TGTools

Generated by Doxygen 1.8.20

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 tgt Namespace Reference

Namespaces

- Actor
- Font
- Map
- Material
- Model
- Pipe
- Sampler
- Shader
- Texture
- Util

4.2 tgt::Actor Namespace Reference

Classes

- struct ActorData

Functions

- const Result add (const std::string &name, const std::string &material)
- const Result _dataHeader (const fs::path &name, ActorData ∗data)
- const Result remove (const std::string &name)
- const std::string list ()
- template<class T , class U , class V , typename = std::enable_if_t<Util::_validJson<T> && Util::_validString<V> && Util::_valid↩String<U>>>
  const Result change (V actorname, U key, T value)
- void setData (const void ∗data, const uint32_t byteSize, const std::string &name, bool append=false)
- const Result getData (const void ∗∗data, const fs::path &name, size_t ∗ptr=nullptr)

Variables

- constexpr auto ACTOR_SUBFOLDER = "Actors"
- constexpr auto ACTOR_VERTEX_EXTENSION = "tgvmdl"
- constexpr auto ACTOR_INDEX_EXTENSION = "tgimdl"
- constexpr auto MATRIX_PROPERTY = "matrix"
- constexpr auto ANIMATION_PROPERTY = "animationIndex"
- constexpr auto DYNAMIK_TRANSFORM_PROPERTY = "transformIndex"
- constexpr auto MATERIAL_PROPERTY = "material"
- constexpr auto LAYER_PROPERTY = "layer"
- constexpr auto INDEX_COUNT = "indexDrawCount"
- constexpr auto VERTEX_COUNT = "vertexCount"
- constexpr std::array SUPPORTED_PROPERTIES
- const auto ACTOR_PATH = fs::path(Util::RESOURCE_LOCATION).append(ACTOR_SUBFOLDER)

4.2.1 Function Documentation

4.2.1.1 _dataHeader()

```
const Result tgt::Actor::_dataHeader (
            const fs::path & name,
            ActorData * data )
```

Here is the caller graph for this function:

4.2.1.2 add()

```
const Result tgt::Actor::add (
            const std::string & name,
            const std::string & material )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.1.3 change()

```
template<class T , class U , class V , typename = std::enable_if_t<Util::_validJson<T> &&
Util::_validString<V> && Util::_validString<U>>>
const Result tgt::Actor::change (
            V actorname,
            U key,
            T value )  [inline]
```

Here is the call graph for this function:

4.2.1.4 getData()

```
const Result tgt::Actor::getData (
             const void ** data,
             const fs::path & name,
             size_t * ptr = nullptr )  [inline]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.2.1.5 list()

```
const std::string tgt::Actor::list ( )  [inline]
```

Here is the call graph for this function:

4.2.1.6 remove()

```
const Result tgt::Actor::remove (
             const std::string & name )  [inline]
```

Here is the call graph for this function:

4.2.1.7 setData()

```
void tgt::Actor::setData (
             const void * data,
             const uint32_t byteSize,
             const std::string & name,
             bool append = false )  [inline]
```

Here is the call graph for this function:

4.2.2 Variable Documentation

4.2.2.1 ACTOR_INDEX_EXTENSION

```
constexpr auto tgt::Actor::ACTOR_INDEX_EXTENSION = "tgimdl"  [constexpr]
```

4.2.2.2 ACTOR_PATH

```
const auto tgt::Actor::ACTOR_PATH = fs::path(Util::RESOURCE_LOCATION).append(ACTOR_SUBFOLDER)
```

4.2.2.3 ACTOR_SUBFOLDER

```
constexpr auto tgt::Actor::ACTOR_SUBFOLDER = "Actors"  [constexpr]
```

4.2.2.4 ACTOR_VERTEX_EXTENSION

```
constexpr auto tgt::Actor::ACTOR_VERTEX_EXTENSION = "tgvmdl"  [constexpr]
```

4.2.2.5 ANIMATION_PROPERTY

```
constexpr auto tgt::Actor::ANIMATION_PROPERTY = "animationIndex"  [constexpr]
```

4.2.2.6 DYNAMIK_TRANSFORM_PROPERTY

```
constexpr auto tgt::Actor::DYNAMIK_TRANSFORM_PROPERTY = "transformIndex"  [constexpr]
```

4.2.2.7 INDEX_COUNT

```
constexpr auto tgt::Actor::INDEX_COUNT = "indexDrawCount"  [constexpr]
```

4.2.2.8 LAYER_PROPERTY

```
constexpr auto tgt::Actor::LAYER_PROPERTY = "layer"  [constexpr]
```

4.2.2.9 MATERIAL_PROPERTY

```
constexpr auto tgt::Actor::MATERIAL_PROPERTY = "material"  [constexpr]
```

4.2.2.10 MATRIX_PROPERTY

```
constexpr auto tgt::Actor::MATRIX_PROPERTY = "matrix"  [constexpr]
```

4.2.2.11 SUPPORTED_PROPERTIES

```
constexpr std::array tgt::Actor::SUPPORTED_PROPERTIES  [constexpr]
```

Initial value:

```
= { MATRIX_PROPERTY, ANIMATION_PROPERTY,
       DYNAMIK_TRANSFORM_PROPERTY, MATERIAL_PROPERTY, LAYER_PROPERTY, INDEX_COUNT, VERTEX_COUNT }
```

4.2.2.12 VERTEX_COUNT

```
constexpr auto tgt::Actor::VERTEX_COUNT = "vertexCount"  [constexpr]
```

4.3 tgt::Font Namespace Reference

Classes

- struct TGFHeader

Functions

- static const Result generatefontbitmap (const fs::path &font, const std::string &resourceLocation, const std::string &texture)
- const Result add (const std::string &path)
- const Result remove (const std::string &name)
- const std::string list ()

Variables

- constexpr auto TGF_HEADER_VERSION = 1
- constexpr auto FONT_SUBFOLDER = "Font"
- constexpr auto FONT_EXTENSION = ".tgf"
- const auto FONT_PATH = fs::path(Util::RESOURCE_LOCATION).append(FONT_SUBFOLDER)

4.3.1 Function Documentation

4.3.1.1 add()

```
const Result tgt::Font::add (
             const std::string & path )
```

Here is the call graph for this function:

4.3.1.2 generatefontbitmap()

```
static const Result tgt::Font::generatefontbitmap (
             const fs::path & font,
             const std::string & resourceLocation,
             const std::string & texture )  [static]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.3.1.3 list()

```
const std::string tgt::Font::list ( )  [inline]
```

Here is the call graph for this function:

4.3.1.4 remove()

```
const Result tgt::Font::remove (
             const std::string & name )  [inline]
```

Here is the call graph for this function:

4.3.2 Variable Documentation

4.3.2.1 FONT_EXTENSION

```
constexpr auto tgt::Font::FONT_EXTENSION = ".tgf"  [constexpr]
```

4.3.2.2 FONT_PATH

```
const auto tgt::Font::FONT_PATH = fs::path(Util::RESOURCE_LOCATION).append(FONT_SUBFOLDER)
```

4.3.2.3 FONT_SUBFOLDER

```
constexpr auto tgt::Font::FONT_SUBFOLDER = "Font"  [constexpr]
```

4.3.2.4 TGF_HEADER_VERSION

```
constexpr auto tgt::Font::TGF_HEADER_VERSION = 1  [constexpr]
```

4.4 tgt::Map Namespace Reference

Functions

- const Result create (const std::string &mapname)
- static void textureToMapFile (FILE ∗fp, const js::json &map)
- static Result materialToMapFile (FILE ∗fp, const js::json &map)
- static Result actorToMapFile (FILE ∗fp, const js::json &map)
- const Result make (const std::string &mapname)
- const Result add (const std::string &mapname, const std::string &name)
- const Result remove (const std::string &mapname, const std::string &name)
- const Result remove (const std::string &mapname)
- const std::string list ()
- const bool checkDependent (const std::string &dependency)

Variables

- constexpr auto MAP_SUBFOLDER = "Maps"
- constexpr auto MAP_EXTENSION = ".tgr"
- constexpr auto TEXTURE_PROPERTY = "textures"
- constexpr auto ACTOR_PROPERTY = "actors"
- constexpr auto MATERIAL_PROPERTY = "materials"
- constexpr auto FONT_PROPERTY = "fonts"
- const auto MAP_PATH = fs::path(Util::RESOURCE_LOCATION).append(MAP_SUBFOLDER)

4.4.1 Function Documentation

4.4.1.1 actorToMapFile()

```
static Result tgt::Map::actorToMapFile (
             FILE ∗ fp,
             const js::json & map )  [static]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.2 add()

```
const Result tgt::Map::add (
             const std::string & mapname,
             const std::string & name )
```

Here is the call graph for this function:

