MilliSuono

Generated by Doxygen 1.15.0

Chapter 1

Class Index

1.1 Class List

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

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 ms::Event Struct Reference

Structure representing an event.

`#include <Port.hpp>`

Public Member Functions

- Event (const std::string &type, const ControlValue &value, int sampleOffset)
  Constructor to initialize an event.

Public Attributes

- std::string **type**
  The type of the event.
- ControlValue **value**
  The value associated with the event.
- int **sampleOffset**
  The sample offset where the event occurs.

3.1.1 Detailed Description

Structure representing an event.

Contains information about the event type, the associated value, and the sample offset where it occurs.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 Event()

```
ms::Event::Event (
            const std::string & type,
            const ControlValue & value,
            int sampleOffset)  [inline]
```

Constructor to initialize an event.

Parameters

type	The type of the event.
value	The value associated with the event.
sampleOffset	The sample offset where the event occurs.

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

- include/core/Port.hpp

3.2 ms::Node Class Reference

Public Member Functions

- Node (const std::string &id)
  Constructor to initialize a Node.
- const std::string & getId () const
  Retrieves the unique identifier (id) of the node.
- const std::vector< Param > & getParams () const
  Retrieves the list of parameters (Param) for the node.
- void setParams (const std::vector< Param > &newParams)
  Sets a new list of parameters for the node.

3.2.1 Constructor & Destructor Documentation

3.2.1.1 Node()

```
ms::Node::Node (
            const std::string & id)  [inline]
```

Constructor to initialize a Node.

Parameters

id	The unique identifier for the Node.

3.2.2 Member Function Documentation

3.2.2.1 getId()

```
const std::string & ms::Node::getId () const  [inline]
```

Retrieves the unique identifier (id) of the node.

Returns

const std::string& The node's id.

3.2.2.2 getParams()

```
const std::vector< Param > & ms::Node::getParams () const  [inline]
```

Retrieves the list of parameters (Param) for the node.

Returns

const std::vector<Param>& The collection of parameters.

3.2.2.3 setParams()

```
void ms::Node::setParams (
            const std::vector< Param > & newParams)  [inline]
```

Sets a new list of parameters for the node.

Parameters

newParams	The new collection of parameters to be assigned.

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

- include/core/Node.hpp

3.3 ms::Param Struct Reference

Structure representing a parameter.

```
#include <Node.hpp>
```

Public Attributes

- std::string **name**
  The name of the parameter.
- ControlValue **value**
  The value of the parameter.

3.3.1 Detailed Description

Structure representing a parameter.

Contains the parameter's name and its value.

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

- include/core/Node.hpp

3.4 ms::Port Struct Reference

Structure representing a port.

`#include <Port.hpp>`

Public Member Functions

- Port (const std::string &name, PortType type)
  Constructor to initialize a port.

Public Attributes

- std::string **name**
  The name of the port.
- PortType **type**
  The type of the port.

3.4.1 Detailed Description

Structure representing a port.

Contains the port name and its type.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 Port()

```
ms::Port::Port (
            const std::string & name,
            PortType type)  [inline]
```

Constructor to initialize a port.

Parameters

name	The name of the port.
type	The type of the port.

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

- include/core/Port.hpp

Chapter 4

File Documentation

4.1 Node.hpp

```
00001 #pragma once
00002 #include "Port.hpp"
00003 #include <string>
00004 #include <unordered_map>
00005 #include <vector>
00006
00007 namespace ms {
00013 struct Param {
00014   std::string name;
00015   ControlValue value;
00016 };
00017
00018 class Node {
00019 public:
00024   Node(const std::string &id) : id(id) {}
00025
00028   const std::string &getId() const { return id; }
00029
00032   const std::vector<Param> &getParams() const { return params; }
00033
00038   void setParams(const std::vector<Param> &newParams) { params = newParams; }
00039
00040 private:
00041   const std::string id;
00042   std::vector<Param> params;
00043 };
00044
00045 } // namespace ms
```

4.2 Port.hpp

```
00001 #pragma once
00002 #include <string>
00003 #include <variant>
00004 namespace ms {
00005
00009 enum class PortType { Audio, Control, Event };
00010
00016 using ControlValue = std::variant<float, int, bool, std::string>;
00017
00024 struct Event {
00025   std::string type;
00026   ControlValue value;
00027   int sampleOffset;
00028
00035   Event(const std::string &type, const ControlValue &value, int sampleOffset)
00036       : type(type), value(value), sampleOffset(sampleOffset) {}
00037 };
00038
00044 struct Port {
00045   std::string name;
00046   PortType type;
00047
```

```
00053   Port(const std::string &name, PortType type) : name(name), type(type) {}
00054 };
00055
00056 } // namespace ms
```

Index