NamedItemRoutines

5.3

Generated by Doxygen 1.8.14

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

Chapter 5

Module Documentation

5.1 Models

Modules

- Utils

5.1.1 Detailed Description

5.2 Utils

Modules

- NamedItem

5.2.1 Detailed Description

5.3 NamedItem

Files

- file named_item.hh

  Define the NamedItem utility class.
- file named_item_messages.hh

  Define the class NamedItemMessages, the class that specifies the message IDs used in the named item model.
- file named_item.cc

  Construct the name of a NamedItem object by conjoining the passed parameters with a dot.
- file named_item_demangle.cc

  Demangle a C++ name, isolated from other NamedItem methods because this has the potential to get big and ugly if JEOD is ported to a number of different systems.
- file named_item_messages.cc

  Implement the class NamedItemMessages.

Namespaces

- jeod

  Namespace jeod.

Macros

- #define __has_include(x) 0

Functions

- bool operator== (const jeod::NamedItem &lhs, const std::string &rhs)

  Comparison to string.
- bool operator== (const std::string &lhs, const jeod::NamedItem &rhs)

5.3.1 Detailed Description

5.3.2 Macro Definition Documentation

5.3.2.1 __has_include

```
#define __has_include(
              x ) 0
```

Definition at line 32 of file named_item_demangle.cc.

5.3.3 Function Documentation

5.3.3.1 operator==() [1/2]

```
bool operator== (
             const jeod::NamedItem & lhs,
             const std::string & rhs )
```

Comparison to string.

Definition at line 103 of file named_item.cc.

References jeod::NamedItem::get_name().

5.3.3.2 operator==() [2/2]

```
bool operator== (
             const std::string & lhs,
             const jeod::NamedItem & rhs )
```

Definition at line 108 of file named_item.cc.

References jeod::NamedItem::get_name().

Chapter 6

Namespace Documentation

6.1 jeod Namespace Reference

Namespace jeod.

Data Structures

- class NamedItem

  Provides a set of static methods for constructing dot-conjoined names.
- class NamedItemMessages

  Specifies the message IDs used in the named_item model.

6.1.1 Detailed Description

Namespace jeod.

Chapter 7

Data Structure Documentation

7.1 jeod::NamedItem Class Reference

Provides a set of static methods for constructing dot-conjoined names.

```
#include <named_item.hh>
```

Public Types

- using size_type = std::string::size_type

  The size type used in std::string.

Public Member Functions

- NamedItem (std::string name_in=std::string(), bool frozen_in=false)

  Default constructor.
- NamedItem (const NamedItem &)=default

  Copy constructor.
- NamedItem (NamedItem &&)=default

  Move constructor.
- virtual ∼NamedItem ()=default

  Destructor.
- NamedItem & operator= (const NamedItem &src)

  Copy assignment.
- NamedItem & operator= (NamedItem &&src)

  Move assignment.
- NamedItem & operator= (const std::string &name_in)

  Assignment from a string.
- bool operator== (const NamedItem &rhs)

  Comparison of names.
- std::string get_name () const

  Getter for name.
- const char ∗ c_str () const

Getter for name, as a C-style string.

- size_type size () const

  Getter for the length of the name.
- bool get_is_frozen () const

  Getter for is_frozen.
- bool ends_with (size_type pos1, const std::string &other) const

  Compare the end of this string to a C-style string.
- std::string suffix (const std::string &test_name) const

  Given a dot-conjoined test name, find the part of the test name that follows this name, as a prefix.
- void set_name (const std::string &nameIn)
- template<typename... Type>
  void set_name (const std::string &nameIn, Type... namesIn)

  Set the name as a dot-conjoined string of the given inputs.
- void verify_unfrozen_name () const

  Verify that the name is not frozen.
- void validate_name (const char ∗file, unsigned int line, const std::string &variable_type, const std::string &variable_name)

  Checks whether a name is trivially invalid, failing if it is.
- void freeze_name ()

  Freeze the name – i.e., denote that the name as no longer settable.

