)</a>	<b>b2Mat22</b> [inline]
<a href="#">Set(float32 angle)</a>	<b>b2Mat22</b> [inline]
<a href="#">SetIdentity()</a>	<b>b2Mat22</b> [inline]
<a href="#">SetZero()</a>	<b>b2Mat22</b> [inline]
<a href="#">Solve(const b2Vec2 &amp;b) const</a>	<b>b2Mat22</b> [inline]

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2MouseJoint Member List

This is the complete list of members for **b2MouseJoint**, including all inherited members.

<code>GetAnchor1()</code> const	<b>b2MouseJoint</b>	[virtual]
<code>GetAnchor2()</code> const	<b>b2MouseJoint</b>	[virtual]
<code>GetBody1()</code>	<b>b2Joint</b>	[inline]
<code>GetBody2()</code>	<b>b2Joint</b>	[inline]
<code>GetNext()</code>	<b>b2Joint</b>	[inline]
<code>GetReactionForce()</code> const	<b>b2MouseJoint</b>	[virtual]
<code>GetReactionTorque()</code> const	<b>b2MouseJoint</b>	[virtual]
<code>GetType()</code> const	<b>b2Joint</b>	[inline]
<code>GetUserData()</code>	<b>b2Joint</b>	[inline]
<code>SetTarget(const b2Vec2 &amp;target)</code>	<b>b2MouseJoint</b>	
<code>SetUserData(void *data)</code>	<b>b2Joint</b>	[inline]

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2MouseJointDef Member List

This is the complete list of members for **b2MouseJointDef**, including all inherited members.

<b>body1</b>	b2JointDef
<b>body2</b>	b2JointDef
<b>collideConnected</b>	b2JointDef
<b>dampingRatio</b>	b2MouseJointDef
<b>frequencyHz</b>	b2MouseJointDef
<b>maxForce</b>	b2MouseJointDef
<b>target</b>	b2MouseJointDef
<b>timeStep</b>	b2MouseJointDef
<b>type</b>	b2JointDef
<b>userData</b>	b2JointDef

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2OBB Member List

This is the complete list of members for **b2OBB**, including all inherited members.

center	b2OBB
extents	b2OBB
R	b2OBB

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2PolygonDef Member List

This is the complete list of members for **b2PolygonDef**, including all inherited members.

<a href="#">b2ShapeDef()</a>	<a href="#">b2ShapeDef</a>	[inline]
<a href="#">density</a>	<a href="#">b2ShapeDef</a>	
<a href="#">filter</a>	<a href="#">b2ShapeDef</a>	
<a href="#">friction</a>	<a href="#">b2ShapeDef</a>	
<a href="#">isSensor</a>	<a href="#">b2ShapeDef</a>	
<a href="#">restitution</a>	<a href="#">b2ShapeDef</a>	
<a href="#">SetAsBox(float32 hx, float32 hy)</a>	<a href="#">b2PolygonDef</a>	
<a href="#">SetAsBox(float32 hx, float32 hy, const b2Vec2 &amp;center, float32 angle)</a>	<a href="#">b2PolygonDef</a>	
<a href="#">type</a>	<a href="#">b2ShapeDef</a>	
<a href="#">userData</a>	<a href="#">b2ShapeDef</a>	
<a href="#">vertexCount</a>	<a href="#">b2PolygonDef</a>	
<a href="#">vertices</a>	<a href="#">b2PolygonDef</a>	

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2PolygonShape Member List

This is the complete list of members for **b2PolygonShape**, including all inherited members.

<b>Centroid</b> (const b2XForm &xf) const	<b>b2PolygonShape</b>	
<b>ComputeAABB</b> (b2AABB *aabb, const b2XForm &transform) const	<b>b2PolygonShape</b>	[virtual]
<b>ComputeMass</b> (b2MassData *massData) const	<b>b2PolygonShape</b>	[virtual]
<b>ComputeSweptAABB</b> (b2AABB *aabb, const b2XForm &transform1, const b2XForm &transform2) const	<b>b2PolygonShape</b>	[virtual]
<b>GetBody()</b>	<b>b2Shape</b>	[inline]
<b>GetCentroid()</b> const	<b>b2PolygonShape</b>	[inline]
<b>GetCoreVertices()</b> const	<b>b2PolygonShape</b>	[inline]
<b>GetFilterData()</b> const	<b>b2Shape</b>	[inline]
<b>GetFirstVertex</b> (const b2XForm &xf) const	<b>b2PolygonShape</b>	[inline]
<b>GetFriction()</b> const	<b>b2Shape</b>	[inline]
<b>GetNext()</b>	<b>b2Shape</b>	[inline]
<b>GetNormals()</b> const	<b>b2PolygonShape</b>	[inline]
<b>GetOBB()</b> const	<b>b2PolygonShape</b>	[inline]
<b>GetRestitution()</b> const	<b>b2Shape</b>	[inline]
<b>GetSweepRadius()</b> const	<b>b2Shape</b>	[inline]
<b>GetType()</b> const	<b>b2Shape</b>	[inline]
<b>GetUserData()</b>	<b>b2Shape</b>	[inline]
<b>GetVertexCount()</b> const	<b>b2PolygonShape</b>	[inline]
<b>GetVertices()</b> const	<b>b2PolygonShape</b>	[inline]
<b>IsSensor()</b> const	<b>b2Shape</b>	[inline]
<b>SetFilterData</b> (const b2FilterData &filter)	<b>b2Shape</b>	[inline]
<b>SetUserData</b> (void *data)	<b>b2Shape</b>	[inline]
<b>Support</b> (const b2XForm &xf, const b2Vec2 &d) const	<b>b2PolygonShape</b>	
<b>TestPoint</b> (const b2XForm &transform, const b2Vec2 &p) const	<b>b2PolygonShape</b>	[virtual]
<b>TestSegment</b> (const b2XForm &transform, float32 *lambda, b2Vec2 *normal, const b2Segment &segment, float32 maxLambda) const	<b>b2PolygonShape</b>	[virtual]

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2PrismaticJoint Member List

This is the complete list of members for **b2PrismaticJoint**, including all inherited members.

<b>EnableLimit</b> (bool flag)	<b>b2PrismaticJoint</b>	
<b>EnableMotor</b> (bool flag)	<b>b2PrismaticJoint</b>	
<b>GetAnchor1</b> () const	<b>b2PrismaticJoint</b>	[virtual]
<b>GetAnchor2</b> () const	<b>b2PrismaticJoint</b>	[virtual]
<b>GetBody1</b> ()	<b>b2Joint</b>	[inline]
<b>GetBody2</b> ()	<b>b2Joint</b>	[inline]
<b>GetJointSpeed</b> () const	<b>b2PrismaticJoint</b>	
<b>GetJointTranslation</b> () const	<b>b2PrismaticJoint</b>	
<b>GetLowerLimit</b> () const	<b>b2PrismaticJoint</b>	
<b>GetMotorForce</b> () const	<b>b2PrismaticJoint</b>	
<b>GetMotorSpeed</b> () const	<b>b2PrismaticJoint</b>	[inline]
<b>GetNext</b> ()	<b>b2Joint</b>	[inline]
<b>GetReactionForce</b> () const	<b>b2PrismaticJoint</b>	[virtual]
<b>GetReactionTorque</b> () const	<b>b2PrismaticJoint</b>	[virtual]
<b>GetType</b> () const	<b>b2Joint</b>	[inline]
<b>GetUpperLimit</b> () const	<b>b2PrismaticJoint</b>	
<b>GetUserData</b> ()	<b>b2Joint</b>	[inline]
<b>IsLimitEnabled</b> () const	<b>b2PrismaticJoint</b>	
<b>IsMotorEnabled</b> () const	<b>b2PrismaticJoint</b>	
<b>SetLimits</b> (float32 lower, float32 upper)	<b>b2PrismaticJoint</b>	
<b>SetMaxMotorForce</b> (float32 force)	<b>b2PrismaticJoint</b>	
<b>SetMotorSpeed</b> (float32 speed)	<b>b2PrismaticJoint</b>	
<b>SetUserData</b> (void *data)	<b>b2Joint</b>	[inline]

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2PrismaticJointDef Member List

This is the complete list of members for **b2PrismaticJointDef**, including all inherited members.

