

NamedItemRoutines

5.0

Generated by Doxygen 1.8.5

Wed Jun 1 2022 12:07:27

Contents

1	Module Index	1
1.1	Modules	1
2	Namespace Index	3
2.1	Namespace List	3
3	Data Structure Index	5
3.1	Data Structures	5
4	File Index	7
4.1	File List	7
5	Module Documentation	9
5.1	Models	9
5.1.1	Detailed Description	9
5.2	Utils	10
5.2.1	Detailed Description	10
5.3	NamedItem	11
5.3.1	Detailed Description	11
5.3.2	Macro Definition Documentation	11
5.3.2.1	__has_include	11
5.3.2.2	MAX_NAME_ITEMS	11
5.3.2.3	PATH	11
6	Namespace Documentation	13
6.1	jeod Namespace Reference	13
6.1.1	Detailed Description	13
7	Data Structure Documentation	15
7.1	jeod::NamedItem Class Reference	15
7.1.1	Detailed Description	17
7.1.2	Member Typedef Documentation	17
7.1.2.1	size_type	17
7.1.3	Constructor & Destructor Documentation	17

7.1.3.1	NamedItem	17
7.1.3.2	~NamedItem	18
7.1.4	Member Function Documentation	18
7.1.4.1	c_str	18
7.1.4.2	construct_name	18
7.1.4.3	construct_name	18
7.1.4.4	construct_name	18
7.1.4.5	construct_name	19
7.1.4.6	construct_name	19
7.1.4.7	construct_name	19
7.1.4.8	construct_name	20
7.1.4.9	construct_name_string	20
7.1.4.10	construct_name_string	21
7.1.4.11	demangle	21
7.1.4.12	ends_with	21
7.1.4.13	freeze_name	21
7.1.4.14	get_is_frozen	22
7.1.4.15	get_name	22
7.1.4.16	operator=	22
7.1.4.17	operator=	22
7.1.4.18	operator=	22
7.1.4.19	set_name	22
7.1.4.20	set_name	23
7.1.4.21	size	23
7.1.4.22	suffix	23
7.1.4.23	suffix	23
7.1.4.24	unfreeze_name	24
7.1.4.25	va_construct_name	24
7.1.4.26	validate_name	24
7.1.4.27	validate_name	25
7.1.4.28	vconstruct_name	25
7.1.4.29	verify_unfrozen_name	25
7.1.5	Friends And Related Function Documentation	25
7.1.5.1	init_attrjeod__NamedItem	26
7.1.5.2	InputProcessor	26
7.1.6	Field Documentation	26
7.1.6.1	__pad0__	26
7.1.6.2	is_frozen	26
7.1.6.3	is_frozen	26
7.1.6.4	name	26

7.2	jeod::NamedItemMessages Class Reference	27
7.2.1	Detailed Description	27
7.2.2	Constructor & Destructor Documentation	27
7.2.2.1	NamedItemMessages	27
7.2.2.2	NamedItemMessages	27
7.2.3	Member Function Documentation	27
7.2.3.1	operator=	27
7.2.4	Friends And Related Function Documentation	27
7.2.4.1	init_attrjeod__NamedItemMessages	27
7.2.4.2	InputProcessor	27
7.2.5	Field Documentation	28
7.2.5.1	bad_args	28
7.2.5.2	frozen_name	28
7.2.5.3	invalid_name	28
8	File Documentation	29
8.1	named_item.cc File Reference	29
8.1.1	Detailed Description	29
8.2	named_item.hh File Reference	29
8.2.1	Detailed Description	30
8.3	named_item_demangle.cc File Reference	30
8.3.1	Detailed Description	30
8.4	named_item_messages.cc File Reference	30
8.4.1	Detailed Description	31
8.5	named_item_messages.hh File Reference	31
8.5.1	Detailed Description	31
	Index	32

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

Models	9
Utils	10
NamedItem	11

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all namespaces with brief descriptions:

jeod	Namespace jeod	13
----------------------	--------------------------	----

Chapter 3

Data Structure Index

3.1 Data Structures

Here are the data structures with brief descriptions:

jeod::NamedItem	
Provides a set of static methods for constructing dot-conjoined names	15
jeod::NamedItemMessages	
Specifies the message IDs used in the named_item model	27