Static Public Member Functions

- static std::string construct_name (const std::string &nameIn)

  Construct a name from the given input, as a string.
- template<typename... Type>
  static std::string construct_name (const std::string &nameIn, Type... namesIn)

  Construct a name as a dot-conjoined string of the given inputs.
- static std::string suffix (const std::string &prefix, const std::string &name)

  Given a prefix and a dot-conjoined name, find the part of the name that follows the prefix.
- static const std::string demangle (const std::type_info &info)

  Demangle a C++ name.
- static void validate_name (const char ∗file, unsigned int line, const std::string &variable_value, const std↵ ::string &variable_type, const std::string &variable_name)

  Checks whether a name is trivially invalid, failing if it is.

Protected Member Functions

- void unfreeze_name ()

  Unfreeze the name – i.e., denote that the name is now settable.

Private Attributes

- std::string name

  The item's name.
- bool is_frozen

  Indicates whether the name is frozen.

Friends

- class InputProcessor
- void init_attrjeod__NamedItem ()

7.1.1 Detailed Description

Provides a set of static methods for constructing dot-conjoined names.

The methods defined in this class allocate memory and do not release it. Releasing that memory is the responsibility of the calling function. Use the macro JEOD_DELETE_ARRAY to release this memory.

Prior to JEOD 4.0, the NamedItem class was not instantiable. It is in JEOD 4.0. The NamedItem class forms the basis of a thing with a name, with the name being a std::string. The construct_name functions and & related functions that allocate a C-style string are deprecated.

Definition at line 94 of file named_item.hh.

7.1.2 Member Typedef Documentation

7.1.2.1 size_type

```
using jeod::NamedItem::size_type = std::string::size_type
```

The size type used in std::string.

Definition at line 100 of file named_item.hh.

7.1.3 Constructor & Destructor Documentation

7.1.3.1 NamedItem() [1/3]

```
jeod::NamedItem::NamedItem (
            std::string name_in = std::string(),
            bool frozen_in = false )  [inline]
```

Default constructor.

This is the default constructor by virtue of the defaults.

Parameters

name_in	Initial value of the name, defaults to the empty string.
frozen_in	Initial value of is_frozen, defaults to false.

Definition at line 164 of file named_item.hh.

7.1.3.2 NamedItem() [2/3]

```
jeod::NamedItem::NamedItem (
            const NamedItem &  )  [default]
```

Copy constructor.

The default implementation works fine.

7.1.3.3 NamedItem() [3/3]

```
jeod::NamedItem::NamedItem (
            NamedItem &&  )  [default]
```

Move constructor.

The default implementation works fine.

7.1.3.4 ∼NamedItem()

```
virtual jeod::NamedItem::∼NamedItem ( )  [virtual], [default]
```

Destructor.

The default implementation virtually works fine.

7.1.4 Member Function Documentation

7.1.4.1 c_str()

```
const char* jeod::NamedItem::c_str ( ) const  [inline]
```

Getter for name, as a C-style string.

Definition at line 242 of file named_item.hh.

7.1.4.2 construct_name() [1/2]

```
static std::string jeod::NamedItem::construct_name (
            const std::string & nameIn )  [inline], [static]
```

Construct a name from the given input, as a string.

The input must not be the empty string or the null pointer.

Definition at line 106 of file named_item.hh.

7.1.4.3 construct_name() [2/2]

```
template<typename...  Type>
static std::string jeod::NamedItem::construct_name (
            const std::string & nameIn,
            Type...  namesIn )  [inline], [static]
```

Construct a name as a dot-conjoined string of the given inputs.

Each input must not be the empty string or the null pointer.

Template Parameters

Type	Types of the remaining arguments to construct_name.

Parameters

nameIn	First argument to construct_name.
names↩ In	Remaining arguments to construct_name.

Returns

The given inputs as a dot-conjoined string.

Definition at line 120 of file named_item.hh.

7.1.4.4 demangle()

```
const std::string jeod::NamedItem::demangle (
            const std::type_info & info )  [static]
```

Demangle a C++ name.

Returns

Demangled name

Parameters

in	*info*	Typeinfo to be demangled

Definition at line 58 of file named_item_demangle.cc.