<code>body1</code>	<code>b2JointDef</code>
<code>body2</code>	<code>b2JointDef</code>
<code>collideConnected</code>	<code>b2JointDef</code>
<code>enableLimit</code>	<code>b2PrismaticJointDef</code>
<code>enableMotor</code>	<code>b2PrismaticJointDef</code>
<code>Initialize(b2Body *body1, b2Body *body2, const b2Vec2 &amp;anchor, const b2Vec2 &amp;axis)</code>	<code>b2PrismaticJointDef</code>
<code>localAnchor1</code>	<code>b2PrismaticJointDef</code>
<code>localAnchor2</code>	<code>b2PrismaticJointDef</code>
<code>localAxis1</code>	<code>b2PrismaticJointDef</code>
<code>lowerTranslation</code>	<code>b2PrismaticJointDef</code>
<code>maxMotorForce</code>	<code>b2PrismaticJointDef</code>
<code>motorSpeed</code>	<code>b2PrismaticJointDef</code>
<code>referenceAngle</code>	<code>b2PrismaticJointDef</code>
<code>type</code>	<code>b2JointDef</code>
<code>upperTranslation</code>	<code>b2PrismaticJointDef</code>
<code>userData</code>	<code>b2JointDef</code>

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2PulleyJoint Member List

This is the complete list of members for **b2PulleyJoint**, including all inherited members.

<a href="#">GetAnchor1()</a> const	<b>b2PulleyJoint</b>	[virtual]
<a href="#">GetAnchor2()</a> const	<b>b2PulleyJoint</b>	[virtual]
<a href="#">GetBody1()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetBody2()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetGroundAnchor1()</a> const	<b>b2PulleyJoint</b>	
<a href="#">GetGroundAnchor2()</a> const	<b>b2PulleyJoint</b>	
<a href="#">GetLength1()</a> const	<b>b2PulleyJoint</b>	
<a href="#">GetLength2()</a> const	<b>b2PulleyJoint</b>	
<a href="#">GetNext()</a>	<b>b2Joint</b>	[inline]
<a href="#">GetRatio()</a> const	<b>b2PulleyJoint</b>	
<a href="#">GetReactionForce()</a> const	<b>b2PulleyJoint</b>	[virtual]
<a href="#">GetReactionTorque()</a> const	<b>b2PulleyJoint</b>	[virtual]
<a href="#">GetType()</a> const	<b>b2Joint</b>	[inline]
<a href="#">GetUserData()</a>	<b>b2Joint</b>	[inline]
<a href="#">SetUserData(void *data)</a>	<b>b2Joint</b>	[inline]

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2PulleyJointDef Member List

This is the complete list of members for **b2PulleyJointDef**, including all inherited members.

<a href="#">body1</a>	<a href="#">b2JointDef</a>
<a href="#">body2</a>	<a href="#">b2JointDef</a>
<a href="#">collideConnected</a>	<a href="#">b2JointDef</a>
<a href="#">groundAnchor1</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">groundAnchor2</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">Initialize(b2Body *body1, b2Body *body2, const b2Vec2 &amp;groundAnchor1, const b2Vec2 &amp;groundAnchor2, const b2Vec2 &amp;anchor1, const b2Vec2 &amp;anchor2, float32 ratio)</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">length1</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">length2</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">localAnchor1</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">localAnchor2</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">maxLength1</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">maxLength2</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">ratio</a>	<a href="#">b2PulleyJointDef</a>
<a href="#">type</a>	<a href="#">b2JointDef</a>
<a href="#">userData</a>	<a href="#">b2JointDef</a>

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

# b2RevoluteJoint Member List

This is the complete list of members for **b2RevoluteJoint**, including all inherited members.

<b>EnableLimit</b> (bool flag)	<b>b2RevoluteJoint</b>
<b>EnableMotor</b> (bool flag)	<b>b2RevoluteJoint</b>
<b>GetAnchor1</b> () const	<b>b2RevoluteJoint</b> [virtual]
<b>GetAnchor2</b> () const	<b>b2RevoluteJoint</b> [virtual]
<b>GetBody1</b> ()	<b>b2Joint</b> [inline]
<b>GetBody2</b> ()	<b>b2Joint</b> [inline]
<b>GetJointAngle</b> () const	<b>b2RevoluteJoint</b>
<b>GetJointSpeed</b> () const	<b>b2RevoluteJoint</b>
<b>GetLowerLimit</b> () const	<b>b2RevoluteJoint</b>
<b>GetMotorSpeed</b> () const	<b>b2RevoluteJoint</b> [inline]
<b>GetMotorTorque</b> () const	<b>b2RevoluteJoint</b>
<b>GetNext</b> ()	<b>b2Joint</b> [inline]
<b>GetReactionForce</b> () const	<b>b2RevoluteJoint</b> [virtual]
<b>GetReactionTorque</b> () const	<b>b2RevoluteJoint</b> [virtual]
<b>GetType</b> () const	<b>b2Joint</b> [inline]
<b>GetUpperLimit</b> () const	<b>b2RevoluteJoint</b>
<b>GetUserData</b> ()	<b>b2Joint</b> [inline]
<b>IsLimitEnabled</b> () const	<b>b2RevoluteJoint</b>
<b>IsMotorEnabled</b> () const	<b>b2RevoluteJoint</b>
<b>SetLimits</b> (float32 lower, float32 upper)	<b>b2RevoluteJoint</b>
<b>SetMaxMotorTorque</b> (float32 torque)	<b>b2RevoluteJoint</b>
<b>SetMotorSpeed</b> (float32 speed)	<b>b2RevoluteJoint</b>
<b>SetUserData</b> (void *data)	<b>b2Joint</b> [inline]

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2RevoluteJointDef Member List

This is the complete list of members for **b2RevoluteJointDef**, including all inherited members.

body1	b2JointDef
body2	b2JointDef
collideConnected	b2JointDef
enableLimit	b2RevoluteJointDef
enableMotor	b2RevoluteJointDef
Initialize(b2Body *body1, b2Body *body2, const b2Vec2 &anchor)	b2RevoluteJointDef
localAnchor1	b2RevoluteJointDef
localAnchor2	b2RevoluteJointDef
lowerAngle	b2RevoluteJointDef
maxMotorTorque	b2RevoluteJointDef
motorSpeed	b2RevoluteJointDef
referenceAngle	b2RevoluteJointDef
type	b2JointDef
upperAngle	b2RevoluteJointDef
userData	b2JointDef

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Segment Member List

This is the complete list of members for **b2Segment**, including all inherited members.

p1	b2Segment
p2	b2Segment
TestSegment(float32 *lambda, b2Vec2 *normal, const b2Segment &segment, float32 maxLambda) const	b2Segment

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2Shape Member List

This is the complete list of members for **b2Shape**, including all inherited members.