4.4.1.3 checkDependent()

```
const bool tgt::Map::checkDependent (
             const std::string & dependency )  [inline]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.4 create()

```
const Result tgt::Map::create (
             const std::string & mapname )
```

Here is the call graph for this function:

4.4.1.5 list()

```
const std::string tgt::Map::list ( )  [inline]
```

Here is the call graph for this function:

4.4.1.6 make()

```
const Result tgt::Map::make (
             const std::string & mapname )
```

Here is the call graph for this function:

4.4.1.7 materialToMapFile()

```
static Result tgt::Map::materialToMapFile (
             FILE * fp,
             const js::json & map )  [static]
```

Here is the caller graph for this function:

4.4.1.8 remove() [1/2]

```
const Result tgt::Map::remove (
             const std::string & mapname )  [inline]
```

Here is the call graph for this function:

4.4.1.9 remove() [2/2]

```
const Result tgt::Map::remove (
             const std::string & mapname,
             const std::string & name )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.1.10 textureToMapFile()

```
static void tgt::Map::textureToMapFile (
             FILE * fp,
             const js::json & map )  [static]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.4.2 Variable Documentation

4.4.2.1 ACTOR_PROPERTY

```
constexpr auto tgt::Map::ACTOR_PROPERTY = "actors"  [constexpr]
```

4.4.2.2 FONT_PROPERTY

```
constexpr auto tgt::Map::FONT_PROPERTY = "fonts"  [constexpr]
```

4.4.2.3 MAP_EXTENSION

```
constexpr auto tgt::Map::MAP_EXTENSION = ".tgr"  [constexpr]
```

4.4.2.4 MAP_PATH

```
const auto tgt::Map::MAP_PATH = fs::path(Util::RESOURCE_LOCATION).append(MAP_SUBFOLDER)
```

4.4.2.5 MAP_SUBFOLDER

```
constexpr auto tgt::Map::MAP_SUBFOLDER = "Maps"  [constexpr]
```

4.4.2.6 MATERIAL_PROPERTY

```
constexpr auto tgt::Map::MATERIAL_PROPERTY = "materials"  [constexpr]
```

4.4.2.7 TEXTURE_PROPERTY

```
constexpr auto tgt::Map::TEXTURE_PROPERTY = "textures"  [constexpr]
```

4.5 tgt::Material Namespace Reference

Functions

- const Result add (const std::string &name, const std::string &texture, const uint32_t color)
- const std::string list ()
- const Result remove (const std::string &name)

Variables

- constexpr auto MATERIAL_SUBFOLDER = "Materials"
- constexpr uint32_t DEFAULT_COLOR = 0xFFFFFFFF
- constexpr auto TEXTURE_PROPERTY = "texture"
- constexpr auto COLOR_PROPERTY = "color"
- constexpr std::array SUPPORTED_PROPERTIES = { TEXTURE_PROPERTY, COLOR_PROPERTY }
- const auto MATERIAL_PATH = fs::path(Util::RESOURCE_LOCATION).append(MATERIAL_SUBFOLDER)

4.5.1 Function Documentation

4.5.1.1 add()

```
const Result tgt::Material::add (
            const std::string & name,
            const std::string & texture,
            const uint32_t color )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.5.1.2 list()

```
const std::string tgt::Material::list ( )  [inline]
```

Here is the call graph for this function:

4.5.1.3 remove()

```
const Result tgt::Material::remove (
            const std::string & name )  [inline]
```

Here is the call graph for this function:

4.5.2 Variable Documentation

4.5.2.1 COLOR_PROPERTY

```
constexpr auto tgt::Material::COLOR_PROPERTY = "color"  [constexpr]
```

4.5.2.2 DEFAULT_COLOR

```
constexpr uint32_t tgt::Material::DEFAULT_COLOR = 0xFFFFFFFF  [constexpr]
```

4.5.2.3 MATERIAL_PATH

```
const auto tgt::Material::MATERIAL_PATH = fs::path(Util::RESOURCE_LOCATION).append(MATERIAL_SUBFOLDER)
```

4.5.2.4 MATERIAL_SUBFOLDER

```
constexpr auto tgt::Material::MATERIAL_SUBFOLDER = "Materials"  [constexpr]
```

4.5.2.5 SUPPORTED_PROPERTIES

```
constexpr std::array tgt::Material::SUPPORTED_PROPERTIES = { TEXTURE_PROPERTY, COLOR_PROPERTY
}  [constexpr]
```

4.5.2.6 TEXTURE_PROPERTY

```
constexpr auto tgt::Material::TEXTURE_PROPERTY = "texture"  [constexpr]
```

4.6 tgt::Model Namespace Reference

Functions

- static void recursive (const tinygltf::Model &model, const tinygltf::Node &node, const std::string &map=nullptr)
- const Result loadGltf (const std::string &path, const std::string &map)

4.6.1 Function Documentation

4.6.1.1 loadGltf()

```
const Result tgt::Model::loadGltf (
             const std::string & path,
             const std::string & map )
```

Here is the call graph for this function:

4.6.1.2 recursive()

```
static void tgt::Model::recursive (
             const tinygltf::Model & model,
             const tinygltf::Node & node,
             const std::string & map = nullptr )  [static]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.7 tgt::Pipe Namespace Reference

Enumerations

- enum CullMode { CullMode::NONE = 0, CullMode::FRONT_BIT = 1, CullMode::BACK_BIT = 2, CullMode::FRONT_AND_BACK = 3 }
- enum PolygonMode { PolygonMode::FILL = 0, PolygonMode::LINE = 1, PolygonMode::POINT = 2 }

Functions

- const Result create (const std::string &name, const CullMode cullmode, const PolygonMode polygonMode)
- const Result addShader (const std::string &name, const std::string &shadername)
- const Result removeShader (const std::string &name, const std::string &shadername)
- const Result remove (const std::string &name)
- const std::string list ()
- const bool checkDependent (const std::string &dependency)
- const Result compile ()

Variables

- constexpr CullMode CULLMODE_MIN = CullMode::NONE
- constexpr CullMode CULLMODE_MAX = CullMode::FRONT_AND_BACK
- constexpr PolygonMode POLYGONMODE_MIN = PolygonMode::FILL
- constexpr PolygonMode POLYGONMODE_MAX = PolygonMode::POINT
- constexpr auto PIPE_SUBFOLDER = "Pipes"
- constexpr auto CULLMODE_PROPERTY = "cullmode"
- constexpr auto POLYGONMODE_PROPERTY = "polygonmode"
- constexpr auto SHADER_PROPERTY = "shader"
- constexpr auto INPUTBINDINGS_PROPERTY = "inputs"
- constexpr std::array SUPPORTED_PROPERTIES = { CULLMODE_PROPERTY, POLYGONMODE_PROPERTY, SHADER_PROPERTY, INPUTBINDINGS_PROPERTY }
- const auto PIPE_PATH = fs::path(Util::RESOURCE_LOCATION).append(PIPE_SUBFOLDER)

4.7.1 Enumeration Type Documentation

4.7.1.1 CullMode

```
enum tgt::Pipe::CullMode  [strong]
```

Enumerator

NONE	
FRONT_BIT	
BACK_BIT	
FRONT_AND_BACK	

4.7.1.2 PolygonMode

```
enum tgt::Pipe::PolygonMode  [strong]
```

Enumerator

FILL	
LINE	
POINT	

4.7.2 Function Documentation

4.7.2.1 addShader()

```
const Result tgt::Pipe::addShader (
             const std::string & name,
             const std::string & shadername )
```

Here is the call graph for this function:

4.8 tgt::Sampler Namespace Reference

Classes

- struct SamplerInfo

Enumerations

- enum SamplerAddressMode {
  SamplerAddressMode::REPEAT = 0, SamplerAddressMode::MIRRORED_REPEAT = 1, SamplerAddressMode::CLAMP_TO_E
  = 2, SamplerAddressMode::CLAMP_TO_BORDER = 3,
  SamplerAddressMode::MIRROR_CLAMP_TO_EDGE = 4 }
- enum SamplerFilter { SamplerFilter::NEAREST = 0, SamplerFilter::LINEAR = 1 }

Functions

- const Result add (const std::string &name, const SamplerAddressMode umode, const SamplerAddressMode vmode, const SamplerFilter magfilter, const SamplerFilter minFilter)
- const Result add (const std::string &name, const SamplerAddressMode mode, const SamplerFilter filter)
- const Result remove (const std::string &name)
- const std::string list ()
- const Result change (const std::string &name, const std::string &key, const std::string &value)
- const Result write (FILE ∗file, const js::json &jsonarray)

Variables

- constexpr auto SAMPLER_SUBFOLDER = "Sampler"
- constexpr auto UMODE_PROPERTY = "u_address_mode"
- constexpr auto VMODE_PROPERTY = "v_address_mode"
- constexpr auto MAG_FILTER_PROPERTY = "magnification_filter"
- constexpr auto MIN_FILTER_PROPERTY = "minification_filter"
- const auto SAMPLER_PATH = fs::path(Util::RESOURCE_LOCATION).append(SAMPLER_SUBFOLDER)
- constexpr std::array SUPPORTED_PROPERTIES = {UMODE_PROPERTY, VMODE_PROPERTY, MAG_FILTER_PROPERTY, MIN_FILTER_PROPERTY}
- constexpr SamplerAddressMode SAMPLER_ADDRESS_MODE_MIN = SamplerAddressMode::REPEAT
- constexpr SamplerAddressMode SAMPLER_ADDRESS_MODE_MAX = SamplerAddressMode::MIRROR_CLAMP_TO_EDGE
- constexpr SamplerFilter SAMPLER_FILTER_MIN = SamplerFilter::NEAREST
- constexpr SamplerFilter SAMPLER_FILTER_MAX = SamplerFilter::LINEAR

4.8.1 Enumeration Type Documentation

4.8.1.1 SamplerAddressMode