7.1.4.5 ends_with()

```
bool jeod::NamedItem::ends_with (
            size_type pos1,
            const std::string & other ) const  [inline]
```

Compare the end of this string to a C-style string.

See std::string::compare.

Parameters

pos1	The start index in the name.
other	The C-style null-terminated string.

Returns

True if the end part of the name equals the given C-style string.

Definition at line 270 of file named_item.hh.

7.1.4.6 freeze_name()

```
void jeod::NamedItem::freeze_name ( )  [inline]
```

Freeze the name – i.e., denote that the name as no longer settable.

Definition at line 329 of file named_item.hh.

7.1.4.7 get_is_frozen()

```
bool jeod::NamedItem::get_is_frozen ( ) const  [inline]
```

Getter for is_frozen.

Definition at line 258 of file named_item.hh.

7.1.4.8 get_name()

```
std::string jeod::NamedItem::get_name ( ) const  [inline]
```

Getter for name.

Definition at line 234 of file named_item.hh.

Referenced by operator==().

7.1.4.9 operator=() [1/3]

```
NamedItem& jeod::NamedItem::operator= (
              const NamedItem & src )  [inline]
```

Copy assignment.

Only the name is copied, and only if the name isn't frozen.

Definition at line 195 of file named_item.hh.

7.1.4.10 operator=() [2/3]

```
NamedItem& jeod::NamedItem::operator= (
              NamedItem && src )  [inline]
```

Move assignment.

The default implementation works fine.

Definition at line 205 of file named_item.hh.

7.1.4.11 operator=() [3/3]

```
NamedItem& jeod::NamedItem::operator= (
              const std::string & name_in )  [inline]
```

Assignment from a string.

Definition at line 216 of file named_item.hh.

7.1.4.12 operator==()

```
bool jeod::NamedItem::operator== (
            const NamedItem & rhs )  [inline]
```

Comparison of names.

Definition at line 226 of file named_item.hh.

7.1.4.13 set_name() [1/2]

```
void jeod::NamedItem::set_name (
            const std::string & nameIn )  [inline]
```

Definition at line 288 of file named_item.hh.

7.1.4.14 set_name() [2/2]

```
template<typename...  Type>
void jeod::NamedItem::set_name (
            const std::string & nameIn,
            Type...  namesIn )  [inline]
```

Set the name as a dot-conjoined string of the given inputs.

Each input must not be the empty string or the null pointer.

Template Parameters

Type	Types of the remaining arguments to construct_name.

Parameters

nameIn	First argument to construct_name.
names↩ In	Remaining arguments to construct_name.

Definition at line 300 of file named_item.hh.

7.1.4.15 size()

```
size_type jeod::NamedItem::size ( ) const  [inline]
```

Getter for the length of the name.

Definition at line 250 of file named_item.hh.

7.1.4.16 suffix() [1/2]

```
std::string jeod::NamedItem::suffix (
             const std::string & prefix,
             const std::string & name )  [static]
```

Given a prefix and a dot-conjoined name, find the part of the name that follows the prefix.

For names of the form "prefix.suffix", this function returns a pointer to "suffix". The function returns the input name if the name does not start with "prefix.".

Returns

Suffix

Parameters

in	*prefix*	Prefix
in	*name*	Name, possibly prefixed

Definition at line 51 of file named_item.cc.

References name.

7.1.4.17 suffix() [2/2]

```
std::string jeod::NamedItem::suffix (
             const std::string & test_name ) const  [inline]
```

Given a dot-conjoined test name, find the part of the test name that follows this name, as a prefix.

For names of the form "prefix.suffix", this function returns a pointer to "suffix". The function returns the input name if the name does not start with "prefix.".

Returns

Suffix

Parameters

in	*test_name*	Test name, possibly prefixed

Definition at line 283 of file named_item.hh.

7.1.4.18 unfreeze_name()

```
void jeod::NamedItem::unfreeze_name ( )  [inline], [protected]
```

Unfreeze the name – i.e., denote that the name is now settable.

This exists solely to parallel freeze_name().