<code>ComputeAABB(b2AABB *aabb, const b2XForm &amp;xf) const =0</code>	<b>b2Shape</b>	[pure virtual]
<code>ComputeMass(b2MassData *massData) const =0</code>	<b>b2Shape</b>	[pure virtual]
<code>ComputeSweptAABB(b2AABB *aabb, const b2XForm &amp;xf1, const b2XForm &amp;xf2) const =0</code>	<b>b2Shape</b>	[pure virtual]
<code>GetBody()</code>	<b>b2Shape</b>	[inline]
<code>GetFilterData() const</code>	<b>b2Shape</b>	[inline]
<code>GetFriction() const</code>	<b>b2Shape</b>	[inline]
<code>GetNext()</code>	<b>b2Shape</b>	[inline]
<code>GetRestitution() const</code>	<b>b2Shape</b>	[inline]
<code>GetSweepRadius() const</code>	<b>b2Shape</b>	[inline]
<code>GetType() const</code>	<b>b2Shape</b>	[inline]
<code>GetUserData()</code>	<b>b2Shape</b>	[inline]
<code>IsSensor() const</code>	<b>b2Shape</b>	[inline]
<code>SetFilterData(const b2FilterData &amp;filter)</code>	<b>b2Shape</b>	[inline]
<code>SetUserData(void *data)</code>	<b>b2Shape</b>	[inline]
<code>TestPoint(const b2XForm &amp;xf, const b2Vec2 &amp;p) const =0</code>	<b>b2Shape</b>	[pure virtual]
<code>TestSegment(const b2XForm &amp;xf, float32 *lambda, b2Vec2 *normal, const b2Segment &amp;segment, float32 maxLambda) const =0</code>	<b>b2Shape</b>	[pure virtual]

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2ShapeDef Member List

This is the complete list of members for **b2ShapeDef**, including all inherited members.

<a href="#">b2ShapeDef()</a>	<a href="#">b2ShapeDef</a> [inline]
<a href="#">density</a>	<a href="#">b2ShapeDef</a>
<a href="#">filter</a>	<a href="#">b2ShapeDef</a>
<a href="#">friction</a>	<a href="#">b2ShapeDef</a>
<a href="#">isSensor</a>	<a href="#">b2ShapeDef</a>
<a href="#">restitution</a>	<a href="#">b2ShapeDef</a>
<a href="#">type</a>	<a href="#">b2ShapeDef</a>
<a href="#">userData</a>	<a href="#">b2ShapeDef</a>

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#) [Classes](#) [Files](#)

---

[Alphabetical List](#) [Class List](#) [Class Hierarchy](#) [Class Members](#)

---

## b2Sweep Member List

This is the complete list of members for **b2Sweep**, including all inherited members.

a	b2Sweep
Advance(float32 t)	b2Sweep
c	b2Sweep
GetXForm(b2XForm *xf, float32 t) const	b2Sweep
localCenter	b2Sweep
t0	b2Sweep

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Vec2 Member List

This is the complete list of members for **b2Vec2**, including all inherited members.

<a href="#">b2Vec2()</a>	<b>b2Vec2</b> [inline]
<a href="#">b2Vec2(float32 x, float32 y)</a>	<b>b2Vec2</b> [inline]
<a href="#">IsValid() const</a>	<b>b2Vec2</b> [inline]
<a href="#">Length() const</a>	<b>b2Vec2</b> [inline]
<a href="#">LengthSquared() const</a>	<b>b2Vec2</b> [inline]
<a href="#">Normalize()</a>	<b>b2Vec2</b> [inline]
<a href="#">operator *=(float32 a)</a>	<b>b2Vec2</b> [inline]
<a href="#">operator+=(const b2Vec2 &amp;v)</a>	<b>b2Vec2</b> [inline]
<a href="#">operator-() const</a>	<b>b2Vec2</b> [inline]
<a href="#">operator-=(const b2Vec2 &amp;v)</a>	<b>b2Vec2</b> [inline]
<a href="#">Set(float32 x_, float32 y_)</a>	<b>b2Vec2</b> [inline]
<a href="#">SetZero()</a>	<b>b2Vec2</b> [inline]

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2Version Member List

This is the complete list of members for **b2Version**, including all inherited members.

major	b2Version
minor	b2Version
revision	b2Version

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

# b2World Member List

This is the complete list of members for **b2World**, including all inherited members.

<a href="#">b2World</a> (const b2AABB &worldAABB, const b2Vec2 &gravity, bool doSleep)	<b>b2World</b>
<a href="#">CreateBody</a> (const b2BodyDef *def)	<b>b2World</b>
<a href="#">CreateJoint</a> (const b2JointDef *def)	<b>b2World</b>
<a href="#">DestroyBody</a> (b2Body *body)	<b>b2World</b>
<a href="#">DestroyJoint</a> (b2Joint *joint)	<b>b2World</b>
<a href="#">GetBodyCount</a> () const	<b>b2World</b> [inline]
<a href="#">GetBodyList</a> ()	<b>b2World</b> [inline]
<a href="#">GetContactCount</a> () const	<b>b2World</b> [inline]
<a href="#">GetGroundBody</a> ()	<b>b2World</b> [inline]
<a href="#">GetJointCount</a> () const	<b>b2World</b> [inline]
<a href="#">GetJointList</a> ()	<b>b2World</b> [inline]
<a href="#">GetPairCount</a> () const	<b>b2World</b>
<a href="#">GetProxyCount</a> () const	<b>b2World</b>
<a href="#">Query</a> (const b2AABB &aabb, b2Shape **shapes, int32 maxCount)	<b>b2World</b>
<a href="#">Refilter</a> (b2Shape *shape)	<b>b2World</b>
<a href="#">SetBoundaryListener</a> (b2BoundaryListener *listener)	<b>b2World</b>
<a href="#">SetContactFilter</a> (b2ContactFilter *filter)	<b>b2World</b>
<a href="#">SetContactListener</a> (b2ContactListener *listener)	<b>b2World</b>
<a href="#">SetContinuousPhysics</a> (bool flag)	<b>b2World</b> [inline]
<a href="#">SetDebugDraw</a> (b2DebugDraw *debugDraw)	<b>b2World</b>
<a href="#">SetDestructionListener</a> (b2DestructionListener *listener)	<b>b2World</b>
<a href="#">SetGravity</a> (const b2Vec2 &gravity)	<b>b2World</b> [inline]
<a href="#">SetPositionCorrection</a> (bool flag)	<b>b2World</b> [inline]
<a href="#">SetWarmStarting</a> (bool flag)	<b>b2World</b> [inline]
<a href="#">Step</a> (float32 timeStep, int32 iterations)	<b>b2World</b>
<a href="#">Validate</a> ()	<b>b2World</b>
<a href="#">~b2World</a> ()	<b>b2World</b>

[Main Page](#)[Classes](#)[Files](#)[Alphabetical List](#)[Class List](#)[Class Hierarchy](#)[Class Members](#)

## b2XForm Member List

This is the complete list of members for **b2XForm**, including all inherited members.

