# Overview

Material effects expose the way a surface material reacts to other materials. For example, a metal material will react to bullet impacts differently (i.e. by generating sparks) than a grass material does (i.e. by generating dirt or dust).

Because hard-coding these effects in code would be impossible to keep updated, these effects are exposed through a small number of asset files.



# Files Overview

- The **SurfaceTypes**, defined in `Game/Libs/MaterialEffects/SurfaceTypes.xml` defines the physical properties of all the available material types. It can be edited with any text editor.
- The **MaterialEffects**, which usually resides in `Game/Libs/MaterialEffects/MaterialEffects.xml`, define the interaction of 2 materials, i. e. what effect to generate on an interactive event between 2 surface types. For example, it defines that when a bullet collides with the soil surface, a dirt particle effect is spawned. This file has to be read and edited with Microsoft Excel.
- The **Effect** libraries, found in `Game/Libs/MaterialEffects/FXLibs`, contain the associated effects. The ammo files in `Game/Scripts/Entities/Items/Ammo` define what material a bullet is made from.

# Creating a New Surface Type

## Adding a Surface type to SurfaceTypes.xml

A surface type is defined by the XML element **SurfaceType**. Its attribute **name** is the one that can be selected in the Sandbox Editor.

Surface types defined here will appear as options in the drop down list of the Material Editor. The other attribute **type** is optional. It can be used in other processes by game code. The physical parameters are defined by the **Physics** element.

### Physics block

| Parameter | Description |
|---|---|
| **friction** | A material's friction (friction is the average of contacting materials). A recommended default is 0.8. Internally it's clamped to 0.1 if it's below that. |
| **elasticity** | Also known as Bounciness. It's averaged from 2 materials. Clamped to 0 as the elasticity can't be negative. Recommended default is 0, unless you specifically want something bouncy. |
| **breakable_id** | Procedural breakability index; a corresponding boolean cutter shape should be registered in the physics.lua. |
| **pierceability** | Integer from 0-15, with 0 being the least pierceable. For each ray, a hit is pierceable if ray's pierceability is less than the material's pierceability. |
| **break_energy** | Energy= (mass*velocity^2/2) and it refers to the colliding object that triggers breaking (typically should be tweaked based on bullets' params). |
| **hit_points** | Each collision above break_energy takes round_to_int(collision_energy/break_energy) hitpoints; real breaking only happens when hitpoints are depleted. |
| **hit_maxdmg** | Max amount of hitpoints a single hit can take. |
| **hit_radius** | Hitpoints can be localized with this radius (i.e. it'll require that all hits are within this radius from each other); 0 disables. |
| **hit_lifetime** | Lifetime for localized hits info, it expires after that (note that it's just a hint; it can expire earlier if there are too many newer hit infos, or later if there's no concurrency). |
| **hole_size** | Characteristic size for boolean carving; shape registered in physics.lua is scaled by this size / shape's characteristic size (specified in physics.lua). |
| **hole_size_explosion** | Same as above, but for explosions (first one is for collisions with bullets and general physicalized objects). |
| **breakable_2d** | If set, uses 2D breakability instead of 3D boolean carving. |

| | |
|---|---|
| **vehicle_only_collisions** | Entire node will only collide with vehicle (even if material with this flag is only applied to some triangles of it). |

```
<SurfaceType name="mat_new_material">
 <Physics friction="0.5" elasticity="1"  pierceability="3" can_shatter="1"  />
</SurfaceType>
```

The breakable parameters are defined by the **breakable_2d** element.