Chapter 4

File Index

4.1 File List

Here is a list of all files with brief descriptions:

named_item.cc	Construct the name of a NamedItem object by conjoining the passed parameters with a dot . . .	29
named_item.hh	Define the NamedItem utility class	29
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	30
named_item_messages.cc	Implement the class NamedItemMessages	30
named_item_messages.hh	Define the class NamedItemMessages, the class that specifies the message IDs used in the named item model	31

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 MAX_NAME_ITEMS 8`
- `#define __has_include(x) 0`
- `#define PATH "utils/named_item/"`

5.3.1 Detailed Description

5.3.2 Macro Definition Documentation

5.3.2.1 `#define __has_include(x) 0`

Definition at line 35 of file `named_item_demangle.cc`.

5.3.2.2 `#define MAX_NAME_ITEMS 8`

Definition at line 55 of file `named_item.cc`.

Referenced by `jeod::NamedItem::va_construct_name()`.

5.3.2.3 `#define PATH "utils/named_item/"`

Definition at line 43 of file `named_item_messages.cc`.

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` (`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.
- const `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 char *other) const
Compare the end of this string to a C-style string.
- const char * `suffix` (const char *test_name) const
Given a dot-conjoined test name, find the part of the test name that follows this name, as a prefix.
- template<typename Arg >
void `set_name` (Arg &&arg)

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

- `template<typename First, typename... Rest>`
`void set_name (First &&first, Rest &&...rest)`

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 char *variable_type, const char *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 char * construct_name (const char *name_item1)`

Create a copy of the provided name.

- `static char * construct_name (const char *name_item1, const char *name_item2)`

Construct a name as a dot-conjoined string.

- `static char * construct_name (const char *name_item1, const char *name_item2, const char *name_item3)`

Construct a name as a dot-conjoined string.

- `static char * construct_name (const char *name_item1, const char *name_item2, const char *name_item3, const char *name_item4)`

Construct a name as a dot-conjoined string.

- `static char * construct_name (const char *name_item1, const char *name_item2, const char *name_item3, const char *name_item4, const char *name_item5)`

Construct a name as a dot-conjoined string.

- `static char * construct_name (const char *name_item1, const char *name_item2, const char *name_item3, const char *name_item4, const char *name_item5, const char *name_item6)`

Construct a name as a dot-conjoined string.

- `static char * construct_name (const char *name_item1, const char *name_item2, const char *name_item3, const char *name_item4, const char *name_item5, const char *name_item6, const char *name_item7)`

Construct a name as a dot-conjoined string.

- `static char * vconstruct_name (const char *name_item,...)`

Construct a name as a dot-conjoined string.

- `static char * va_construct_name (const char *name_item, va_list args)`

Construct a name as a dot-conjoined string.

- `static const char * suffix (const char *prefix, const char *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 char *variable_value, const char *variable_type, const char *variable_name)`

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

- `template<typename Arg >`

`static std::string construct_name_string (Arg &&arg)`

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

- `template<typename First, typename... Rest>`

`static std::string construct_name_string (First &&first, Rest &&...rest)`

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

Data Fields

- `__pad0__`: `name` {std::move(name_in)}
Default constructor.
- `is_frozen`

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 66 of file `named_item.hh`.

7.1.2 Member Typedef Documentation

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

The size type used in std::string.

Definition at line 75 of file `named_item.hh`.

7.1.3 Constructor & Destructor Documentation

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

Move constructor.

The default implementation works fine.

Referenced by `construct_name_string()`.

7.1.3.2 `virtual jeod::NamedItem::~~NamedItem () [virtual],[default]`

Destructor.

The default implementation virtually works fine.

7.1.4 Member Function Documentation

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

Getter for name, as a C-style string.

Definition at line 408 of file `named_item.hh`.

References `name`.

7.1.4.2 `static char* jeod::NamedItem::construct_name (const char * name_item1) [inline],[static]`

Create a copy of the provided name.

Returns

The constructed name

Parameters

<code>in</code>	<code><i>name_item1</i></code>	First part of the name
-----------------	--------------------------------	------------------------

Definition at line 83 of file `named_item.hh`.

References `vconstruct_name()`.

7.1.4.3 `static char* jeod::NamedItem::construct_name (const char * name_item1, const char * name_item2) [inline],[static]`

Construct a name as a dot-conjoined string.