**b2XForm()**

**b2XForm** [inline]

**b2XForm**(const b2Vec2 &position, const b2Mat22 &R)

**b2XForm** [inline]

**SetIdentity()**

**b2XForm** [inline]

---

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*



## - a -

- Add() : **b2ContactListener**
- Advance() : **b2Sweep**
- AllowSleeping() : **b2Body**
- AppendFlags() : **b2DebugDraw**
- ApplyForce() : **b2Body**
- ApplyImpulse() : **b2Body**
- ApplyTorque() : **b2Body**



## - a -

- a : **b2Sweep**
- allowSleep : **b2BodyDef**
- angle : **b2BodyDef**
- angularDamping : **b2BodyDef**

## - b -

- body1 : **b2JointDef**
- body2 : **b2JointDef**

## - c -

- c : **b2Sweep**
- categoryBits : **b2FilterData**
- center : **b2MassData** , **b2OBB**
- collideConnected : **b2JointDef**
- contact : **b2ContactEdge**

## - d -

- dampingRatio : **b2DistanceJointDef** , **b2MouseJointDef**
- density : **b2ShapeDef**

## - e -

- enableLimit : **b2PrismaticJointDef** , **b2RevoluteJointDef**
- enableMotor : **b2RevoluteJointDef** , **b2PrismaticJointDef**
- extents : **b2OBB**

## - f -

- filter : **b2ShapeDef**
- fixedRotation : **b2BodyDef**
- flip : **b2ContactID::Features**
- frequencyHz : **b2MouseJointDef** , **b2DistanceJointDef**
- friction : **b2ShapeDef** , **b2ContactPoint**

- g -

- groundAnchor1 : **b2PulleyJointDef**
- groundAnchor2 : **b2PulleyJointDef**
- groupIndex : **b2FilterData**

- i -

- I : **b2MassData**
- id : **b2ManifoldPoint** , **b2ContactResult** , **b2ContactPoint**
- incidentEdge : **b2ContactID::Features**
- incidentVertex : **b2ContactID::Features**
- isBullet : **b2BodyDef**
- isSensor : **b2ShapeDef**
- isSleeping : **b2BodyDef**

- j -

- joint : **b2JointEdge**
- joint1 : **b2GearJointDef**
- joint2 : **b2GearJointDef**

- k -

- key : **b2ContactID**

- l -

- length : **b2DistanceJointDef**
- length1 : **b2PulleyJointDef**
- length2 : **b2PulleyJointDef**
- linearDamping : **b2BodyDef**
- localAnchor1 : **b2RevoluteJointDef** , **b2DistanceJointDef** ,  
**b2PrismaticJointDef** , **b2PulleyJointDef**

- localAnchor2 : **b2DistanceJointDef** , **b2PrismaticJointDef** , **b2PulleyJointDef** , **b2RevoluteJointDef**
- localAxis1 : **b2PrismaticJointDef**
- localCenter : **b2Sweep**
- localPoint1 : **b2ManifoldPoint**
- localPoint2 : **b2ManifoldPoint**
- lowerAngle : **b2RevoluteJointDef**
- lowerBound : **b2AABB**
- lowerTranslation : **b2PrismaticJointDef**

- m -

- major : **b2Version**
- maskBits : **b2FilterData**
- mass : **b2MassData**
- massData : **b2BodyDef**
- maxForce : **b2MouseJointDef**
- maxLength1 : **b2PulleyJointDef**
- maxLength2 : **b2PulleyJointDef**
- maxMotorForce : **b2PrismaticJointDef**
- maxMotorTorque : **b2RevoluteJointDef**
- minor : **b2Version**
- motorSpeed : **b2RevoluteJointDef** , **b2PrismaticJointDef**

- n -

- next : **b2ContactEdge** , **b2JointEdge**
- normal : **b2ContactPoint** , **b2ContactResult** , **b2Manifold**
- normalImpulse : **b2ContactResult** , **b2ManifoldPoint**

- o -

- other : **b2ContactEdge** , **b2JointEdge**

- p -

- p1 : **b2Segment**
- p2 : **b2Segment**
- pointCount : **b2Manifold**
- points : **b2Manifold**

- position : **b2ContactResult** , **b2BodyDef** , **b2ContactPoint**
- prev : **b2JointEdge** , **b2ContactEdge**

- r -

- R : **b2OBB**
- ratio : **b2GearJointDef** , **b2PulleyJointDef**
- referenceAngle : **b2PrismaticJointDef** , **b2RevoluteJointDef**
- referenceEdge : **b2ContactID::Features**
- restitution : **b2ContactPoint** , **b2ShapeDef**
- revision : **b2Version**

- s -

- separation : **b2ManifoldPoint** , **b2ContactPoint**
- shape1 : **b2ContactResult** , **b2ContactPoint**
- shape2 : **b2ContactResult** , **b2ContactPoint**

- t -

- t0 : **b2Sweep**
- tangentImpulse : **b2ManifoldPoint** , **b2ContactResult**
- target : **b2MouseJointDef**
- timeStep : **b2MouseJointDef**
- type : **b2ShapeDef** , **b2JointDef**

- u -

- upperAngle : **b2RevoluteJointDef**
- upperBound : **b2AABB**
- upperTranslation : **b2PrismaticJointDef**
- userData : **b2ShapeDef** , **b2BodyDef** , **b2JointDef**

- v -

- velocity : **b2ContactPoint**
  - vertexCount : **b2PolygonDef**
  - vertices : **b2PolygonDef**
-

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

[Main Page](#)

[Classes](#)

[Files](#)

[Alphabetical List](#)

[Class List](#)

[Class Hierarchy](#)

[Class Members](#)

[All](#)

[Functions](#)

[Variables](#)

[Enumerator](#)

- e\_aabbBit : [b2DebugDraw](#)
- e\_centerOfMassBit : [b2DebugDraw](#)
- e\_coreShapeBit : [b2DebugDraw](#)
- e\_jointBit : [b2DebugDraw](#)
- e\_obbBit : [b2DebugDraw](#)
- e\_pairBit : [b2DebugDraw](#)
- e\_shapeBit : [b2DebugDraw](#)

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- **b** -

- b2BodyDef() : [b2BodyDef](#)
- body1 : [b2JointDef](#)
- body2 : [b2JointDef](#)

The screenshot shows a top-level navigation bar with several tabs: Main Page, Classes, Files, Alphabetical List, Class List, Class Hierarchy, Class Members, All, Functions, Variables, Enumerator, and a scrollable letter navigation bar containing 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', and '~'. The 'Class Members' tab is currently selected.

