

# Material Config XML

Example XML:

```
<materials version="3">
  <material name="mat_name" render_template="generic:DIFFUSE_TEXTURE:NORMALMAP" version="2">
    <diffuse_texture[]file="texture/file/path/my_texture_df"/>
    <bump_normal_texture[]file="texture/file/path/my_texture_nm"/>
  </material>
</materials>
```

TEMPLATES ARE **NOT** MODULAR!

A full list of template variables can be found in `shaders/base.render_template_database`

- `<materials/>` contains all of the material xml for the file.
  - Typically uses `version="3"`
- `<material/>` contains the information of a material.
  - Typically uses `version="2"`
- `<diffuse_texture/>` a variable that often uses a texture with the identifier `_df` at the end.
- `<bump_normal_texture/>` a variable that often uses a texture with the identifier `_nm` at the end.
- `<material_texture/>` a variable that often uses a texture with the identifier `_gsma` at the end. (Gloss, Specular, Metalness, Alpha)
- `<reflection_texture/>` is a variable that loads a cubemap to a material that accepts cubemaps.
  - `<reflection_texture type="cubemap" global_texture="current_global_texture/>` will use the Environment set cubemap.
  - `<reflection_texture type="cubemap" file="texture/file/path/my_cubemap"/>` will use a locked cubemap and wont change from environments.

## Special Use Cases

- `<material/>` control options:
  - `name="mat_name"` give your material a name to go with your models material name.
  - `render_template="template_name"` a full list of templates can be found in `shaders/base.render_template_database`
  - `unique="true"` is usually used for materials that will change per-unit. Unit contours for example.

- `src="name"` will clone the values from an existing `<material>` to your current material.
    - cloned materials can be modified further by including variables or textures to override the source copy.
  - `decal_material="id"` graphic mesh faces with the material will use the `decal_material` variable for hit effects.
    - List of decal IDs can be found in `lib\tweak_data\tweakdata.lua` `self.materials = {list}`
  - `diffuse_color="255 255 255 255"` potentially unused. (Red, Green, Blue, Alpha)
  - `version="2"` most if not all templates use version 2.
  - `<variable/>` control options:
    - Animations can control some variables using the listener type.
      - Standard variable: `<variable name="il_tint" type="vector3" value="1 1 1"/>`
      - Listener variable: `<variable name="il_tint" type="listener" value="light::color" object="lo_lightobject"/>`
- 

Revision #11

Created 21 August 2020 04:45:15 by Rex

Updated 2 July 2021 16:43:16 by Rex