Returns

The constructed name

Parameters

<code>in</code>	<code><i>name_item1</i></code>	First part of the name
<code>in</code>	<code><i>name_item2</i></code>	Second part of the name

Definition at line 95 of file `named_item.hh`.

References `vconstruct_name()`.

7.1.4.4 `static char* jeod::NamedItem::construct_name (const char * name_item1, const char * name_item2, const char * name_item3) [inline],[static]`

Construct a name as a dot-conjoined string.

Returns

The constructed name

Parameters

in	<i>name_item1</i>	First part of the name
in	<i>name_item2</i>	Second part of the name
in	<i>name_item3</i>	Third part of the name

Definition at line 111 of file named_item.hh.

References `vconstruct_name()`.

7.1.4.5 `static char* jeod::NamedItem::construct_name (const char * name_item1, const char * name_item2, const char * name_item3, const char * name_item4)` `[inline]`, `[static]`

Construct a name as a dot-conjoined string.

Returns

The constructed name

Parameters

in	<i>name_item1</i>	First part of the name
in	<i>name_item2</i>	Second part of the name
in	<i>name_item3</i>	Third part of the name
in	<i>name_item4</i>	Fourth part of the name

Definition at line 130 of file named_item.hh.

References `vconstruct_name()`.

7.1.4.6 `static char* jeod::NamedItem::construct_name (const char * name_item1, const char * name_item2, const char * name_item3, const char * name_item4, const char * name_item5)` `[inline]`, `[static]`

Construct a name as a dot-conjoined string.

Returns

The constructed name

Parameters

in	<i>name_item1</i>	First part of the name
in	<i>name_item2</i>	Second part of the name
in	<i>name_item3</i>	Third part of the name
in	<i>name_item4</i>	Fourth part of the name
in	<i>name_item5</i>	Fifth part of the name

Definition at line 152 of file named_item.hh.

References `vconstruct_name()`.

7.1.4.7 `static char* jeod::NamedItem::construct_name (const char * name_item1, const char * name_item2, const char * name_item3, const char * name_item4, const char * name_item5, const char * name_item6)` `[inline]`, `[static]`

Construct a name as a dot-conjoined string.

Returns

The constructed name

Parameters

in	<i>name_item1</i>	First part of the name
in	<i>name_item2</i>	Second part of the name
in	<i>name_item3</i>	Third part of the name
in	<i>name_item4</i>	Fourth part of the name
in	<i>name_item5</i>	Fifth part of the name
in	<i>name_item6</i>	Sixth part of the name

Definition at line 177 of file named_item.hh.

References `vconstruct_name()`.

```
7.1.4.8 static char* jeod::NamedItem::construct_name ( const char * name_item1, const char * name_item2, const char *
name_item3, const char * name_item4, const char * name_item5, const char * name_item6, const char * name_item7
) [inline],[static]
```

Construct a name as a dot-conjoined string.

Returns

The constructed name

Parameters

in	<i>name_item1</i>	First part of the name
in	<i>name_item2</i>	Second part of the name
in	<i>name_item3</i>	Third part of the name
in	<i>name_item4</i>	Fourth part of the name
in	<i>name_item5</i>	Fifth part of the name
in	<i>name_item6</i>	Sixth part of the name
in	<i>name_item7</i>	Seventh part of the name

Definition at line 205 of file named_item.hh.

References `vconstruct_name()`.

```
7.1.4.9 template<typename Arg > static std::string jeod::NamedItem::construct_name_string ( Arg && arg ) [inline],
[static]
```

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

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

Template Parameters

<i>Arg</i>	Type of the argument to <code>construct_name_string</code> .
------------	--

Parameters

<i>arg</i>	Argument to <code>construct_name_string</code> .
------------	--

Returns

`std::string` that is conceptually equal to `(==) arg`.

Definition at line 299 of file named_item.hh.

References `validate_name()`.

Referenced by `set_name()`.

7.1.4.10 `template<typename First, typename... Rest> static std::string jeod::NamedItem::construct_name_string (First && first, Rest &&... rest) [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

<i>First</i>	Type of the first argument to <code>construct_name_string</code> .
<i>Rest</i>	Types of the remaining arguments to <code>construct_name_string</code> .