Here is a list of all documented class members with links to the class documentation for each member:

- C -

- c : [b2Sweep](#)
- categoryBits : [b2FilterData](#)
- center : [b2MassData](#) , [b2OBB](#)
- Centroid() : [b2PolygonShape](#)
- ClearFlags() : [b2DebugDraw](#)
- collideConnected : [b2JointDef](#)
- ComputeAABB() : [b2CircleShape](#) , [b2PolygonShape](#) , [b2Shape](#)
- ComputeMass() : [b2CircleShape](#) , [b2PolygonShape](#) , [b2Shape](#)
- ComputeSweptAABB() : [b2Shape](#) , [b2PolygonShape](#) , [b2CircleShape](#)
- contact : [b2ContactEdge](#)
- CreateBody() : [b2World](#)
- CreateJoint() : [b2World](#)
- CreateShape() : [b2Body](#)

The screenshot shows a top-level navigation bar with several tabs: 'Main Page', 'Classes', 'Files', 'Alphabetical List', 'Class List', 'Class Hierarchy', and 'Class Members'. Below this is a secondary set of tabs: 'All', 'Functions', 'Variables', and 'Enumerator'. At the bottom is a horizontal bar containing letters from 'a' to 'x' and a tilde (~).

Here is a list of all documented class members with links to the class documentation for each member:

- d -

- dampingRatio : [b2DistanceJointDef](#) , [b2MouseJointDef](#)
- density : [b2ShapeDef](#)
- DestroyBody() : [b2World](#)
- DestroyJoint() : [b2World](#)
- DestroyShape() : [b2Body](#)
- DrawCircle() : [b2DebugDraw](#)
- DrawPolygon() : [b2DebugDraw](#)
- DrawSegment() : [b2DebugDraw](#)
- DrawSolidCircle() : [b2DebugDraw](#)
- DrawSolidPolygon() : [b2DebugDraw](#)
- DrawXForm() : [b2DebugDraw](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- e -

- e\_aabbBit : [b2DebugDraw](#)
- e\_centerOfMassBit : [b2DebugDraw](#)
- e\_coreShapeBit : [b2DebugDraw](#)
- e\_jointBit : [b2DebugDraw](#)
- e\_obbBit : [b2DebugDraw](#)
- e\_pairBit : [b2DebugDraw](#)
- e\_shapeBit : [b2DebugDraw](#)
- enableLimit : [b2PrismaticJointDef](#) , [b2RevoluteJointDef](#)
- EnableLimit() : [b2PrismaticJoint](#) , [b2RevoluteJoint](#)
- EnableMotor() : [b2RevoluteJoint](#)
- enableMotor : [b2RevoluteJointDef](#)
- EnableMotor() : [b2PrismaticJoint](#)
- enableMotor : [b2PrismaticJointDef](#)
- extents : [b2OBB](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- f -

- filter : [b2ShapeDef](#)
- fixedRotation : [b2BodyDef](#)
- flip : [b2ContactID::Features](#)
- frequencyHz : [b2MouseJointDef](#) , [b2DistanceJointDef](#)
- friction : [b2ShapeDef](#) , [b2ContactPoint](#)



Here is a list of all documented class members with links to the class documentation for each member:

## - g -

- GetAnchor1() : [b2DistanceJoint](#) , [b2GearJoint](#) , [b2MouseJoint](#) , [b2PrismaticJoint](#) , [b2Joint](#) , [b2PulleyJoint](#) , [b2RevoluteJoint](#)
- GetAnchor2() : [b2DistanceJoint](#) , [b2GearJoint](#) , [b2Joint](#) , [b2MouseJoint](#) , [b2PrismaticJoint](#) , [b2PulleyJoint](#) , [b2RevoluteJoint](#)
- GetAngle() : [b2Body](#) , [b2Mat22](#)
- GetAngularVelocity() : [b2Body](#)
- GetBody() : [b2Shape](#)
- GetBody1() : [b2Joint](#)
- GetBody2() : [b2Joint](#)
- GetBodyCount() : [b2World](#)
- GetBodyList() : [b2World](#)
- GetCentroid() : [b2PolygonShape](#)
- GetContactCount() : [b2World](#)
- GetCoreVertices() : [b2PolygonShape](#)
- GetFilterData() : [b2Shape](#)
- GetFirstVertex() : [b2PolygonShape](#)
- GetFlags() : [b2DebugDraw](#)
- GetFriction() : [b2Shape](#)
- GetGroundAnchor1() : [b2PulleyJoint](#)
- GetGroundAnchor2() : [b2PulleyJoint](#)
- GetGroundBody() : [b2World](#)
- GetInertia() : [b2Body](#)
- GetJointAngle() : [b2RevoluteJoint](#)
- GetJointCount() : [b2World](#)
- GetJointList() : [b2Body](#) , [b2World](#)
- GetJointSpeed() : [b2PrismaticJoint](#) , [b2RevoluteJoint](#)
- GetJointTranslation() : [b2PrismaticJoint](#)
- GetLength1() : [b2PulleyJoint](#)

- GetLength2() : **b2PulleyJoint**
- GetLinearVelocity() : **b2Body**
- GetLinearVelocityFromLocalPoint() : **b2Body**
- GetLinearVelocityFromWorldPoint() : **b2Body**
- GetLocalCenter() : **b2Body**
- GetLocalPoint() : **b2Body**
- GetLocalPosition() : **b2CircleShape**
- GetLocalVector() : **b2Body**
- GetLowerLimit() : **b2PrismaticJoint , b2RevoluteJoint**
- GetManifoldCount() : **b2Contact**
- GetManifolds() : **b2Contact**
- GetMass() : **b2Body**
- GetMotorForce() : **b2PrismaticJoint**
- GetMotorSpeed() : **b2PrismaticJoint , b2RevoluteJoint**
- GetMotorTorque() : **b2RevoluteJoint**
- GetNext() : **b2Shape , b2Body , b2Contact , b2Joint**
- GetNormals() : **b2PolygonShape**
- GetOBB() : **b2PolygonShape**
- GetPairCount() : **b2World**
- GetPosition() : **b2Body**
- GetProxyCount() : **b2World**
- GetRadius() : **b2CircleShape**
- GetRatio() : **b2GearJoint , b2PulleyJoint**
- GetReactionForce() : **b2PulleyJoint , b2PrismaticJoint , b2GearJoint , b2MouseJoint , b2Joint , b2RevoluteJoint , b2DistanceJoint**
- GetReactionTorque() : **b2PrismaticJoint , b2Joint , b2MouseJoint , b2RevoluteJoint , b2PulleyJoint , b2DistanceJoint , b2GearJoint**
- GetRestitution() : **b2Shape**
- GetShape1() : **b2Contact**
- GetShape2() : **b2Contact**
- GetShapeList() : **b2Body**
- GetSweepRadius() : **b2Shape**
- GetType() : **b2Joint , b2Shape**
- GetUpperLimit() : **b2PrismaticJoint , b2RevoluteJoint**
- GetUserData() : **b2Body , b2Shape , b2Joint**
- GetVertexCount() : **b2PolygonShape**
- GetVertices() : **b2PolygonShape**

- GetWorld() : **b2Body**
  - GetWorldCenter() : **b2Body**
  - GetWorldPoint() : **b2Body**
  - GetWorldVector() : **b2Body**
  - GetXForm() : **b2Sweep , b2Body**
  - groundAnchor1 : **b2PulleyJointDef**
  - groundAnchor2 : **b2PulleyJointDef**
  - groupIndex : **b2FilterData**
- 

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

The screenshot shows the Doxygen documentation interface for class members. At the top, there are tabs for 'Main Page', 'Classes', and 'Files'. Below that is a sub-navigation bar with tabs for 'Alphabetical List', 'Class List', 'Class Hierarchy', and 'Class Members'. Under 'Class Members', there are further tabs for 'All', 'Functions', 'Variables', and 'Enumerator'. At the bottom of this bar is a horizontal menu with letters 'a' through 'z' and a tilde character (~). The 'Functions' tab is currently selected.