```
enum tgt::Sampler::SamplerAddressMode  [strong]
```

Enumerator

REPEAT	
MIRRORED_REPEAT	
CLAMP_TO_EDGE	
CLAMP_TO_BORDER	
MIRROR_CLAMP_TO_EDGE	

4.8.1.2 SamplerFilter

```
enum tgt::Sampler::SamplerFilter  [strong]
```

Enumerator

NEAREST	
LINEAR	

4.8.2 Function Documentation

4.8.2.1 add() [1/2]

```
const Result tgt::Sampler::add (
             const std::string & name,
             const SamplerAddressMode mode,
             const SamplerFilter filter )  [inline]
```

Here is the call graph for this function:

4.8.2.2 add() [2/2]

```
const Result tgt::Sampler::add (
             const std::string & name,
             const SamplerAddressMode umode,
             const SamplerAddressMode vmode,
             const SamplerFilter magfilter,
             const SamplerFilter minFilter )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.8.2.3 change()

```
const Result tgt::Sampler::change (
             const std::string & name,
             const std::string & key,
             const std::string & value )  [inline]
```

Here is the call graph for this function:

4.8.2.4 list()

```
const std::string tgt::Sampler::list ( )  [inline]
```

Here is the call graph for this function:

4.8.2.5 remove()

```
const Result tgt::Sampler::remove (
             const std::string & name )  [inline]
```

Here is the call graph for this function:

4.8.2.6 write()

```
const Result tgt::Sampler::write (
             FILE * file,
             const js::json & jsonarray )  [inline]
```

Here is the call graph for this function:

4.8.3 Variable Documentation

4.8.3.1 MAG_FILTER_PROPERTY

```
constexpr auto tgt::Sampler::MAG_FILTER_PROPERTY = "magnification_filter"  [constexpr]
```

4.8.3.2 MIN_FILTER_PROPERTY

```
constexpr auto tgt::Sampler::MIN_FILTER_PROPERTY = "minification_filter"  [constexpr]
```

4.8.3.3 SAMPLER_ADDRESS_MODE_MAX

```
constexpr SamplerAddressMode tgt::Sampler::SAMPLER_ADDRESS_MODE_MAX = SamplerAddressMode::MIRROR_CLAMP_TO_EDGE
[constexpr]
```

4.8.3.4 SAMPLER_ADDRESS_MODE_MIN

```
constexpr SamplerAddressMode tgt::Sampler::SAMPLER_ADDRESS_MODE_MIN = SamplerAddressMode::RE←↩
PEAT  [constexpr]
```

4.8.3.5 SAMPLER_FILTER_MAX

```
constexpr SamplerFilter tgt::Sampler::SAMPLER_FILTER_MAX = SamplerFilter::LINEAR  [constexpr]
```

4.8.3.6 SAMPLER_FILTER_MIN

```
constexpr SamplerFilter tgt::Sampler::SAMPLER_FILTER_MIN = SamplerFilter::NEAREST  [constexpr]
```

4.8.3.7 SAMPLER_PATH

```
const auto tgt::Sampler::SAMPLER_PATH = fs::path(Util::RESOURCE_LOCATION).append(SAMPLER_SUBFOLDER)
```

4.8.3.8 SAMPLER_SUBFOLDER

```
constexpr auto tgt::Sampler::SAMPLER_SUBFOLDER = "Sampler"  [constexpr]
```

4.8.3.9 SUPPORTED_PROPERTIES

```
constexpr std::array tgt::Sampler::SUPPORTED_PROPERTIES = {UMODE_PROPERTY, VMODE_PROPERTY,
MAG_FILTER_PROPERTY, MIN_FILTER_PROPERTY}  [constexpr]
```

4.8.3.10 UMODE_PROPERTY

```
constexpr auto tgt::Sampler::UMODE_PROPERTY = "u_address_mode"  [constexpr]
```

4.8.3.11 VMODE_PROPERTY

```
constexpr auto tgt::Sampler::VMODE_PROPERTY = "v_address_mode"  [constexpr]
```

4.9 tgt::Shader Namespace Reference

Namespaces

- ShaderInput

Typedefs

- using ShaderType = EShLanguage

Functions

- const Result add (const std::string &name, ShaderType type)
- const Result remove (const std::string &name)
- const std::string list ()

Variables

- constexpr auto SHADER_EXTENSION = ".glsl"
- constexpr auto SHADER_SUBFOLDER = "Shaders"
- constexpr auto SHADER_TYPE_PROPERTY = "shaderType"
- constexpr std::array SUPPORTED_PROPERTIES = { SHADER_TYPE_PROPERTY }
- const auto SHADER_PATH = fs::path(Util::RESOURCE_LOCATION).append(SHADER_SUBFOLDER)

4.9.1 Typedef Documentation

4.9.1.1 ShaderType

```
using tgt::Shader::ShaderType = typedef EShLanguage
```

4.9.2 Function Documentation

4.9.2.1 add()

```
const Result tgt::Shader::add (
            const std::string & name,
            ShaderType type )
```

Here is the call graph for this function:

4.9.2.2 list()

```
const std::string tgt::Shader::list ( )  [inline]
```

Here is the call graph for this function:

4.9.2.3 remove()

```
const Result tgt::Shader::remove (
             const std::string & name )
```

Here is the call graph for this function:

4.9.3 Variable Documentation

4.9.3.1 SHADER_EXTENSION

```
constexpr auto tgt::Shader::SHADER_EXTENSION = ".glsl"  [constexpr]
```

4.9.3.2 SHADER_PATH

```
const auto tgt::Shader::SHADER_PATH = fs::path(Util::RESOURCE_LOCATION).append(SHADER_SUBFOLDER)
```

4.9.3.3 SHADER_SUBFOLDER

```
constexpr auto tgt::Shader::SHADER_SUBFOLDER = "Shaders"  [constexpr]
```

4.9.3.4 SHADER_TYPE_PROPERTY

```
constexpr auto tgt::Shader::SHADER_TYPE_PROPERTY = "shaderType"  [constexpr]
```

4.9.3.5 SUPPORTED_PROPERTIES

```
constexpr std::array tgt::Shader::SUPPORTED_PROPERTIES = { SHADER_TYPE_PROPERTY }  [constexpr]
```

4.10 tgt::Shader::ShaderInput Namespace Reference

Typedefs

- using ShaderInputStride = uint32_t

Enumerations

- enum ShaderInputType { ShaderInputType::POSITION, ShaderInputType::POSITION_UV, ShaderInputType::POSITION_NORM
ShaderInputType::POSITION_NORMAL_UV }

Variables

- constexpr ShaderInputStride POS = sizeof(float) ∗ 3
- constexpr ShaderInputStride POS_TEX = POS + sizeof(float) ∗ 2
- constexpr ShaderInputStride POS_NOR = POS + sizeof(float) ∗ 3
- constexpr ShaderInputStride POS_NOR_TEX = POS_NOR + sizeof(float) ∗ 2

4.10.1 Typedef Documentation

4.10.1.1 ShaderInputStride

```
using tgt::Shader::ShaderInput::ShaderInputStride = typedef uint32_t
```

4.10.2 Enumeration Type Documentation

4.10.2.1 ShaderInputType

```
enum tgt::Shader::ShaderInput::ShaderInputType  [strong]
```

Enumerator

POSITION	
POSITION_UV	
POSITION_NORMAL	
POSITION_NORMAL_UV	

4.10.3 Variable Documentation

4.10.3.1 POS

```
constexpr ShaderInputStride tgt::Shader::ShaderInput::POS = sizeof(float) ∗ 3  [constexpr]
```

4.10.3.2 POS_NOR

```
constexpr ShaderInputStride tgt::Shader::ShaderInput::POS_NOR = POS + sizeof(float) * 3  [constexpr]
```