Definition at line 339 of file named_item.hh.

7.1.4.19 validate_name() [1/2]

```
void jeod::NamedItem::validate_name (
             const char * file,
             unsigned int line,
             const std::string & variable_value,
             const std::string & variable_type,
             const std::string & variable_name )  [static]
```

Checks whether a name is trivially invalid, failing if it is.

Parameters

in	*file*	Usually **FILE**
in	*line*	Usually **LINE**
in	*variable_value*	Value to check
in	*variable_type*	Variable description
in	*variable_name*	Variable name

Definition at line 71 of file named_item.cc.

References jeod::NamedItemMessages::invalid_name.

7.1.4.20 validate_name() [2/2]

```
void jeod::NamedItem::validate_name (
             const char * file,
             unsigned int line,
             const std::string & variable_type,
             const std::string & variable_name )  [inline]
```

Checks whether a name is trivially invalid, failing if it is.

Parameters

in	*file*	Usually **FILE**
in	*line*	Usually **LINE**
in	*variable_type*	Variable description
in	*variable_name*	Variable name

Definition at line 318 of file named_item.hh.

7.1.4.21 verify_unfrozen_name()

```
void jeod::NamedItem::verify_unfrozen_name ( ) const
```

Verify that the name is not frozen.

Definition at line 89 of file named_item.cc.

References jeod::NamedItemMessages::frozen_name, is_frozen, and name.

7.1.5 Friends And Related Function Documentation

7.1.5.1 init_attrjeod__NamedItem

```
void init_attrjeod__NamedItem ( )  [friend]
```

7.1.5.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 96 of file named_item.hh.

7.1.6 Field Documentation

7.1.6.1 is_frozen

```
bool jeod::NamedItem::is_frozen  [private]
```

Indicates whether the name is frozen.

trick_units(–)

Definition at line 353 of file named_item.hh.

Referenced by verify_unfrozen_name().

7.1.6.2 name

```
std::string jeod::NamedItem::name  [private]
```

The item's name.

trick_units(–)

Definition at line 348 of file named_item.hh.

Referenced by suffix(), and verify_unfrozen_name().

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

- named_item.hh
- named_item.cc
- named_item_demangle.cc

7.2 jeod::NamedItemMessages Class Reference

Specifies the message IDs used in the named_item model.

```
#include <named_item_messages.hh>
```

Public Member Functions

- NamedItemMessages ()=delete
- NamedItemMessages (const NamedItemMessages &)=delete
- NamedItemMessages & operator= (const NamedItemMessages &)=delete

Static Public Attributes

- static const char ∗ bad_args = "utils/named_item/" "bad_args"

  Error issued when the arguments to named item are invalid.
- static const char ∗ invalid_name = "utils/named_item/" "invalid_name"

  Error issued when a name is the null pointer or an empty string.
- static const char ∗ frozen_name = "utils/named_item/" "frozen_name"

  Error issued when set_name is called with the name marked as frozen.

Friends

- class InputProcessor
- void init_attrjeod__NamedItemMessages ()

7.2.1 Detailed Description

Specifies the message IDs used in the named_item model.

Definition at line 77 of file named_item_messages.hh.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 NamedItemMessages() [1/2]

```
jeod::NamedItemMessages::NamedItemMessages ( )  [delete]
```

7.2.2.2 NamedItemMessages() [2/2]

```
jeod::NamedItemMessages::NamedItemMessages (
            const NamedItemMessages &  )  [delete]
```

7.2.3 Member Function Documentation

7.2.3.1 operator=()

```
NamedItemMessages& jeod::NamedItemMessages::operator= (
            const NamedItemMessages &  )  [delete]
```

7.2.4 Friends And Related Function Documentation

7.2.4.1 init_attrjeod__NamedItemMessages

```
void init_attrjeod__NamedItemMessages ( )  [friend]
```

7.2.4.2 InputProcessor

```
friend class InputProcessor  [friend]
```

Definition at line 79 of file named_item_messages.hh.

