



Class GXFExtensionRegistrar

Table of contents

Class Documentation

- Defined in [File gxf_extension_registrar.hpp](#)

Class Documentation

class GXFExtensionRegistrar

Class to register a GXF extension.

This class is a helper class to register a GXF extension.

`GXFLoadExtensionFromPointer()` API is used to register the extension programmatically.

Public Types

enum class TypeKind

Kind of the Type.

Values:

enumerator kExtension

Extension.

enumerator kComponent

[Component](#).

Public Functions

GXFExtensionRegistrar() = delete

inline explicit GXFExtensionRegistrar(gxf_context_t context, const char *extension_name, const char *extension_description = "", gxf_tid_t tid = {0, 0})

Construct a new [GXFExtensionRegistrar](#) object.

If `tid` is not provided, a random `tid` is generated and used to register the extension.

Parameters

- **context** – The pointer to the GXF context.
- **extension_name** – The name of the extension.
- **tid** – The type ID of the extension to use.

```
inline bool is_allocated(gxf_tid_t tid, TypeKind kind) const
```

Check if the given tid is already allocated.

Parameters

- **tid** – The tid to check.
- **kind** – The kind of the tid to check.

Returns

true If the tid is already allocated. Otherwise, false.

```
inline gxf_tid_t allocate_tid(TypeKind kind)
```

Create a random tid that is not allocated.

Parameters

kind – The kind of the tid to create.

Returns

The allocated tid.

```
template<typename T, typename Base>
```

```
inline bool add_component(const char *description = "", gxf_tid_t tid = {0, 0})
```

Add a component to the extension.

If `tid` is not provided, a random `tid` is generated and used to register the component.

Template Parameters

- **T** – The type of the component.
- **Base** – The base type of the component.

Parameters

- **description** – The description of the component.
- **tid** – The type ID of the component to use.

Returns

true If the component is added successfully. Otherwise, false.

```
template<typename T>
inline bool add_type(const char *description = "", gfx_tid_t tid = {0, 0})
```

Add a type to the extension.

If `tid` is not provided, a random `tid` is generated and used to register the type.

Template Parameters

T – The type to add.

Parameters

- **description** – The description of the type.
- **tid** – The type ID of the type to use.

Returns

true If the type is added successfully. Otherwise, false.

```
inline bool register_extension(nvidia::gfx::Extension **out_extension_ptr = nullptr)
```

Register the extension.

Parameters

out_extension_ptr – If provided, the pointer to the extension is set to this pointer.

Returns

true If the extension is registered successfully. Otherwise, false.

```
inline void reset(gxf_context_t context, const char *extension_name, const char *extension_description = "", gxf_tid_t tid = {0, 0})
```

Reset the [GXFExtensionRegistrar](#) object.

If `tid` is not provided, a random `tid` is generated and used to register the extension.

Parameters

- **context** – The pointer to the GXF context.
- **extension_name** – The name of the extension.
- **tid** – The type ID of the extension to use.

Public Static Functions

```
static inline gxf_tid_t create_random_tid()
```

Create a random tid object.

Generate a sudo random tid using `std::mt19937_64`. This implementation does not guarantee that the tid is unique.

Returns

A random tid.