4.10.3.3 POS_NOR_TEX

```
constexpr ShaderInputStride tgt::Shader::ShaderInput::POS_NOR_TEX = POS_NOR + sizeof(float) *
2  [constexpr]
```

4.10.3.4 POS_TEX

```
constexpr ShaderInputStride tgt::Shader::ShaderInput::POS_TEX = POS + sizeof(float) * 2  [constexpr]
```

4.11 tgt::Texture Namespace Reference

Functions

- const Result add (const char *path)
- const Result add (const std::string &path)
- const Result remove (const char *name)
- const Result remove (const std::string &name)
- const std::string list ()

Variables

- constexpr auto TEXTURE_EXTENSION = ".tgx"
- constexpr auto TEXTURE_SUBFOLDER = "Textures"
- const auto TEXTURE_PATH = fs::path(Util::RESOURCE_LOCATION).append(TEXTURE_SUBFOLDER)

4.11.1 Function Documentation

4.11.1.1 add() [1/2]

```
const Result tgt::Texture::add (
            const char * path )
```

Here is the call graph for this function: Here is the caller graph for this function:

4.11.1.2 add() [2/2]

```
const Result tgt::Texture::add (
             const std::string & path )
```

Here is the call graph for this function:

4.11.1.3 list()

```
const std::string tgt::Texture::list ( )
```

Here is the call graph for this function:

4.11.1.4 remove() [1/2]

```
const Result tgt::Texture::remove (
             const char * name )
```

4.11.1.5 remove() [2/2]

```
const Result tgt::Texture::remove (
             const std::string & name )
```

Here is the call graph for this function:

4.11.2 Variable Documentation

4.11.2.1 TEXTURE_EXTENSION

```
constexpr auto tgt::Texture::TEXTURE_EXTENSION = ".tgx"  [constexpr]
```

4.11.2.2 TEXTURE_PATH

```
const auto tgt::Texture::TEXTURE_PATH = fs::path(Util::RESOURCE_LOCATION).append(TEXTURE_SUBFOLDER)
```

4.11.2.3 TEXTURE_SUBFOLDER

```
constexpr auto tgt::Texture::TEXTURE_SUBFOLDER = "Textures"  [constexpr]
```

4.12 tgt::Util Namespace Reference

Classes

- class scope_exit

Functions

- template<class T , typename = std::enable_if_t<std::is_invocable_r_v<const Result, T, js::json&>>>
  const Result jsonUpdatet (const fs::path &path, T lambda)
- template<class T , class U , size_t S, typename = std::enable_if_t<_validJson<T> && _validString<U>>>
  const Result change (const fs::path &path, const std::string &key, const T &value, const std::array< U, S > &supported)
- template<class U , typename = std::enable_if_t<_validString<U> || std::is_null_pointer_v<U>>>
  const fs::path getResource (fs::path resource, const std::string &name, const U &extension)
- const fs::path getResource (const fs::path &resource, const std::string &name)
- const uint8_t ∗ readFile (const std::string &name, size_t ∗sizeptr=nullptr)
- template<class U , typename = std::enable_if_t<std::is_invocable_r_v<bool, U, const fs::path&>>>
  const std::string collect (const fs::path &path, U lambda)
- template<class U , typename = std::enable_if_t<std::is_invocable_r_v<bool, U, const fs::path&>>>
  const bool find (const fs::path &path, const U lambda)
- template<class T , typename = std::enable_if_t<std::is_invocable_v<T, const js::json&>>>
  const Result writeToFile (FILE ∗file, const js::json &jsonarray, T lambda)
- template<class T , typename = std::enable_if_t<std::is_invocable_r_v<bool, T, const std::string&> || std::is_invocable_r_v<bool, T, const fs::path&> || std::is_null_pointer_v<T>>>
  const Result remove (const fs::path &parentpath, const std::string &name, const std::string &filter, T lambda)
- template<class T , typename = std::enable_if_t<std::is_invocable_r_v<bool, T, const std::string&> || std::is_invocable_r_v<bool, T, const fs::path&>>>
  const Result remove (const fs::path &path, const std::string &name, T lambda)
- const Result remove (const fs::path &path, const std::string &name)

Variables

- constexpr auto RESOURCE_LOCATION = "Resource"
- constexpr auto JSON = ".json"
- constexpr auto JSON_FILTER = [ ](fs::path path) { return path.extension() == JSON; }
- template<class T , class... U>
  constexpr bool _isAnyOf = std::disjunction_v<std::is_same<T, U>...>
- template<class T >
  constexpr bool _validString = _isAnyOf<T, std::string, const std::string, char∗, const char∗, char∗ const, const char∗ const>
- template<class T >
  constexpr bool _validPath = _isAnyOf<T, fs::path, const fs::path>
- template<class T >
  constexpr bool _validPathString = _validPath<T> || _validString<T>
- template<class T >
  constexpr bool _validJson = std::is_arithmetic_v<T> || _validString<T>

4.12.1 Function Documentation

4.12.1.1 change()

```
template<class T , class U , size_t S, typename = std::enable_if_t<_validJson<T> && _valid↩
String<U>>>
const Result tgt::Util::change (
            const fs::path & path,
            const std::string & key,
            const T & value,
            const std::array< U, S > & supported )  [inline]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.12.1.2 collect()

```
template<class U , typename = std::enable_if_t<std::is_invocable_r_v<bool, U, const fs↩
::path&>>>
const std::string tgt::Util::collect (
            const fs::path & path,
            U lambda )  [inline]
```

Here is the caller graph for this function:

4.12.1.3 find()

```
template<class U , typename = std::enable_if_t<std::is_invocable_r_v<bool, U, const fs↩
::path&>>>
const bool tgt::Util::find (
            const fs::path & path,
            const U lambda )  [inline]
```

Here is the caller graph for this function:

4.12.1.4 getResource() [1/2]

```
const fs::path tgt::Util::getResource (
            const fs::path & resource,
            const std::string & name )  [inline]
```

Here is the call graph for this function:

4.12.1.5 getResource() [2/2]

```
template<class U , typename = std::enable_if_t<_validString<U> || std::is_null_pointer_v<↩
U>>>
const fs::path tgt::Util::getResource (
            fs::path resource,
            const std::string & name,
            const U & extension )  [inline]
```

Here is the caller graph for this function:

4.12.1.6 jsonUpdatet()

```
template<class T , typename = std::enable_if_t<std::is_invocable_r_v<const Result, T, js↩
::json&>>>
const Result tgt::Util::jsonUpdatet (
             const fs::path & path,
             T lambda )  [inline]
```

Here is the caller graph for this function:

4.12.1.7 readFile()

```
const uint8_t* tgt::Util::readFile (
             const std::string & name,
             size_t * sizeptr = nullptr )  [inline]
```

Here is the caller graph for this function:

4.12.1.8 remove() [1/3]

```
template<class T , typename = std::enable_if_t<std::is_invocable_r_v<bool, T, const std↩
::string&> || std::is_invocable_r_v<bool, T, const fs::path&> || std::is_null_pointer_v<T>>>
const Result tgt::Util::remove (
             const fs::path & parentpath,
             const std::string & name,
             const std::string & filter,
             T lambda )  [inline]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.12.1.9 remove() [2/3]

```
const Result tgt::Util::remove (
             const fs::path & path,
             const std::string & name )  [inline]
```

Here is the call graph for this function: Here is the caller graph for this function:

4.12.1.10 remove() [3/3]

```
template<class T , typename = std::enable_if_t<std::is_invocable_r_v<bool, T, const std↩
::string&> || std::is_invocable_r_v<bool, T, const fs::path&>>>
const Result tgt::Util::remove (
             const fs::path & path,
             const std::string & name,
             T lambda )  [inline]
```

Here is the call graph for this function:

4.12.1.11 writeToFile()

```
template<class T , typename = std::enable_if_t<std::is_invocable_v<T, const js::json&>>>
const Result tgt::Util::writeToFile (
             FILE * file,
             const js::json & jsonarray,
             T lambda )  [inline]
```

Here is the caller graph for this function:

4.12.2 Variable Documentation

4.12.2.1 _isAnyOf

```
template<class T , class... U>
constexpr bool tgt::Util::_isAnyOf = std::disjunction_v<std::is_same<T, U>...>  [constexpr]
```

4.12.2.2 _validJson

```
template<class T >
constexpr bool tgt::Util::_validJson = std::is_arithmetic_v<T> || _validString<T>  [constexpr]
```

4.12.2.3 _validPath

```
template<class T >
constexpr bool tgt::Util::_validPath = _isAnyOf<T, fs::path, const fs::path>  [constexpr]
```