Here is a list of all documented class members with links to the class documentation for each member:

- i -

- I : [b2MassData](#)
- id : [b2ManifoldPoint](#) , [b2ContactResult](#) , [b2ContactPoint](#)
- incidentEdge : [b2ContactID::Features](#)
- incidentVertex : [b2ContactID::Features](#)
- Initialize() : [b2PrismaticJointDef](#) , [b2PulleyJointDef](#) ,  
[b2RevoluteJointDef](#) , [b2DistanceJointDef](#)
- IsBullet() : [b2Body](#)
- isBullet : [b2BodyDef](#)
- IsDynamic() : [b2Body](#)
- IsFrozen() : [b2Body](#)
- IsLimitEnabled() : [b2RevoluteJoint](#) , [b2PrismaticJoint](#)
- IsMotorEnabled() : [b2PrismaticJoint](#) , [b2RevoluteJoint](#)
- IsSensor() : [b2Shape](#)
- isSensor : [b2ShapeDef](#)
- IsSleeping() : [b2Body](#)
- isSleeping : [b2BodyDef](#)
- IsSolid() : [b2Contact](#)
- IsStatic() : [b2Body](#)
- IsValid() : [b2Vec2](#) , [b2AABB](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- j -

- joint : [b2JointEdge](#)
- joint1 : [b2GearJointDef](#)
- joint2 : [b2GearJointDef](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- **k** -

- key : **b2ContactID**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- I -

- length : [b2DistanceJointDef](#)
- Length() : [b2Vec2](#)
- length1 : [b2PulleyJointDef](#)
- length2 : [b2PulleyJointDef](#)
- LengthSquared() : [b2Vec2](#)
- linearDamping : [b2BodyDef](#)
- localAnchor1 : [b2PrismaticJointDef](#) , [b2PulleyJointDef](#) ,  
[b2RevoluteJointDef](#) , [b2DistanceJointDef](#)
- localAnchor2 : [b2DistanceJointDef](#) , [b2PulleyJointDef](#) ,  
[b2PrismaticJointDef](#) , [b2RevoluteJointDef](#)
- localAxis1 : [b2PrismaticJointDef](#)
- localCenter : [b2Sweep](#)
- localPoint1 : [b2ManifoldPoint](#)
- localPoint2 : [b2ManifoldPoint](#)
- lowerAngle : [b2RevoluteJointDef](#)
- lowerBound : [b2AABB](#)
- lowerTranslation : [b2PrismaticJointDef](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- m -

- b2Mat22() : [b2Mat22](#)
- major : [b2Version](#)
- maskBits : [b2FilterData](#)
- mass : [b2MassData](#)
- massData : [b2BodyDef](#)
- maxForce : [b2MouseJointDef](#)
- maxLength1 : [b2PulleyJointDef](#)
- maxLength2 : [b2PulleyJointDef](#)
- maxMotorForce : [b2PrismaticJointDef](#)
- maxMotorTorque : [b2RevoluteJointDef](#)
- minor : [b2Version](#)
- motorSpeed : [b2RevoluteJointDef](#) , [b2PrismaticJointDef](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- n -

- next : **b2ContactEdge** , **b2JointEdge**
- normal : **b2ContactPoint** , **b2ContactResult** , **b2Manifold**
- normalImpulse : **b2ManifoldPoint** , **b2ContactResult**
- Normalize() : **b2Vec2**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- O -

- operator \*=() : [b2Vec2](#)
- operator+\_\_() : [b2Vec2](#)
- operator-\_\_() : [b2Vec2](#)
- operator-=\_\_() : [b2Vec2](#)
- other : [b2ContactEdge](#) , [b2JointEdge](#)

The screenshot shows a top-level navigation bar with several tabs: Main Page, Classes, Files, Alphabetical List, Class List, Class Hierarchy, and Class Members. Below this is a secondary navigation bar with tabs: All, Functions, Variables, and Enumerator. At the bottom is a horizontal bar containing letters from 'a' to 'z'.

Here is a list of all documented class members with links to the class documentation for each member:

- p -

- p1 : [b2Segment](#)
- p2 : [b2Segment](#)
- Persist() : [b2ContactListener](#)
- pointCount : [b2Manifold](#)
- points : [b2Manifold](#)
- position : [b2ContactPoint](#) , [b2ContactResult](#) , [b2BodyDef](#)
- prev : [b2JointEdge](#) , [b2ContactEdge](#)
- PutToSleep() : [b2Body](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- **q** -

- Query() : **b2World**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

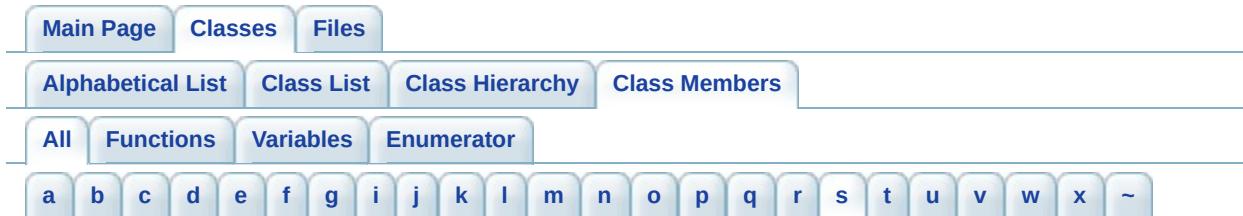
All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- **r** -

- R : [b2OBB](#)
- ratio : [b2GearJointDef](#) , [b2PulleyJointDef](#)
- referenceAngle : [b2PrismaticJointDef](#) , [b2RevoluteJointDef](#)
- referenceEdge : [b2ContactID::Features](#)
- Refilter() : [b2World](#)
- Remove() : [b2ContactListener](#)
- restitution : [b2ContactPoint](#) , [b2ShapeDef](#)
- Result() : [b2ContactListener](#)
- revision : [b2Version](#)



Here is a list of all documented class members with links to the class documentation for each member:

