



Class ExtensionManager

Table of contents

Inheritance Relationships

Class Documentation

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

Inheritance Relationships

Derived Type

- `public holoscan::gxf::GXFExtensionManager` ([Class GXFExtensionManager](#))

Class Documentation

class ExtensionManager

Class to manage extensions.

This class is a helper class to manage extensions.

Subclassed by [holoscan::gxf::GXFExtensionManager](#)

Public Functions

inline explicit ExtensionManager(void *context)

Construct a new [ExtensionManager](#) object.

Parameters

context – The context.

virtual ~ExtensionManager() = default

Destroy the [ExtensionManager](#) object.

inline virtual void refresh()

Refresh the extension list.

Based on the current context, construct the internal extension list.

inline virtual bool load_extension(const std::string &file_name, bool no_error_message = false, const std::string &search_path_envs =

"HOLOSCAN_LIB_PATH")

Load an extension.

This method loads an extension and stores the extension handler so that it can be unloaded when the class is destroyed.

Parameters

- **file_name** – The file name of the extension (e.g. libmyextension.so).
- **no_error_message** – If true, no error message will be printed if the extension is not found.
- **search_path_envs** – The environment variable names that contains the search paths for the extension. The environment variable names are separated by a comma (.). (default: "HOLOSCAN_LIB_PATH").

Returns

true if the extension is loaded successfully, false otherwise.

```
inline virtual bool load_extensions_from_yaml(const YAML::Node &node, bool  
no_error_message = false, const std::string &search_path_envs =  
"HOLOSCAN_LIB_PATH", const std::string &key = "extensions")
```

Load extensions from a yaml file.

The yaml file should contain a list of extension file names under the key "extensions".

For example:

```
extensions: - /path/to/extension1.so - /path/to/extension2.so -  
/path/to/extension3.so
```

Parameters

- **node** – The yaml node.

- **no_error_message** – If true, no error message will be printed if the extension is not found.
- **search_path_envs** – The environment variable names that contains the search paths for the extension. The environment variable names are separated by a comma (.). (default: "HOLOSCAN_LIB_PATH").
- **key** – The key in the yaml node that contains the extension file names (default: "extensions").

Returns

true if the extension is loaded successfully, false otherwise.

Protected Attributes

`void *context_ = nullptr`

The context.

© Copyright 2022-2024, NVIDIA.. PDF Generated on 06/06/2024