4.12.2.4 _validPathString

```
template<class T >
constexpr bool tgt::Util::_validPathString = _validPath<T> || _validString<T>  [constexpr]
```

4.12.2.5 _validString

```
template<class T >
constexpr bool tgt::Util::_validString = _isAnyOf<T, std::string, const std::string, char*,
const char*, char* const, const char* const>  [constexpr]
```

4.12.2.6 JSON

```
constexpr auto tgt::Util::JSON = ".json"  [constexpr]
```

4.12.2.7 JSON_FILTER

```
constexpr auto tgt::Util::JSON_FILTER = [](fs::path path) { return path.extension() == JSON; }
[constexpr]
```

4.12.2.8 RESOURCE_LOCATION

```
constexpr auto tgt::Util::RESOURCE_LOCATION = "Resource"  [constexpr]
```

Chapter 5

Class Documentation

5.1 tgt::Actor::ActorData Struct Reference

```
#include <Actor.hpp>
```

Public Attributes

- float matrix [16]
- uint32_t animationIndex
- uint32_t transformIndex
- uint32_t material
- uint32_t layer
- uint32_t instanceSize
- uint32_t instanceOffset
- uint32_t indexDrawCount
- uint32_t vertexCount

5.1.1 Member Data Documentation

5.1.1.1 animationIndex

```
uint32_t tgt::Actor::ActorData::animationIndex
```

5.1.1.2 indexDrawCount

```
uint32_t tgt::Actor::ActorData::indexDrawCount
```

5.1.1.3 instanceOffset

```
uint32_t tgt::Actor::ActorData::instanceOffset
```

5.1.1.4 instanceSize

```
uint32_t tgt::Actor::ActorData::instanceSize
```

5.1.1.5 layer

```
uint32_t tgt::Actor::ActorData::layer
```

5.1.1.6 material

```
uint32_t tgt::Actor::ActorData::material
```

5.1.1.7 matrix

```
float tgt::Actor::ActorData::matrix[16]
```

5.1.1.8 transformIndex

```
uint32_t tgt::Actor::ActorData::transformIndex
```

5.1.1.9 vertexCount

```
uint32_t tgt::Actor::ActorData::vertexCount
```

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

- TGTools/TGTools/public/Actor.hpp

5.2 tgt::Sampler::SamplerInfo Struct Reference

```
#include <Sampler.hpp>
```

Public Attributes

- uint8_t umode
- uint8_t vmode
- uint8_t magfilter
- uint8_t minfilter

5.2.1 Member Data Documentation

5.2.1.1 magfilter

```
uint8_t tgt::Sampler::SamplerInfo::magfilter
```

5.2.1.2 minfilter

```
uint8_t tgt::Sampler::SamplerInfo::minfilter
```

5.2.1.3 umode

```
uint8_t tgt::Sampler::SamplerInfo::umode
```

5.2.1.4 vmode

```
uint8_t tgt::Sampler::SamplerInfo::vmode
```

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

- TGTools/TGTools/public/Sampler.hpp

5.3 tgt::Util::scope_exit< T, typename > Class Template Reference

```
#include <Util.hpp>
```

Public Member Functions

- scope_exit (T x)
- ∼scope_exit ()

5.3.1 Constructor & Destructor Documentation

5.3.1.1 scope_exit()

```
template<typename T , typename = std::enable_if_t<std::is_invocable_v<T>>>
tgt::Util::scope_exit< T, typename >::scope_exit (
            T x ) [inline]
```

5.3.1.2 ∼scope_exit()

```
template<typename T , typename = std::enable_if_t<std::is_invocable_v<T>>>
tgt::Util::scope_exit< T, typename >::∼scope_exit ( ) [inline]
```

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

- TGTools/TGTools/public/Util.hpp

5.4 tgt::Font::TGFHeader Struct Reference

Public Attributes

- uint32_t version = TGF_HEADER_VERSION
- uint32_t textureindex = 0
- uint32_t startindex
- uint32_t count

5.4.1 Member Data Documentation

5.4.1.1 count

```
uint32_t tgt::Font::TGFHeader::count
```

5.4.1.2 startindex

```
uint32_t tgt::Font::TGFHeader::startindex
```

5.4.1.3 textureindex

```
uint32_t tgt::Font::TGFHeader::textureindex = 0
```

5.4.1.4 version

```
uint32_t tgt::Font::TGFHeader::version = TGF_HEADER_VERSION
```

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

- TGTools/TGTools/private/Font.cpp

Chapter 6

File Documentation

6.1 TGTools/TGTools/CMakeLists.txt File Reference

6.2 TGTools/TGTools/private/Actor.cpp File Reference

```
#include "../public/Actor.hpp"
#include "../public/json.hpp"
#include "../public/Material.hpp"
```

Include dependency graph for Actor.cpp:

Namespaces

- tgt
- tgt::Actor

Functions

- const Result tgt::Actor::add (const std::string &name, const std::string &material)
- const Result tgt::Actor::_dataHeader (const fs::path &name, ActorData ∗data)

6.3 TGTools/TGTools/private/Font.cpp File Reference

```
#include "../public/Font.hpp"
#include "../public/json.hpp"
#include "../../../stb/stb_rect_pack.h"
#include "../../../stb/stb_truetype.h"
#include "../../../stb/stb_image_write.h"
#include <iostream>
```

Include dependency graph for Font.cpp:

Classes

- struct tgt::Font::TGFHeader

Namespaces

- tgt
- tgt::Font

Macros

- #define STB_TRUETYPE_IMPLEMENTATION
- #define STB_RECT_PACK_IMPLEMENTATION
- #define STB_IMAGE_WRITE_IMPLEMENTATION

Functions

- static const Result tgt::Font::generatefontbitmap (const fs::path &font, const std::string &resourceLocation, const std::string &texture)
- const Result tgt::Font::add (const std::string &path)

Variables

- constexpr auto tgt::Font::TGF_HEADER_VERSION = 1

6.3.1 Macro Definition Documentation

6.3.1.1 STB_IMAGE_WRITE_IMPLEMENTATION

```
#define STB_IMAGE_WRITE_IMPLEMENTATION
```

6.3.1.2 STB_RECT_PACK_IMPLEMENTATION

```
#define STB_RECT_PACK_IMPLEMENTATION
```

6.3.1.3 STB_TRUETYPE_IMPLEMENTATION

```
#define STB_TRUETYPE_IMPLEMENTATION
```

6.4 TGTools/TGTools/private/Map.cpp File Reference

```
#include "../public/Map.hpp"
#include "../public/json.hpp"
#include <fstream>
#include "../public/Actor.hpp"
#include "../public/Font.hpp"
#include "../public/Texture.hpp"
#include "../public/Material.hpp"
#include <algorithm>
```

Include dependency graph for Map.cpp:

Namespaces

- tgt
- tgt::Map

Macros

- #define CHECK_RESULT(statement) result = statement; if(result != Result::SUCCESS) return result
- #define WRITE_INT(fp, dt) fwrite(&dt, sizeof(uint32_t), 1, fp)
- #define WRITE_CINT(fp, num) check = num; fwrite(&check, sizeof(uint32_t), 1, fp)
- #define WRITE_CHECK(fp) constexpr auto WRITE_CINT(fp, UINT32_MAX);
- #define WRITE_SIZE(fp) WRITE_INT(fp, size)

Functions

- const Result tgt::Map::create (const std::string &mapname)
- static void tgt::Map::textureToMapFile (FILE ∗fp, const js::json &map)
- static Result tgt::Map::materialToMapFile (FILE ∗fp, const js::json &map)
- static Result tgt::Map::actorToMapFile (FILE ∗fp, const js::json &map)
- const Result tgt::Map::make (const std::string &mapname)
- const Result tgt::Map::add (const std::string &mapname, const std::string &name)
- const Result tgt::Map::remove (const std::string &mapname, const std::string &name)

6.4.1 Macro Definition Documentation

6.4.1.1 CHECK_RESULT

```
#define CHECK_RESULT(
            statement ) result = statement; if(result != Result::SUCCESS) return result
```

6.4.1.2 WRITE_CHECK

```
#define WRITE_CHECK(
              fp ) constexpr auto WRITE_CINT(fp, UINT32_MAX);
```

6.4.1.3 WRITE_CINT

```
#define WRITE_CINT(
              fp,
              num ) check = num; fwrite(&check, sizeof(uint32_t), 1, fp)
```

6.4.1.4 WRITE_INT

```
#define WRITE_INT(
              fp,
              dt ) fwrite(&dt, sizeof(uint32_t), 1, fp)
```

6.4.1.5 WRITE_SIZE

```
#define WRITE_SIZE(
              fp ) WRITE_INT(fp, size)
```

6.5 TGTools/TGTools/private/Material.cpp File Reference

```
#include "../public/Material.hpp"
#include "../public/Texture.hpp"
```