- S -

- b2ShapeDef() : [b2ShapeDef](#)
- SayGoodbye() : [b2DestructionListener](#)
- separation : [b2ContactPoint](#) , [b2ManifoldPoint](#)
- Set() : [b2Vec2](#) , [b2Mat22](#)
- SetAngularVelocity() : [b2Body](#)
- SetAsBox() : [b2PolygonDef](#)
- SetBoundaryListener() : [b2World](#)
- SetBullet() : [b2Body](#)
- SetContactFilter() : [b2World](#)
- SetContactListener() : [b2World](#)
- SetContinuousPhysics() : [b2World](#)
- SetDebugDraw() : [b2World](#)
- SetDestructionListener() : [b2World](#)
- SetFilterData() : [b2Shape](#)
- SetFlags() : [b2DebugDraw](#)
- SetGravity() : [b2World](#)
- SetIdentity() : [b2Mat22](#) , [b2XForm](#)
- SetLimits() : [b2PrismaticJoint](#) , [b2RevoluteJoint](#)
- SetLinearVelocity() : [b2Body](#)
- SetMass() : [b2Body](#)
- SetMassFromShapes() : [b2Body](#)
- SetMaxMotorForce() : [b2PrismaticJoint](#)
- SetMaxMotorTorque() : [b2RevoluteJoint](#)
- SetMotorSpeed() : [b2RevoluteJoint](#) , [b2PrismaticJoint](#)
- SetPositionCorrection() : [b2World](#)
- SetTarget() : [b2MouseJoint](#)
- SetUserData() : [b2Shape](#) , [b2Joint](#) , [b2Body](#)
- SetWarmStarting() : [b2World](#)
- SetXForm() : [b2Body](#)

- SetZero() : **b2Vec2** , **b2Mat22**
  - shape1 : **b2ContactPoint** , **b2ContactResult**
  - shape2 : **b2ContactResult** , **b2ContactPoint**
  - ShouldCollide() : **b2ContactFilter**
  - Solve() : **b2Mat22**
  - Step() : **b2World**
  - Support() : **b2PolygonShape**
- 

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- t -

- t0 : [b2Sweep](#)
- tangentImpulse : [b2ManifoldPoint](#) , [b2ContactResult](#)
- target : [b2MouseJointDef](#)
- TestPoint() : [b2CircleShape](#) , [b2PolygonShape](#) , [b2Shape](#)
- TestSegment() : [b2Shape](#) , [b2Segment](#) , [b2PolygonShape](#) , [b2CircleShape](#)
- timeStep : [b2MouseJointDef](#)
- type : [b2ShapeDef](#) , [b2JointDef](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- **u** -

- upperAngle : [b2RevoluteJointDef](#)
- upperBound : [b2AABB](#)
- upperTranslation : [b2PrismaticJointDef](#)
- userData : [b2ShapeDef](#) , [b2BodyDef](#) , [b2JointDef](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- **v** -

- [b2Vec2\(\)](#) : **b2Vec2**
- [Validate\(\)](#) : **b2World**
- [velocity](#) : **b2ContactPoint**
- [vertexCount](#) : **b2PolygonDef**
- [vertices](#) : **b2PolygonDef**
- [Violation\(\)](#) : **b2BoundaryListener**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- W -

- b2World() : [b2World](#)
- WakeUp() : [b2Body](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- X -

- b2XForm() : **b2XForm**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e f g i j k l m n o p q r s t u v w x ~

Here is a list of all documented class members with links to the class documentation for each member:

- - -
- `~b2World()` : **b2World**

- b2Alloc() : [b2Settings.h](#)
  - b2CollideCircles() : [b2Collision.h](#)
  - b2CollidePolygonAndCircle() : [b2Collision.h](#)
  - b2CollidePolygons() : [b2Collision.h](#)
  - b2Distance() : [b2Collision.h](#)
  - b2Free() : [b2Settings.h](#)
  - b2MixFriction() : [b2Settings.h](#)
  - b2MixRestitution() : [b2Settings.h](#)
  - b2TimeOfImpact() : [b2Collision.h](#)
-



- b2-angularSleepTolerance : [\*\*b2Settings.h\*\*](#)
- b2-angularSlop : [\*\*b2Settings.h\*\*](#)
- b2-byteCount : [\*\*b2Settings.h\*\*](#)
- b2-contactBaumgarte : [\*\*b2Settings.h\*\*](#)
- b2-linearSleepTolerance : [\*\*b2Settings.h\*\*](#)
- b2-linearSlop : [\*\*b2Settings.h\*\*](#)
- b2-maxAngularCorrection : [\*\*b2Settings.h\*\*](#)
- b2-maxAngularVelocity : [\*\*b2Settings.h\*\*](#)
- b2-maxLinearCorrection : [\*\*b2Settings.h\*\*](#)
- b2-maxLinearVelocity : [\*\*b2Settings.h\*\*](#)
- b2-maxTOIContactsPerIsland : [\*\*b2Settings.h\*\*](#)
- b2-timeToSleep : [\*\*b2Settings.h\*\*](#)
- b2-toiSlop : [\*\*b2Settings.h\*\*](#)
- b2-velocityThreshold : [\*\*b2Settings.h\*\*](#)
- b2-version : [\*\*b2Settings.h\*\*](#)

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- b -

- b2BodyDef() : **b2BodyDef**

Generated on Sun Apr 13 15:21:27 2008 for Box2D by **doxygen** 1.5.4



## - C -

- Centroid() : [b2PolygonShape](#)
- ClearFlags() : [b2DebugDraw](#)
- ComputeAABB() : [b2PolygonShape](#) , [b2Shape](#) , [b2CircleShape](#)
- ComputeMass() : [b2Shape](#) , [b2CircleShape](#) , [b2PolygonShape](#)
- ComputeSweptAABB() : [b2CircleShape](#) , [b2PolygonShape](#) , [b2Shape](#)
- CreateBody() : [b2World](#)
- CreateJoint() : [b2World](#)
- CreateShape() : [b2Body](#)



## - d -

- DestroyBody() : **b2World**
- DestroyJoint() : **b2World**
- DestroyShape() : **b2Body**
- DrawCircle() : **b2DebugDraw**
- DrawPolygon() : **b2DebugDraw**
- DrawSegment() : **b2DebugDraw**
- DrawSolidCircle() : **b2DebugDraw**
- DrawSolidPolygon() : **b2DebugDraw**
- DrawXForm() : **b2DebugDraw**



- e -

- EnableLimit() : **b2PrismaticJoint , b2RevoluteJoint**
- EnableMotor() : **b2PrismaticJoint , b2RevoluteJoint**



## - g -

- GetAnchor1() : **b2DistanceJoint** , **b2GearJoint** , **b2MouseJoint** , **b2PrismaticJoint** , **b2Joint** , **b2PulleyJoint** , **b2RevoluteJoint**
- GetAnchor2() : **b2DistanceJoint** , **b2GearJoint** , **b2Joint** , **b2MouseJoint** , **b2PrismaticJoint** , **b2PulleyJoint** , **b2RevoluteJoint**
- GetAngle() : **b2Body** , **b2Mat22**
- GetAngularVelocity() : **b2Body**
- GetBody() : **b2Shape**
- GetBody1() : **b2Joint**
- GetBody2() : **b2Joint**
- GetBodyCount() : **b2World**
- GetBodyList() : **b2World**
- GetCentroid() : **b2PolygonShape**
- GetContactCount() : **b2World**
- GetCoreVertices() : **b2PolygonShape**
- GetFilterData() : **b2Shape**
- GetFirstVertex() : **b2PolygonShape**
- GetFlags() : **b2DebugDraw**
- GetFriction() : **b2Shape**
- GetGroundAnchor1() : **b2PulleyJoint**
- GetGroundAnchor2() : **b2PulleyJoint**
- GetGroundBody() : **b2World**
- GetInertia() : **b2Body**
- GetJointAngle() : **b2RevoluteJoint**
- GetJointCount() : **b2World**
- GetJointList() : **b2Body** , **b2World**
- GetJointSpeed() : **b2PrismaticJoint** , **b2RevoluteJoint**
- GetJointTranslation() : **b2PrismaticJoint**
- GetLength1() : **b2PulleyJoint**
- GetLength2() : **b2PulleyJoint**