Parameters

<i>first</i>	First argument to <code>construct_name_string</code> .
<i>rest</i>	Remaining arguments to <code>construct_name_string</code> .

Returns

The given inputs as a dot-conjoined string.

Definition at line 316 of file `named_item.hh`.

References `NamedItem()`, and `validate_name()`.

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

Demangle a C++ name.

Returns

Demangled name

Parameters

<i>in</i>	<i>info</i>	Typeinfo to be demangled
-----------	-------------	--------------------------

Definition at line 62 of file `named_item_demangle.cc`.

7.1.4.12 `bool jeod::NamedItem::ends_with (size_type pos1, const char * other) const [inline]`

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

See `std::string::compare`.

Parameters

<i>pos1</i>	The start index in the name.
<i>other</i>	The C-style null-terminated string.

Returns

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

Definition at line 436 of file `named_item.hh`.

References `name`.

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

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

Definition at line 507 of file named_item.hh.

References is_frozen.

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

Getter for is_frozen.

Definition at line 424 of file named_item.hh.

References is_frozen.

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

Getter for name.

Definition at line 400 of file named_item.hh.

References name.

7.1.4.16 `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 368 of file named_item.hh.

References name, and verify_unfrozen_name().

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

Move assignment.

The default implementation works fine.

Definition at line 378 of file named_item.hh.

References name, and verify_unfrozen_name().

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

Assignment from a string.

Definition at line 389 of file named_item.hh.

References name, and verify_unfrozen_name().

7.1.4.19 `template<typename Arg > void jeod::NamedItem::set_name (Arg && arg)` `[inline]`

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

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

Template Parameters

<i>Arg</i>	Type of the argument to construct_name_string.
------------	--

Parameters

<i>arg</i>	Argument to construct_name_string.
------------	------------------------------------

Definition at line 461 of file named_item.hh.

References construct_name_string(), name, and verify_unfrozen_name().

7.1.4.20 `template<typename First , typename... Rest> void jeod::NamedItem::set_name (First && first, Rest &&... rest)`
`[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

<i>First</i>	Type of the first argument to construct_name_string.
<i>Rest</i>	Types of the remaining arguments to construct_name_string.

Parameters

<i>first</i>	First argument to construct_name_string.
<i>rest</i>	Remaining arguments to construct_name_string.

Definition at line 476 of file named_item.hh.

References construct_name_string(), name, and verify_unfrozen_name().

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

Getter for the length of the name.

Definition at line 416 of file named_item.hh.

References name.

7.1.4.22 `const char * jeod::NamedItem::suffix (const char * prefix, const char * 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

<i>in</i>	<i>prefix</i>	Prefix
<i>in</i>	<i>name</i>	Name, possibly prefixed

Definition at line 159 of file named_item.cc.

References name.

Referenced by suffix().

7.1.4.23 `const char* jeod::NamedItem::suffix (const char * 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

<i>in</i>	<i>test_name</i>	Test name, possibly prefixed
-----------	------------------	------------------------------

Definition at line 449 of file named_item.hh.

References name, and suffix().

7.1.4.24 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 519 of file named_item.hh.

References is_frozen.

7.1.4.25 char * jeod::NamedItem::va_construct_name (const char * name_item, va_list args) [static]

Construct a name as a dot-conjoined string.

Notes –

- This function takes a va_list argument that contains any additional strings to be appended.
- The calling function must form the args argument by invoking va_start().
- The calling function should not invoke va_end(); this is done inside [va_construct_name\(\)](#).
- The last argument embodied in the args argument must be a NULL to signal the end of the argument list.

Returns

The constructed name

Parameters

<i>in</i>	<i>name_item</i>	First part of the name
<i>in</i>	<i>args</i>	Rest of the name

Definition at line 82 of file named_item.cc.

References jeod::NamedItemMessages::bad_args, MAX_NAME_ITEMS, and name.

Referenced by vconstruct_name().

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

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

Parameters

in	<i>file</i>	Usually FILE
in	<i>line</i>	Usually LINE
in	<i>variable_value</i>	Value to check
in	<i>variable_type</i>	Variable description
in	<i>variable_name</i>	Variable name

Definition at line 184 of file named_item.cc.

References jeod::NamedItemMessages::invalid_name.