Include dependency graph for Material.cpp:

Namespaces

- tgt
- tgt::Material

Functions

- const Result tgt::Material::add (const std::string &name, const std::string &texture, const uint32_t color)

6.6 TGTools/TGTools/private/Model.cpp File Reference

```
#include "../public/Model.hpp"
#include "../public/Texture.hpp"
#include "../public/Material.hpp"
#include "../public/Actor.hpp"
#include "../../../tinygltf/tiny_gltf.h"
```

Include dependency graph for Model.cpp:

Namespaces

- tgt
- tgt::Model

Macros

- #define TINYGLTF_IMPLEMENTATION
- #define TINYGLTF_USE_CPP14
- #define STB_IMAGE_IMPLEMENTATION
- #define TINYGLTF_NO_INCLUDE_JSON
- #define getName(text) text.name.empty() ? fs::path(text.uri).stem().string() : text.name

Functions

- static void tgt::Model::recursive (const tinygltf::Model &model, const tinygltf::Node &node, const std::string &map=nullptr)
- const Result tgt::Model::loadGltf (const std::string &path, const std::string &map)

6.6.1 Macro Definition Documentation

6.6.1.1 getName

```
#define getName(
            text ) text.name.empty() ?  fs::path(text.uri).stem().string() :  text.name
```

6.6.1.2 STB_IMAGE_IMPLEMENTATION

```
#define STB_IMAGE_IMPLEMENTATION
```

6.6.1.3 TINYGLTF_IMPLEMENTATION

```
#define TINYGLTF_IMPLEMENTATION
```

6.6.1.4 TINYGLTF_NO_INCLUDE_JSON

```
#define TINYGLTF_NO_INCLUDE_JSON
```

6.6.1.5 TINYGLTF_USE_CPP14

```
#define TINYGLTF_USE_CPP14
```

6.7 TGTools/TGTools/private/Pipe.cpp File Reference

```
#include "../public/Pipe.hpp"
#include "../public/Shader.hpp"
```

Include dependency graph for Pipe.cpp:

Namespaces

- tgt
- tgt::Pipe

Functions

- const Result tgt::Pipe::create (const std::string &name, const CullMode cullmode, const PolygonMode polygonMode)
- const Result tgt::Pipe::addShader (const std::string &name, const std::string &shadername)
- const Result tgt::Pipe::removeShader (const std::string &name, const std::string &shadername)

6.8 TGTools/TGTools/private/Sampler.cpp File Reference

```
#include "../public/Sampler.hpp"
```

Include dependency graph for Sampler.cpp:

Namespaces

- tgt
- tgt::Sampler

Functions

- const Result tgt::Sampler::add (const std::string &name, const SamplerAddressMode umode, const SamplerAddressMode vmode, const SamplerFilter magfilter, const SamplerFilter minFilter)

6.9 TGTools/TGTools/private/Shader.cpp File Reference

```
#include "../public/Shader.hpp"
#include "../public/json.hpp"
#include "../public/Pipe.hpp"
```

Include dependency graph for Shader.cpp:

Namespaces

- tgt
- tgt::Shader

Functions

- const Result tgt::Shader::add (const std::string &name, ShaderType type)
- const Result tgt::Shader::remove (const std::string &name)

6.10 TGTools/TGTools/private/Texture.cpp File Reference

```
#include "../public/Texture.hpp"
#include "../public/Material.hpp"
#include "../public/json.hpp"
```

Include dependency graph for Texture.cpp:

Namespaces

- tgt
- tgt::Texture

Functions

- const Result tgt::Texture::add (const char ∗path)
- const Result tgt::Texture::add (const std::string &path)
- const Result tgt::Texture::remove (const char ∗name)
- const Result tgt::Texture::remove (const std::string &name)
- const std::string tgt::Texture::list ()

6.11 TGTools/TGTools/public/Actor.hpp File Reference

```
#include "Result.hpp"
#include <string>
#include "Util.hpp"
#include <array>
#include "../public/Map.hpp"
```

Include dependency graph for Actor.hpp: This graph shows which files directly or indirectly include this file:

Classes

- struct tgt::Actor::ActorData

Namespaces

- tgt
- tgt::Actor

Functions

- const Result tgt::Actor::add (const std::string &name, const std::string &material)
- const Result tgt::Actor::remove (const std::string &name)
- const std::string tgt::Actor::list ()
- template<class T , class U , class V , typename = std::enable_if_t<Util::_validJson<T> && Util::_validString<V> && Util::_validString<U>>>
  const Result tgt::Actor::change (V actorname, U key, T value)
- const Result tgt::Actor::_dataHeader (const fs::path &name, ActorData ∗data)
- void tgt::Actor::setData (const void ∗data, const uint32_t byteSize, const std::string &name, bool append=false)
- const Result tgt::Actor::getData (const void ∗∗data, const fs::path &name, size_t ∗ptr=nullptr)

Variables

- constexpr auto tgt::Actor::ACTOR_SUBFOLDER = "Actors"
- constexpr auto tgt::Actor::ACTOR_VERTEX_EXTENSION = "tgvmdl"
- constexpr auto tgt::Actor::ACTOR_INDEX_EXTENSION = "tgimdl"
- constexpr auto tgt::Actor::MATRIX_PROPERTY = "matrix"
- constexpr auto tgt::Actor::ANIMATION_PROPERTY = "animationIndex"
- constexpr auto tgt::Actor::DYNAMIK_TRANSFORM_PROPERTY = "transformIndex"
- constexpr auto tgt::Actor::MATERIAL_PROPERTY = "material"
- constexpr auto tgt::Actor::LAYER_PROPERTY = "layer"
- constexpr auto tgt::Actor::INDEX_COUNT = "indexDrawCount"
- constexpr auto tgt::Actor::VERTEX_COUNT = "vertexCount"
- constexpr std::array tgt::Actor::SUPPORTED_PROPERTIES
- const auto tgt::Actor::ACTOR_PATH = fs::path(Util::RESOURCE_LOCATION).append(ACTOR_SUBFOLDER)

6.12 TGTools/TGTools/public/Font.hpp File Reference

```
#include "Result.hpp"
#include <string>
#include "Util.hpp"
#include "Texture.hpp"
```

Include dependency graph for Font.hpp: This graph shows which files directly or indirectly include this file:

Namespaces

- tgt
- tgt::Font

Functions

- const Result tgt::Font::add (const std::string &path)
- const Result tgt::Font::remove (const std::string &name)
- const std::string tgt::Font::list ()

Variables

- constexpr auto tgt::Font::FONT_SUBFOLDER = "Font"
- constexpr auto tgt::Font::FONT_EXTENSION = ".tgf"
- const auto tgt::Font::FONT_PATH = fs::path(Util::RESOURCE_LOCATION).append(FONT_SUBFOLDER)

6.13 TGTools/TGTools/public/Map.hpp File Reference

```
#include "Result.hpp"
#include <string>
#include "Util.hpp"
```

Include dependency graph for Map.hpp: This graph shows which files directly or indirectly include this file:

Namespaces

- tgt
- tgt::Map

Functions

- const Result tgt::Map::create (const std::string &mapname)
- const Result tgt::Map::remove (const std::string &mapname)
- const std::string tgt::Map::list ()
- const Result tgt::Map::make (const std::string &mapname)
- const Result tgt::Map::add (const std::string &mapname, const std::string &name)
- const Result tgt::Map::remove (const std::string &mapname, const std::string &name)
- const bool tgt::Map::checkDependent (const std::string &dependency)

Variables

- constexpr auto tgt::Map::MAP_SUBFOLDER = "Maps"
- constexpr auto tgt::Map::MAP_EXTENSION = ".tgr"
- constexpr auto tgt::Map::TEXTURE_PROPERTY = "textures"
- constexpr auto tgt::Map::ACTOR_PROPERTY = "actors"
- constexpr auto tgt::Map::MATERIAL_PROPERTY = "materials"
- constexpr auto tgt::Map::FONT_PROPERTY = "fonts"
- const auto tgt::Map::MAP_PATH = fs::path(Util::RESOURCE_LOCATION).append(MAP_SUBFOLDER)

6.14 TGTools/TGTools/public/Material.hpp File Reference

```
#include "Result.hpp"
#include <string>
#include "Util.hpp"
#include <array>
```

Include dependency graph for Material.hpp: This graph shows which files directly or indirectly include this file:

Namespaces

- tgt
- tgt::Material

Functions

- const Result tgt::Material::add (const std::string &name, const std::string &texture, const uint32_t color)
- const std::string tgt::Material::list ()
- const Result tgt::Material::remove (const std::string &name)