7.2.5 Field Documentation

7.2.5.1 bad_args

```
char const * jeod::NamedItemMessages::bad_args = "utils/named_item/" "bad_args"  [static]
```

Error issued when the arguments to named item are invalid.

trick_units(–)

Definition at line 86 of file named_item_messages.hh.

7.2.5.2 frozen_name

```
char const * jeod::NamedItemMessages::frozen_name = "utils/named_item/" "frozen_name"  [static]
```

Error issued when set_name is called with the name marked as frozen.

trick_units(–)

Definition at line 96 of file named_item_messages.hh.

Referenced by jeod::NamedItem::verify_unfrozen_name().

7.2.5.3 invalid_name

```
char const * jeod::NamedItemMessages::invalid_name = "utils/named_item/" "invalid_name"  [static]
```

Error issued when a name is the null pointer or an empty string.

trick_units(–)

Definition at line 91 of file named_item_messages.hh.

Referenced by jeod::NamedItem::validate_name().

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

- named_item_messages.hh
- named_item_messages.cc

Chapter 8

File Documentation

8.1 named_item.cc File Reference

Construct the name of a NamedItem object by conjoining the passed parameters with a dot.

```
#include <cstdarg>
#include <cstddef>
#include <cstring>
#include "utils/memory/include/jeod_alloc.hh"
#include "utils/message/include/message_handler.hh"
#include "../include/named_item.hh"
#include "../include/named_item_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Functions

- bool operator== (const jeod::NamedItem &lhs, const std::string &rhs)

  Comparison to string.
- bool operator== (const std::string &lhs, const jeod::NamedItem &rhs)

8.1.1 Detailed Description

Construct the name of a NamedItem object by conjoining the passed parameters with a dot.

8.2 named_item.hh File Reference

Define the NamedItem utility class.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include <array>
#include <cstdarg>
#include <string>
#include <typeinfo>
#include <utility>
#include <vector>
```

Data Structures

- class jeod::NamedItem

  Provides a set of static methods for constructing dot-conjoined names.

Namespaces

- jeod

  Namespace jeod.

Functions

- bool operator== (const jeod::NamedItem &lhs, const std::string &rhs)

  Comparison to string.
- bool operator== (const std::string &lhs, const jeod::NamedItem &rhs)

8.2.1 Detailed Description

Define the NamedItem utility class.

8.3 named_item_demangle.cc File Reference

Demangle a C++ name, isolated from other NamedItem methods because this has the potential to get big and ugly if JEOD is ported to a number of different systems.

```
#include <cstdlib>
#include <string>
#include <typeinfo>
#include "../include/named_item.hh"
#include "../include/named_item_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define __has_include(x) 0

8.3.1 Detailed Description

Demangle a C++ name, isolated from other NamedItem methods because this has the potential to get big and ugly if JEOD is ported to a number of different systems.

8.4 named_item_messages.cc File Reference

Implement the class NamedItemMessages.

```
#include "utils/message/include/make_message_code.hh"
#include "../include/named_item_messages.hh"
```

Namespaces

- jeod

  Namespace jeod.

Macros

- #define MAKE_NAMEDITEM_MESSAGE_CODE(id) JEOD_MAKE_MESSAGE_CODE(NamedItem↩ Messages, "utils/named_item/", id)

8.4.1 Detailed Description

Implement the class NamedItemMessages.

8.4.2 Macro Definition Documentation

8.4.2.1 MAKE_NAMEDITEM_MESSAGE_CODE

```
#define MAKE_NAMEDITEM_MESSAGE_CODE(
              id ) JEOD_MAKE_MESSAGE_CODE(NamedItemMessages, "utils/named_item/", id)
```

Definition at line 43 of file named_item_messages.cc.

8.5 named_item_messages.hh File Reference

Define the class NamedItemMessages, the class that specifies the message IDs used in the named item model.

```
#include "utils/sim_interface/include/jeod_class.hh"
```

Data Structures

- class jeod::NamedItemMessages

  Specifies the message IDs used in the named_item model.

Namespaces

- jeod

  Namespace jeod.

8.5.1 Detailed Description

Define the class NamedItemMessages, the class that specifies the message IDs used in the named item model.

Index