Referenced by construct_name_string(), and validate_name().

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

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

Parameters

in	<i>file</i>	Usually FILE
in	<i>line</i>	Usually LINE
in	<i>variable_type</i>	Variable description
in	<i>variable_name</i>	Variable name

Definition at line 495 of file named_item.hh.

References name, and validate_name().

7.1.4.28 char * jeod::NamedItem::vconstruct_name (const char * *name_item*, ...) [static]

Construct a name as a dot-conjoined string.

Note that this is a varargs function. The last argument must be NULL to signal the end of the argument list.

Returns

The constructed name

Parameters

in	<i>name_item</i>	First part of the name
in	...	Rest of the name

Definition at line 62 of file named_item.cc.

References name, and va_construct_name().

Referenced by construct_name().

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

Verify that the name is not frozen.

Definition at line 211 of file named_item.cc.

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

Referenced by operator=(), and set_name().

7.1.5 Friends And Related Function Documentation

7.1.5.1 `void init_attrjeod__NamedItem () [friend]`

7.1.5.2 `friend class InputProcessor [friend]`

Definition at line 68 of file `named_item.hh`.

7.1.6 Field Documentation

7.1.6.1 `jeod::NamedItem::__pad0__`

Default constructor.

This is the default constructor by virtue of the defaults.

Parameters

<code>name_in</code>	Initial value of the name, defaults to the empty string.
<code>frozen_in</code>	Initial value of <code>is_frozen</code> , defaults to false.

Definition at line 339 of file `named_item.hh`.

7.1.6.2 `jeod::NamedItem::is_frozen`

Initial value:

```
{frozen_in}
{ }

NamedItem (const NamedItem&) = default
```

Definition at line 340 of file `named_item.hh`.

Referenced by `freeze_name()`, `get_is_frozen()`, `unfreeze_name()`, and `verify_unfrozen_name()`.

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

Indicates whether the name is frozen.

`trick_units(-)`

Definition at line 535 of file `named_item.hh`.

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

The item's name.

`trick_units(-)`

Definition at line 530 of file `named_item.hh`.

Referenced by `c_str()`, `ends_with()`, `get_name()`, `operator=()`, `set_name()`, `size()`, `suffix()`, `va_construct_name()`, `validate_name()`, `vconstruct_name()`, 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>
```

Static Public Attributes

- static char const * [bad_args](#) = "utils/named_item/" "bad_args"
Error issued when the arguments to named item are invalid.
- static char const * [invalid_name](#) = "utils/named_item/" "invalid_name"
Error issued when a name is the null pointer or an empty string.
- static char const * [frozen_name](#) = "utils/named_item/" "frozen_name"
Error issued when set_name is called with the name marked as frozen.

Private Member Functions

- [NamedItemMessages](#) (void)
- [NamedItemMessages](#) (const [NamedItemMessages](#) &)
- [NamedItemMessages](#) & [operator=](#) (const [NamedItemMessages](#) &)

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 49 of file named_item_messages.hh.

7.2.2 Constructor & Destructor Documentation

7.2.2.1 `jeod::NamedItemMessages::NamedItemMessages (void)` `[private]`

7.2.2.2 `jeod::NamedItemMessages::NamedItemMessages (const NamedItemMessages &)` `[private]`

7.2.3 Member Function Documentation

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

7.2.4 Friends And Related Function Documentation

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

7.2.4.2 `friend class InputProcessor` `[friend]`

Definition at line 52 of file named_item_messages.hh.

7.2.5 Field Documentation

7.2.5.1 `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 62 of file `named_item_messages.hh`.

Referenced by `jeod::NamedItem::va_construct_name()`.

7.2.5.2 `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 72 of file `named_item_messages.hh`.

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

7.2.5.3 `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 67 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 <cstdlib>
#include <cstdio>
#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`.

Macros

- `#define MAX_NAME_ITEMS 8`

8.1.1 Detailed Description

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

Definition in file `named_item.cc`.

8.2 `named_item.hh` File Reference

Define the `NamedItem` utility class.

```
#include "utils/sim_interface/include/jeod_class.hh"
#include <cstdlib>
#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.

8.2.1 Detailed Description

Define the NamedItem utility class.

Definition in file [named_item.hh](#).

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.

Definition in file [named_item_demangle.cc](#).