Variables

- constexpr auto tgt::Material::MATERIAL_SUBFOLDER = "Materials"
- constexpr uint32_t tgt::Material::DEFAULT_COLOR = 0xFFFFFFFF
- constexpr auto tgt::Material::TEXTURE_PROPERTY = "texture"
- constexpr auto tgt::Material::COLOR_PROPERTY = "color"
- constexpr std::array tgt::Material::SUPPORTED_PROPERTIES = { TEXTURE_PROPERTY, COLOR_PR↩ OPERTY }
- const auto tgt::Material::MATERIAL_PATH = fs::path(Util::RESOURCE_LOCATION).append(MATERIAL_↩ SUBFOLDER)

6.15 TGTools/TGTools/public/Model.hpp File Reference

```
#include "../public/Result.hpp"
#include "../public/Util.hpp"
```

Include dependency graph for Model.hpp: This graph shows which files directly or indirectly include this file:

Namespaces

- tgt
- tgt::Model

Functions

- const Result tgt::Model::loadGltf (const std::string &path, const std::string &map)

6.16 TGTools/TGTools/public/Pipe.hpp File Reference

```
#include "Util.hpp"
```

Include dependency graph for Pipe.hpp: This graph shows which files directly or indirectly include this file:

Namespaces

- tgt
- tgt::Pipe

Enumerations

- enum tgt::Pipe::CullMode { tgt::Pipe::CullMode::NONE = 0, tgt::Pipe::CullMode::FRONT_BIT = 1, tgt::Pipe::CullMode::BACK_BIT = 2, tgt::Pipe::CullMode::FRONT_AND_BACK = 3 }
- enum tgt::Pipe::PolygonMode { tgt::Pipe::PolygonMode::FILL = 0, tgt::Pipe::PolygonMode::LINE = 1, tgt::Pipe::PolygonMode::POINT = 2 }

Functions

- const Result tgt::Pipe::create (const std::string &name, const CullMode cullmode, const PolygonMode polygonMode)
- const Result tgt::Pipe::remove (const std::string &name)
- const std::string tgt::Pipe::list ()
- const Result tgt::Pipe::addShader (const std::string &name, const std::string &shadername)
- const Result tgt::Pipe::removeShader (const std::string &name, const std::string &shadername)
- const bool tgt::Pipe::checkDependent (const std::string &dependency)
- const Result tgt::Pipe::compile ()

Variables

- constexpr CullMode tgt::Pipe::CULLMODE_MIN = CullMode::NONE
- constexpr CullMode tgt::Pipe::CULLMODE_MAX = CullMode::FRONT_AND_BACK
- constexpr PolygonMode tgt::Pipe::POLYGONMODE_MIN = PolygonMode::FILL
- constexpr PolygonMode tgt::Pipe::POLYGONMODE_MAX = PolygonMode::POINT
- constexpr auto tgt::Pipe::PIPE_SUBFOLDER = "Pipes"
- constexpr auto tgt::Pipe::CULLMODE_PROPERTY = "cullmode"
- constexpr auto tgt::Pipe::POLYGONMODE_PROPERTY = "polygonmode"
- constexpr auto tgt::Pipe::SHADER_PROPERTY = "shader"
- constexpr auto tgt::Pipe::INPUTBINDINGS_PROPERTY = "inputs"
- constexpr std::array tgt::Pipe::SUPPORTED_PROPERTIES = { CULLMODE_PROPERTY, POLYGONMODE_PROPERTY, SHADER_PROPERTY, INPUTBINDINGS_PROPERTY }
- const auto tgt::Pipe::PIPE_PATH = fs::path(Util::RESOURCE_LOCATION).append(PIPE_SUBFOLDER)

6.17 TGTools/TGTools/public/Result.hpp File Reference

This graph shows which files directly or indirectly include this file:

Enumerations

- enum Result {
  Result::SUCCESS, Result::BAD_ARGUMENTS, Result::ALREADY_EXISTS, Result::DOES_NOT_EXIST,
  Result::DEPENDENT, Result::GENERAL, Result::BAD_STRING, Result::UNSUPPORTED }

6.17.1 Enumeration Type Documentation

6.17.1.1 Result

```
enum Result  [strong]
```

Enumerator

SUCCESS	
BAD_ARGUMENTS	
ALREADY_EXISTS	
DOES_NOT_EXIST	
DEPENDENT	
GENERAL	
BAD_STRING	
UNSUPPORTED	

6.18 TGTools/TGTools/public/Sampler.hpp File Reference

```
#include "Util.hpp"
#include <string>
```

Include dependency graph for Sampler.hpp: This graph shows which files directly or indirectly include this file:

Classes

- struct tgt::Sampler::SamplerInfo

Namespaces

- tgt
- tgt::Sampler

Enumerations

- enum tgt::Sampler::SamplerAddressMode { tgt::Sampler::SamplerAddressMode::REPEAT = 0, tgt::Sampler::SamplerAddressMode::MIRRORED_REPEAT = 1, tgt::Sampler::SamplerAddressMode::CLAMP_TO_EDGE = 2, tgt::Sampler::SamplerAddressMode::CLAMP_TO_BORDER = 3, tgt::Sampler::SamplerAddressMode::MIRROR_CLAMP_TO_EDGE = 4 }
- enum tgt::Sampler::SamplerFilter { tgt::Sampler::SamplerFilter::NEAREST = 0, tgt::Sampler::SamplerFilter::LINEAR = 1 }

Functions

- const Result tgt::Sampler::add (const std::string &name, const SamplerAddressMode umode, const SamplerAddressMode vmode, const SamplerFilter magfilter, const SamplerFilter minFilter)
- const Result tgt::Sampler::add (const std::string &name, const SamplerAddressMode mode, const SamplerFilter filter)
- const Result tgt::Sampler::remove (const std::string &name)
- const std::string tgt::Sampler::list ()
- const Result tgt::Sampler::change (const std::string &name, const std::string &key, const std::string &value)
- const Result tgt::Sampler::write (FILE ∗file, const js::json &jsonarray)

Variables

- constexpr auto tgt::Sampler::SAMPLER_SUBFOLDER = "Sampler"
- constexpr auto tgt::Sampler::UMODE_PROPERTY = "u_address_mode"
- constexpr auto tgt::Sampler::VMODE_PROPERTY = "v_address_mode"
- constexpr auto tgt::Sampler::MAG_FILTER_PROPERTY = "magnification_filter"
- constexpr auto tgt::Sampler::MIN_FILTER_PROPERTY = "minification_filter"
- const auto tgt::Sampler::SAMPLER_PATH = fs::path(Util::RESOURCE_LOCATION).append(SAMPLER_SUBFOLDER)
- constexpr std::array tgt::Sampler::SUPPORTED_PROPERTIES = {UMODE_PROPERTY, VMODE_PROPERTY, MAG_FILTER_PROPERTY, MIN_FILTER_PROPERTY}
- constexpr SamplerAddressMode tgt::Sampler::SAMPLER_ADDRESS_MODE_MIN = SamplerAddressMode::REPEAT
- constexpr SamplerAddressMode tgt::Sampler::SAMPLER_ADDRESS_MODE_MAX = SamplerAddressMode::MIRROR_CLAMP_TO_EDGE
- constexpr SamplerFilter tgt::Sampler::SAMPLER_FILTER_MIN = SamplerFilter::NEAREST
- constexpr SamplerFilter tgt::Sampler::SAMPLER_FILTER_MAX = SamplerFilter::LINEAR

6.19 TGTools/TGTools/public/Shader.hpp File Reference

```
#include "Result.hpp"
#include <string>
#include "Util.hpp"
#include "../../../glslang/glslang/Public/ShaderLang.h"
```

Include dependency graph for Shader.hpp: This graph shows which files directly or indirectly include this file:

Namespaces

- tgt
- tgt::Shader
- tgt::Shader::ShaderInput

Typedefs

- using tgt::Shader::ShaderType = EShLanguage
- using tgt::Shader::ShaderInput::ShaderInputStride = uint32_t

Enumerations

- enum tgt::Shader::ShaderInput::ShaderInputType { tgt::Shader::ShaderInput::ShaderInputType::POSITION, tgt::Shader::ShaderInput::ShaderInputType::POSITION_UV, tgt::Shader::ShaderInput::ShaderInputType::POSITION_NORMAL tgt::Shader::ShaderInput::ShaderInputType::POSITION_NORMAL_UV }

Functions

- const Result tgt::Shader::add (const std::string &name, ShaderType type)
- const Result tgt::Shader::remove (const std::string &name)
- const std::string tgt::Shader::list ()