### breakable_2d block

| Parameter | Description |
|---|---|
| **blast_radius** | Procedural hole size (note that when an object hits the surface, it will use the colliding area's size instead – unless it's smaller than this). |
| **blast_radius_first** | Same as above, but for the first hit that breaks the object. |
| **vert_size_spread** | Vertical hole size = blast_radius * (1+random(0..vert_size_spread)). |
| **rigid_body** | Broken pieces will be physicalized as rigid bodies (otherwise as physical particles). |
| **lifetime** | Pieces will expire after that (currently it'll only happen when the player looks away, but that might change). |
| **cell_size** | Cell size for the breakage grid (default 0.1). |
| **max_patch_tris** | Governs the number of grid triangles in each shard (triangles are taken from the breakage grid, around the hit point and within blast_radius); default is 6. |
| **filter_angle** | Filters the hole's edges with this angle to avoid the 'saw tooth' look; default 0 (disabled), but 75 seems to look good in most cases. |
| **shard_density** | Used to calculate shards' masses and buoyancy (float/sink) properties; default 1200. |
| **max_fracture** | Triggers full fracture when the object is damaged beyond this (0..1); default 1. |
| **particle_effect** | Name of the effect to generate along the broken edges. |
| **fracture_fx** | Single effect for each breakage. |
| **full_fracture_fx** | Effect to play when a full fracture condition is triggered. |
| **no_procedural_full_fracture** | In case of full fracture, doesn't generate broken pieces procedurally, relies only on full_fracture_fx. |
| **use_edge_alpha** | Sets alpha of vertices along the broken edges to 0 (should be used if there's a special shader that expects that). |
| **crack_decal_mtl** | Puts a decal with this material whenever a break occurs. |
| **crack_decal_scale** | Sets the decal size wrt the hole size (0 disables crack decals). |

Besides the **<Physics>** element, the optional **<BreakageParticles>** defines the type of particle that will be spawned on breakage. Its attribute **type** defines the break event and can take one of the following values:

- breakage
- joint_break
- joint_shatter
- destroy

The **effect** attribute points to the particle effect that will be spawned on breakage, along with the attributes **count_per_unit**, **count_scale** and **scale** defining respectively the number of particle systems to be spawned, and their scale. Normally, their value should be **1**.

Note

Ammo surface types do not need to be added to the SurfaceTypes.xml file.

## Adding Events to MaterialEffects.xml

New surface types have to be added both as a row and as a column, and have to be kept in the same order. The cross reference between 2 surface types defines the type of event that will happen on interaction.

Surface types that are only called by code are required to only have rows and must be added at the bottom, below the surface types defined in **SurfaceTypes.xml** (the exception is when it is needed to be used on materials through the Material Editor).

Obsolete:

> *It should be noted that Objects which need to use Alpha Test for Collision (railings, etc) should have a surface type with the suffix _RayProxy. This is also used to allow characters to see through glass objects and such as part of the perception system.*

| | mat_metal_shell_shotgun | mat_metal_shell_small |
|---|---|---|
| **mat_metal_shell_shotgun** | bulletcasings:shotgun_shell_metal_thick | bulletcasings:shotgun_shell_metal_thick |
| **mat_metal_shell_small** | bulletcasings:small_shell_metal_thick | bulletcasings:small_shell_metal_thick |

## Adding Material Effect Events

Material effect (MFX) events are those events that occur at interactions between 2 surface types, as defined in **MaterialEffects.xml**. They are named according to the following schema:

```
<MFX_lib_name>:<MFX_name>
```

For instance, **bulletimpacts:hit_mat_soil** points to the event (and XML element) **hit_mat_soil** defined in `Game/Libs/MaterialEffects/FXLibs/bulletimpacts.xml`.

This event is defined as follows:

```
<Effect name="hit_mat_soil" delay="0.05">
        <Sound>
                <Name>sounds/physics:bullets/impacts:grass</Name>
        </Sound>
        <RandEffect>
                <Decal minscale="0.03" maxscale="0.07">
                        <Filename>textures/decals/metal.dds</Filename>
                        <Material>materials/decals/terrain_soil</Material>
                </Decal>
        </RandEffect>
        <RandEffect>
                <Particle>
                        <Name direction="normal" minscale="0.6" maxscale="1.2" maxscaledist="80">bullet.
hit_soil.a</Name>
                </Particle>
                <Particle>
                        <Name direction="ricochet" minscale="0.6" maxscale="1.2" maxscaledist="80">bullet.
hit_soil.b</Name>
                </Particle>
                <Particle>
                        <Name direction="reverse" minscale="0.6" maxscale="1.2" maxscaledist="80">bullet.
hit_soil.c</Name>
                </Particle>
        </RandEffect>
        <FlowGraph name="dirt_bullet_hit" maxdist="2"/>
</Effect>
```

