

Adding New Decals

Decal system has a library of decal types (represented by *DecalType* structure) and decal instances on a level. Decal instances may be **static**, in which case they have unlimited life span and are saved with level. Decal instances may be **dynamic**, in which case they have limited life span and are not saved with level.

This document describes the process of adding **dynamic** decals to the level.

class DecalChief provides access to both decal library and decal instances related functions. A global instance *g_pDecalChief* of this class is present in the engine. The following code snippet adds decal to the level:

```
DecalParams params;  
params.Pos          = blood_pos;  
params.Dir          = blood_norm;  
params.TypeID       = GetDecalID( r3dHash::MakeHash("default"),  
                                   r3dHash::MakeHash("blood_ground") );  
  
if( params.TypeID != INVALID_DECAL_ID )  
    g_pDecalChief->Add( params );
```

DecalParams structure, filled in this snippet, serves 2 functions. First, it allows passing decal creation parameters to *DecalChief*. Second, it stores dynamic parameters of the decal instance – such as *LifeTime*. Complete list of *DecalParams* fields is as follows:

Name	Type	Description
<i>TypeID</i>	<i>int</i>	Decal type index
<i>Pos</i>	<i>r3dPoint3D</i>	Decal projection position
<i>Dir</i>	<i>r3dPoint3D</i>	Decal projection direction
<i>LifeTime</i>	<i>float</i>	Life time of the decal. If this value is less than 0 (default is -1), it is initialized from the corresponding <i>DecalType</i> structure. Otherwise the value specified overrides the value from <i>DecalType</i> structure
<i>ZRot</i>	<i>float</i>	Rotation of the decal around Z Axis (axis of projection) in radians. In case <i>DecalType</i> structure suggest random rotation, user specified value is overridden with randomly generated value.
<i>ScaleCoef</i>	<i>float</i>	Additional scale coefficient. Default value is 0. In case of 0, <i>DecalType</i> suggested value is used. Otherwise, user specified value is used.

GetDecalID Function.

This function is used to retrieve decal type id according to material and source names. In order to operate faster, string hashes are used. This function uses *MaterialTypes* class internally, which links material type names with impact information. Common *MaterialTypes.xml* is used to define these links.

For terrain impacts, separate *r3dTerrain::GetDecalID* should be used.