Variables

- constexpr ShaderInputStride tgt::Shader::ShaderInput::POS = sizeof(float) * 3
- constexpr ShaderInputStride tgt::Shader::ShaderInput::POS_TEX = POS + sizeof(float) * 2
- constexpr ShaderInputStride tgt::Shader::ShaderInput::POS_NOR = POS + sizeof(float) * 3
- constexpr ShaderInputStride tgt::Shader::ShaderInput::POS_NOR_TEX = POS_NOR + sizeof(float) * 2
- constexpr auto tgt::Shader::SHADER_EXTENSION = ".glsl"
- constexpr auto tgt::Shader::SHADER_SUBFOLDER = "Shaders"
- constexpr auto tgt::Shader::SHADER_TYPE_PROPERTY = "shaderType"
- constexpr std::array tgt::Shader::SUPPORTED_PROPERTIES = { SHADER_TYPE_PROPERTY }
- const auto tgt::Shader::SHADER_PATH = fs::path(Util::RESOURCE_LOCATION).append(SHADER_SUBFOLDER)

6.20 TGTools/TGTools/public/Texture.hpp File Reference

```
#include "Result.hpp"
#include <string>
#include "Util.hpp"
```

Include dependency graph for Texture.hpp: This graph shows which files directly or indirectly include this file:

Namespaces

- tgt
- tgt::Texture

Functions

- const Result tgt::Texture::add (const char *path)
- const Result tgt::Texture::add (const std::string &path)
- const Result tgt::Texture::remove (const char *name)
- const Result tgt::Texture::remove (const std::string &name)
- const std::string tgt::Texture::list ()

Variables

- constexpr auto tgt::Texture::TEXTURE_EXTENSION = ".tgx"
- constexpr auto tgt::Texture::TEXTURE_SUBFOLDER = "Textures"
- const auto tgt::Texture::TEXTURE_PATH = fs::path(Util::RESOURCE_LOCATION).append(TEXTURE_S↩ UBFOLDER)

6.21 TGTools/TGTools/public/Util.hpp File Reference

```
#include "Result.hpp"
#include <string>
#include <filesystem>
#include <fstream>
#include "../public/json.hpp"
#include <array>
```

Include dependency graph for Util.hpp: This graph shows which files directly or indirectly include this file:

Classes

- class tgt::Util::scope_exit< T, typename >

Namespaces

- tgt
- tgt::Util

Macros

- #define JSON_WRITE(path, json)
- #define JSON_LOAD(path, json)
- #define JSON_UPDATE(path, update)
- #define REMOVE_IF_FOUND(name, jobj, needle)
- #define STRING_SYNTAX_CHECK(string) for (char x : string) if (x < 48 || (x > 57 && x < 65) || (x > 90 && x < 97) || x > 122) return Result::BAD_STRING;
- #define STRING_CHECKS_C(string) if(string == nullptr || ∗string == 0) { printf("String check failed in (%s -> L%i)", __FILE__, __LINE__); return Result::BAD_ARGUMENTS;}
- #define STRING_CHECKS(string) if(string.empty()) { printf("String check failed in (%s -> L%i)", __FILE__, __LINE__); return Result::BAD_ARGUMENTS; }
- #define ENUM_CHECKS(enm, min, max) if(enm >= min && enm <= max) { printf("Enum check failed in (%s -> L%i)", __FILE__, __LINE__); return Result::BAD_ARGUMENTS; }
- #define ID_OF(iter, pred) std::distance(iter.begin(), std::find(iter.begin(), iter.end(), pred))
- #define ID_OF_P(iter, pred) std::distance(iter.begin(), std::find_if(iter.begin(), iter.end(), pred))

Functions

- template<class T , typename = std::enable_if_t<std::is_invocable_r_v<const Result, T, js::json&>>>
  const Result tgt::Util::jsonUpdatet (const fs::path &path, T lambda)
- template<class T , class U , size_t S, typename = std::enable_if_t<_validJson<T> && _validString<U>>>
  const Result tgt::Util::change (const fs::path &path, const std::string &key, const T &value, const std::array< U, S > &supported)
- template<class U , typename = std::enable_if_t<_validString<U> || std::is_null_pointer_v<U>>>
  const fs::path tgt::Util::getResource (fs::path resource, const std::string &name, const U &extension)
- const fs::path tgt::Util::getResource (const fs::path &resource, const std::string &name)
- const uint8_t ∗ tgt::Util::readFile (const std::string &name, size_t ∗sizeptr=nullptr)
- template<class U , typename = std::enable_if_t<std::is_invocable_r_v<bool, U, const fs::path&>>>
  const std::string tgt::Util::collect (const fs::path &path, U lambda)
- template<class U , typename = std::enable_if_t<std::is_invocable_r_v<bool, U, const fs::path&>>>
  const bool tgt::Util::find (const fs::path &path, const U lambda)
- template<class T , typename = std::enable_if_t<std::is_invocable_v<T, const js::json&>>>
  const Result tgt::Util::writeToFile (FILE ∗file, const js::json &jsonarray, T lambda)
- template<class T , typename = std::enable_if_t<std::is_invocable_r_v<bool, T, const std::string&> || std::is_invocable_r_v<bool, T, const fs::path&> || std::is_null_pointer_v<T>>>
  const Result tgt::Util::remove (const fs::path &parentpath, const std::string &name, const std::string &filter, T lambda)
- template<class T , typename = std::enable_if_t<std::is_invocable_r_v<bool, T, const std::string&> || std::is_invocable_r_v<bool, T, const fs::path&>>>
  const Result tgt::Util::remove (const fs::path &path, const std::string &name, T lambda)
- const Result tgt::Util::remove (const fs::path &path, const std::string &name)

Variables

- constexpr auto tgt::Util::RESOURCE_LOCATION = "Resource"
- constexpr auto tgt::Util::JSON = ".json"
- constexpr auto tgt::Util::JSON_FILTER = [ ](fs::path path) { return path.extension() == JSON; }
- template<class T , class... U>
  constexpr bool tgt::Util::_isAnyOf = std::disjunction_v<std::is_same<T, U>...>
- template<class T >
  constexpr bool tgt::Util::_validString = _isAnyOf<T, std::string, const std::string, char∗, const char∗, char∗ const, const char∗ const>
- template<class T >
  constexpr bool tgt::Util::_validPath = _isAnyOf<T, fs::path, const fs::path>
- template<class T >
  constexpr bool tgt::Util::_validPathString = _validPath<T> || _validString<T>
- template<class T >
  constexpr bool tgt::Util::_validJson = std::is_arithmetic_v<T> || _validString<T>

6.21.1 Macro Definition Documentation

6.21.1.1 ENUM_CHECKS

```
#define ENUM_CHECKS(
              enm,
              min,
              max ) if(enm >= min && enm <= max) { printf("Enum check failed in (%s -> L%i)",
__FILE__, __LINE__); return Result::BAD_ARGUMENTS; }
```

6.21.1.2 ID_OF

```
#define ID_OF(
              iter,
              pred ) std::distance(iter.begin(), std::find(iter.begin(), iter.end(), pred))
```

6.21.1.3 ID_OF_P

```
#define ID_OF_P(
              iter,
              pred ) std::distance(iter.begin(), std::find_if(iter.begin(), iter.end(), pred))
```

6.21.1.4 JSON_LOAD

```
#define JSON_LOAD(
              path,
              json )
```

Value:

```
std::ifstream path##Input(path);\
path##Input » json
```

6.21.1.5 JSON_UPDATE

```
#define JSON_UPDATE(
              path,
              update )
```

Value:

```
js::json json; \
JSON_LOAD(path, json);\
update \
JSON_WRITE(path, json)
```

6.21.1.6 JSON_WRITE

```
#define JSON_WRITE(
              path,
              json )
```

Value:

```
std::ofstream path##Output(path);\
path##Output « std::setw(4) « json « std::endl
```

6.21.1.7 REMOVE_IF_FOUND

```
#define REMOVE_IF_FOUND(
              name,
              jobj,
              needle )
```

Value:

```
    auto name##json = jobj;\
    auto name##pred = std::remove(name##json.begin(), name##json.end(), path);\
    auto name##found = name##pred == name##json.end();\
    if (name##found)\
        name##json.erase(name##pred)
```

6.21.1.8 STRING_CHECKS

```
#define STRING_CHECKS(
              string ) if(string.empty()) { printf("String check failed in (%s -> L%i)", __F←
ILE__, __LINE__); return Result::BAD_ARGUMENTS; }
```

6.21.1.9 STRING_CHECKS_C

```
#define STRING_CHECKS_C(
              string ) if(string == nullptr || *string == 0) { printf("String check failed in
(%s -> L%i)", __FILE__, __LINE__); return Result::BAD_ARGUMENTS;}
```

6.21.1.10 STRING_SYNTAX_CHECK

```
#define STRING_SYNTAX_CHECK(
              string ) for (char x : string) if (x < 48 || (x > 57 && x < 65) || (x > 90 && x
< 97) || x > 122) return Result::BAD_STRING;
```

Index