- GetLinearVelocity() : **b2Body**
- GetLinearVelocityFromLocalPoint() : **b2Body**
- GetLinearVelocityFromWorldPoint() : **b2Body**
- GetLocalCenter() : **b2Body**
- GetLocalPoint() : **b2Body**
- GetLocalPosition() : **b2CircleShape**
- GetLocalVector() : **b2Body**
- GetLowerLimit() : **b2PrismaticJoint , b2RevoluteJoint**
- GetManifoldCount() : **b2Contact**
- GetManifolds() : **b2Contact**
- GetMass() : **b2Body**
- GetMotorForce() : **b2PrismaticJoint**
- GetMotorSpeed() : **b2PrismaticJoint , b2RevoluteJoint**
- GetMotorTorque() : **b2RevoluteJoint**
- GetNext() : **b2Shape , b2Body , b2Contact , b2Joint**
- GetNormals() : **b2PolygonShape**
- GetOBB() : **b2PolygonShape**
- GetPairCount() : **b2World**
- GetPosition() : **b2Body**
- GetProxyCount() : **b2World**
- GetRadius() : **b2CircleShape**
- GetRatio() : **b2GearJoint , b2PulleyJoint**
- GetReactionForce() : **b2Joint , b2DistanceJoint , b2GearJoint , b2MouseJoint , b2PulleyJoint , b2RevoluteJoint , b2PrismaticJoint**
- GetReactionTorque() : **b2PulleyJoint , b2PrismaticJoint , b2Joint , b2MouseJoint , b2RevoluteJoint , b2DistanceJoint , b2GearJoint**
- GetRestitution() : **b2Shape**
- GetShape1() : **b2Contact**
- GetShape2() : **b2Contact**
- GetShapeList() : **b2Body**
- GetSweepRadius() : **b2Shape**
- GetType() : **b2Joint , b2Shape**
- GetUpperLimit() : **b2RevoluteJoint , b2PrismaticJoint**
- GetUserData() : **b2Body , b2Joint , b2Shape**
- GetVertexCount() : **b2PolygonShape**
- GetVertices() : **b2PolygonShape**
- GetWorld() : **b2Body**

- GetWorldCenter() : **b2Body**
  - GetWorldPoint() : **b2Body**
  - GetWorldVector() : **b2Body**
  - GetXForm() : **b2Sweep , b2Body**
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*



- i -

- Initialize() : **b2DistanceJointDef** , **b2PrismaticJointDef** , **b2RevoluteJointDef** , **b2PulleyJointDef**
- IsBullet() : **b2Body**
- IsDynamic() : **b2Body**
- IsFrozen() : **b2Body**
- IsLimitEnabled() : **b2PrismaticJoint** , **b2RevoluteJoint**
- IsMotorEnabled() : **b2RevoluteJoint** , **b2PrismaticJoint**
- IsSensor() : **b2Shape**
- IsSleeping() : **b2Body**
- IsSolid() : **b2Contact**
- IsStatic() : **b2Body**
- IsValid() : **b2Vec2** , **b2AABB**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- | -

- Length() : **b2Vec2**
- LengthSquared() : **b2Vec2**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- M -

- b2Mat22() : **b2Mat22**

Generated on Sun Apr 13 15:21:27 2008 for Box2D by **doxygen** 1.5.4



- n -

- Normalize() : **b2Vec2**

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by **doxygen** 1.5.4

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

## - O -

- operator \*=() : **b2Vec2**
- operator+\_\_() : **b2Vec2**
- operator-\_\_() : **b2Vec2**
- operator-\_=() : **b2Vec2**



- p -

- Persist() : **b2ContactListener**
- PutToSleep() : **b2Body**



- q -

- Query() : **b2World**

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by **doxygen** 1.5.4

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- r -

- Refilter() : **b2World**
- Remove() : **b2ContactListener**
- Result() : **b2ContactListener**



## - S -

- b2ShapeDef() : [b2ShapeDef](#)
- SayGoodbye() : [b2DestructionListener](#)
- Set() : [b2Mat22](#) , [b2Vec2](#)
- SetAngularVelocity() : [b2Body](#)
- SetAsBox() : [b2PolygonDef](#)
- SetBoundaryListener() : [b2World](#)
- SetBullet() : [b2Body](#)
- SetContactFilter() : [b2World](#)
- SetContactListener() : [b2World](#)
- SetContinuousPhysics() : [b2World](#)
- SetDebugDraw() : [b2World](#)
- SetDestructionListener() : [b2World](#)
- SetFilterData() : [b2Shape](#)
- SetFlags() : [b2DebugDraw](#)
- SetGravity() : [b2World](#)
- SetIdentity() : [b2Mat22](#) , [b2XForm](#)
- SetLimits() : [b2PrismaticJoint](#) , [b2RevoluteJoint](#)
- SetLinearVelocity() : [b2Body](#)
- SetMass() : [b2Body](#)
- SetMassFromShapes() : [b2Body](#)
- SetMaxMotorForce() : [b2PrismaticJoint](#)
- SetMaxMotorTorque() : [b2RevoluteJoint](#)
- SetMotorSpeed() : [b2PrismaticJoint](#) , [b2RevoluteJoint](#)
- SetPositionCorrection() : [b2World](#)
- SetTarget() : [b2MouseJoint](#)
- SetUserData() : [b2Joint](#) , [b2Body](#) , [b2Shape](#)
- SetWarmStarting() : [b2World](#)
- SetXForm() : [b2Body](#)
- SetZero() : [b2Vec2](#) , [b2Mat22](#)
- ShouldCollide() : [b2ContactFilter](#)

- `Solve()` : **b2Mat22**
  - `Step()` : **b2World**
  - `Support()` : **b2PolygonShape**
- 

*Generated on Sun Apr 13 15:21:27 2008 for Box2D by*  *1.5.4*

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- t -

- TestPoint() : **b2CircleShape** , **b2PolygonShape** , **b2Shape**
- TestSegment() : **b2CircleShape** , **b2Segment** , **b2PolygonShape** , **b2Shape**

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by  1.5.4

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- V -

- [b2Vec2\(\)](#) : **b2Vec2**
- [Validate\(\)](#) : **b2World**
- [Violation\(\)](#) : **b2BoundaryListener**

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- W -

- `b2World()` : **b2World**
- `WakeUp()` : **b2Body**



- X -

- b2XForm() : **b2XForm**

---

Generated on Sun Apr 13 15:21:27 2008 for Box2D by **doxygen** 1.5.4

Main Page Classes Files

Alphabetical List Class List Class Hierarchy Class Members

All Functions Variables Enumerator

a b c d e g i l m n o p q r s t v w x ~

- ~ -

- `~b2World()` : **b2World**

Generated on Sun Apr 13 15:21:27 2008 for Box2D by **doxygen** 1.5.4