Each effect Name can have the following optional additional attributes:

| Attribute | Description |
|---|---|
| **direction** | Effect spawn direction:<br>"Normal" : (default) surface normal.<br>"Ricochet" : reflection of hit source direction off surface normal.<br>"Reverse" : reverse of hit source direction. |
| **minscale** / **maxscale** | Scale to apply to effect when it's near / far from camera. |
| **maxscaledist** | Distance at which **maxscale** applies. |
| **maxdist** | Max distance to show effect. |

## Adding an Ammo Surface Type

The surface type for ammunition is the attribute **name** of the XML element **ammo** in an ammo file. Its value is required to be the same as the value in the MFX table.

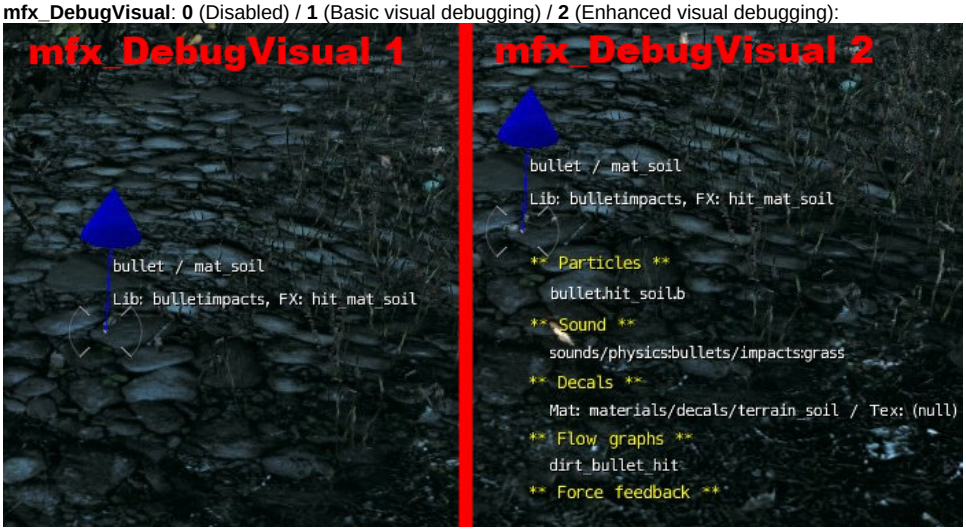
## Debugging

There exist a couple of console variables to debug the Material Effects in the Sandbox Editor or the Launcher:

- **mfx_Debug**: **0** (Disabled) / **1** (Collisions) / **2** (Breakage) / **3** (Both)

```
[MFX] Running effect for:
      : Mat0=mat_metal_shell
      : Mat1=mat_concrete
impact-speed=5.756900 fx-threshold=2.000000 mass=0.018203 speed=0.147297
```

- **mfx_DebugVisual**: **0** (Disabled) / **1** (Basic visual debugging) / **2** (Enhanced visual debugging):




- **mfx_DebugVisualFilter**: **0** (Disabled) / **Name** (Only show visual debug information for this Material Effect).
- **mfx_DebugFlowGraphFX**: **0** (Disabled) / **1** (Show debug information for triggered Material Effect flowgraphs)

```
Material FlowGraphFX manager: Effect Test1 was triggered.
Material FlowGraphFX manager: Effect 'Test1' .Dynamic parameter 'BlendOutTime' set to value 0.056
```

Note

Non visual debug information will be shown in the console.

## Reloading

To reload all Material Effects from the *MaterialEffect.xml* file it is possible to use the console command **mfx_reload**.

Note

Because it is necessary to have matching rows and columns in the MaterialEffects.xml file it is possible to set mfx_debug to 1/2/3 for an automatic check before reloading the Material Effects. Any warnings are outputted to the console.

## Known issues

The MFX table is prone to breakage and unexpected behavior when the order in rows and columns is different from the expected one, e.g. after using find-replace and/or inserting cells.