# Internal Breakable Joint


- Internal Breakable Joint
- Line Constraint
- Plane Constraint
- Point Constraint


Creates an internal breakable joint between physical entity parts. A joint can either connect two parts, or one part to the 'ground' (connecting parts are determined automatically by checking intersections). When the entity receives impulses (from collisions or other sources), it recomputes internal forces between joints and breaks the ones that have their limits breached. Broken joints can optionally create a dynamic constraint between broken parts instead of fully disconnecting them. If the limit that broke was twist, it creates a 'hinge' (1 DOF) constraint around the joint's z axis, otherwise a bend constraint (2 DOF).

Parts that separate from the main entity are always physicalized as rigid entities (PE_RIGID).

| Property | Description |
|---|---|
| **Internal Breakable Joint Settings** | |

| Setting | Description |
|---|---|
| **Max Push Force** | Push force limit along the normal. |
| **Max Pull Force** | Push force limit along the normal. |
| **Max Shift Force** | Force limit orthogonal to the normal. |
| **Max Bend Torque** | Normal bend torque limit. |
| **Max Twist Torque** | Twist torque limit around the normal. |
| **DefaultDamageAccum** | Use default damage accumulation and threshold (from CVar). |
| **Damage Accumulation Fraction** | Accumulated fraction of damage, normalized to 0..1. |
| **Damage Accumulation Threshold** | Only accumulate damage if it exceeds this fraction of the maximum. |
| **Breakable** | Whether the joint is at all breakable. |
| **Direct Breaks Only** | The joint will only break if one of the parts it connects receives an impulse. |
| **Sensor Size** | Sensor sphere diameter for attachment and re-attachment. |

**Dynamic Constraint** — Properties of the constraint created after joint breaking.

| Setting | Description |
|---|---|
| **Min Angle** | Min angle for the dynamic constraint (for bend assumed to be 0). |
| **Max Angle** | Max angle for the dynamic constraint. |
| **Force Limit** | Bend torque limit for breaking the dynamic constraint (0 mean the joint fully breaks at once) |
| **Ignore Collisions** | Ignore collisions between the connected parts. |
| **Damping** | Damping applied along the constraint's allowed degrees of freedom. |

# Line Constraint

Constrains entity movement to a line, relative to another entity. The other entity is passed to the ConstrainToEntity function, or it can be 'the world' if ConstrainToPoint is used. Rotation can optionally be fully disabled as well.

| Setting | Description |
|---|---|
| **Active** | Whether or not the constraint should be added when the component is reset. |

| | |
|---|---|
| **Lock Rotation** | Whether or not the constrained object will be allowed to rotate around the axis (rotation that would steer the entity away from the axis is locked always). |
| **Axis** | Axis around which the physical entity is constrained.<br><br>If one of these axes is already set to a positive value and you change a different one, the difference will be split between both axes. This allows you to create more complex movement along multiple axes at the same time. |
| **Minimum Limit** | Distance limit along the axis. Twist rotation limit (in angles) for splines. A unit of -1 indicates that the minimum position is one meter behind the entity. |
| **Maximum Limit** | \| Distance limit along the axis. Twist rotation limit (in angles) for splines. A unit of +1 indicates that the maximum position is one meter ahead of the entity. |
| **Damping** | Determines how much the object loses momentum (as it moves) while the constraint is active. |
| **Attachment Parameters** | |
| **No Attachment Collisions** | If the Constraint is attached to another entity (either via Target Link Name or Auto Attachment Distance, see below), automatically disable collisions with it. |
| **Auto Attachment Distance** | When >0, will sample physical environment within this distance to find an entity to attach to.<br><br>Any physical constraint constrains 2 entities. the first one is always the one that owns the component. the other one can be auto-detected. |
| **Target Link Name** | Name of the entity link that contains the entity to attach to. |
| **Helper Link Name** | Name of the entity link that contains the 'constraint helper' entity, such a physical area with a surface or a spline.<br><br>Target Link Name and Helper Link Name are needed to give the component a reference to another entity, for example you create an entity link with a certain name, and specify that link name as the Target Link Name. |

## Plane Constraint

Constrains entity movement to a plane, relative to another entity. The other entity is passed to ConstrainToEntity function, or it can be 'the world' if ConstrainToPoint is used. Rotation can optionally be limited as well.

| Setting | Description |
|---|---|
| **Active** | Whether or not the constraint should be added on component reset. |
| **Axis** | The normal of the plane the entity will be constrained to (in the component's frame). |
| **Twist rotation min angle** | Rotation limits minimum around x axis. (If this value is bigger than the max value the x axis will be locked) |
| **Twist rotation max angle** | Rotation limits maximum around x axis. |
| **Bend max angle** | Maximum bend angle of the rotation. |
| **Damping** | Determines how much the object loses momentum (as it moves) while the constraint is active. |
| **Attachment Parameters** | |
| **No Attachment Collisions** | If the Constraint is attached to another entity (either via Target Link Name or Auto Attachment Distance, see below), automatically disable collisions with it. |

| | |
|---|---|
| **Auto Attachment Distance** | When >0, will sample physical environment within this distance to find an entity to attach to.<br><br>Any physical constraint constrains 2 entities. the first one is always the one that owns the component. the other one can be auto-detected. |
| **Target Link Name** | Name of the entity link that contains the entity to attach to. |
| **Helper Link Name** | Name of the entity link that contains the 'constraint helper' entity, such a physical area with a surface or a spline.<br><br>Target Link Name and Helper Link Name are needed to give the component a reference to another entity, for example you create an entity link with a certain name, and specify that link name as the Target Link Name. |

## Point Constraint

Creates a point-to-point constraint between 2 entities. The other entity is passed to ConstrainToEntity function, or it can be 'the world' if ConstrainToPoint is used. Rotation can be constrained in two dimensions: as rotation around the constraint axis and as bending of that axis on one entity's frame relative to it the other entity's frame.

| Setting | Description |
|---|---|
| **Active** | Whether or not the constraint should be added when the component is reset. |
| **Axis** | Axis around which the physical entity is constrained. |
| **Free Position** | Constrains the rotation, leaving the position free. |
| **Minimum X Angle** | Minimum angle for rotation around the axis ('twisting') |
| **Maximum X Angle** | Maximum angle for rotation around the axis ('twisting') |
| **Maximum YZ Angle** | Maximum angle for bending of the axis. |
| **Damping** | Determines how much the object loses momentum (as it moves) while the constraint is active.<br><br>Values of 0.1 to 0.4 are very common and realistic. |
| **Attachment Parameters** | |
| **No Attachment Collisions** | If the Constraint is attached to another entity (either via Target Link Name or Auto Attachment Distance, see below), automatically disable collisions with it. |
| **Auto Attachment Distance** | When >0, will sample physical environment within this distance to find an entity to attach to.<br><br>Any physical constraint constrains 2 entities. the first one is always the one that owns the component. the other one can be auto-detected. |
| **Target Link Name** | Name of the entity link that contains the entity to attach to. |
| **Helper Link Name** | Name of the entity link that contains the 'constraint helper' entity, such a physical area with a surface or a spline.<br><br>Target Link Name and Helper Link Name are needed to give the component a reference to another entity, for example you create an entity link with a certain name, and specify that link name as the Target Link Name. |