8.4 [named_item_messages.cc](#) File Reference

Implement the class NamedItemMessages.

```
#include "../include/named_item_messages.hh"
```

Namespaces

- [jeod](#)
Namespace jeod.

Macros

- `#define PATH "utils/named_item/"`

8.4.1 Detailed Description

Implement the class NamedItemMessages.

Definition in 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.

Definition in file [named_item_messages.hh](#).

Index

- ~NamedItem
 - jeod::NamedItem, [17](#)
- __has_include
 - NamedItem, [11](#)
- __pad0__
 - jeod::NamedItem, [26](#)
- bad_args
 - jeod::NamedItemMessages, [28](#)
- c_str
 - jeod::NamedItem, [18](#)
- construct_name
 - jeod::NamedItem, [18–20](#)
- construct_name_string
 - jeod::NamedItem, [20](#)
- demangle
 - jeod::NamedItem, [21](#)
- ends_with
 - jeod::NamedItem, [21](#)
- freeze_name
 - jeod::NamedItem, [21](#)
- frozen_name
 - jeod::NamedItemMessages, [28](#)
- get_is_frozen
 - jeod::NamedItem, [22](#)
- get_name
 - jeod::NamedItem, [22](#)
- init_attrjeod__NamedItem
 - jeod::NamedItem, [25](#)
- init_attrjeod__NamedItemMessages
 - jeod::NamedItemMessages, [27](#)
- InputProcessor
 - jeod::NamedItem, [26](#)
 - jeod::NamedItemMessages, [27](#)
- invalid_name
 - jeod::NamedItemMessages, [28](#)
- is_frozen
 - jeod::NamedItem, [26](#)
- jeod, [13](#)
- jeod::NamedItem, [15](#)
 - ~NamedItem, [17](#)
 - __pad0__, [26](#)
 - c_str, [18](#)
 - construct_name, [18–20](#)
 - construct_name_string, [20](#)
 - demangle, [21](#)
 - ends_with, [21](#)
 - freeze_name, [21](#)
 - get_is_frozen, [22](#)
 - get_name, [22](#)
 - init_attrjeod__NamedItem, [25](#)
 - InputProcessor, [26](#)
 - is_frozen, [26](#)
 - name, [26](#)
 - NamedItem, [17](#)
 - operator=, [22](#)
 - set_name, [22, 23](#)
 - size, [23](#)
 - size_type, [17](#)
 - suffix, [23](#)
 - unfreeze_name, [24](#)
 - va_construct_name, [24](#)
 - validate_name, [24, 25](#)
 - vconstruct_name, [25](#)
 - verify_unfrozen_name, [25](#)
- jeod::NamedItemMessages, [27](#)
 - bad_args, [28](#)
 - frozen_name, [28](#)
 - init_attrjeod__NamedItemMessages, [27](#)
 - InputProcessor, [27](#)
 - invalid_name, [28](#)
 - NamedItemMessages, [27](#)
 - operator=, [27](#)
- MAX_NAME_ITEMS
 - NamedItem, [11](#)
- Models, [9](#)
- name
 - jeod::NamedItem, [26](#)
- named_item.cc, [29](#)
- named_item.hh, [29](#)
- named_item_demangle.cc, [30](#)
- named_item_messages.cc, [30](#)
- named_item_messages.hh, [31](#)
- NamedItem, [11](#)
 - __has_include, [11](#)
 - jeod::NamedItem, [17](#)
 - MAX_NAME_ITEMS, [11](#)
 - PATH, [11](#)
- NamedItemMessages
 - jeod::NamedItemMessages, [27](#)
- operator=

jeod::NamedItem, [22](#)
jeod::NamedItemMessages, [27](#)

PATH

NamedItem, [11](#)

set_name

jeod::NamedItem, [22](#), [23](#)

size

jeod::NamedItem, [23](#)

size_type

jeod::NamedItem, [17](#)

suffix

jeod::NamedItem, [23](#)

unfreeze_name

jeod::NamedItem, [24](#)

Utils, [10](#)

va_construct_name

jeod::NamedItem, [24](#)

validate_name

jeod::NamedItem, [24](#), [25](#)

vconstruct_name

jeod::NamedItem, [25](#)

verify_unfrozen_name

jeod::NamedItem, [25](#)