

## NamedItemRoutines

5.1

Generated by Doxygen 1.8.5

Mon Jul 31 2023 11:41:32



# Contents

<b>1</b>	<b>Module Index</b>	<b>1</b>
1.1	Modules . . . . .	1
<b>2</b>	<b>Namespace Index</b>	<b>3</b>
2.1	Namespace List . . . . .	3
<b>3</b>	<b>Data Structure Index</b>	<b>5</b>
3.1	Data Structures . . . . .	5
<b>4</b>	<b>File Index</b>	<b>7</b>
4.1	File List . . . . .	7
<b>5</b>	<b>Module Documentation</b>	<b>9</b>
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
5.3.3	Function Documentation . . . . .	12
5.3.3.1	operator== . . . . .	12
5.3.3.2	operator== . . . . .	12
<b>6</b>	<b>Namespace Documentation</b>	<b>13</b>
6.1	jeod Namespace Reference . . . . .	13
6.1.1	Detailed Description . . . . .	13
<b>7</b>	<b>Data Structure Documentation</b>	<b>15</b>
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	operator==	22
7.1.4.20	set_name	22
7.1.4.21	set_name	24
7.1.4.22	size	24
7.1.4.23	suffix	24
7.1.4.24	suffix	25
7.1.4.25	unfreeze_name	25
7.1.4.26	va_construct_name	25
7.1.4.27	validate_name	25
7.1.4.28	validate_name	26
7.1.4.29	vconstruct_name	26
7.1.4.30	verify_unfrozen_name	26
7.1.5	Friends And Related Function Documentation	27
7.1.5.1	init_attrjeod__NamedItem	27
7.1.5.2	InputProcessor	27
7.1.6	Field Documentation	27

7.1.6.1	<code>__pad0__</code>	27
7.1.6.2	<code>is_frozen</code>	27
7.1.6.3	<code>is_frozen</code>	27
7.1.6.4	<code>name</code>	27
7.2	<code>jeod::NamedItemMessages</code> Class Reference	28
7.2.1	Detailed Description	28
7.2.2	Constructor & Destructor Documentation	28
7.2.2.1	<code>NamedItemMessages</code>	28
7.2.2.2	<code>NamedItemMessages</code>	28
7.2.3	Member Function Documentation	28
7.2.3.1	<code>operator=</code>	28
7.2.4	Friends And Related Function Documentation	28
7.2.4.1	<code>init_attrjeod__NamedItemMessages</code>	28
7.2.4.2	<code>InputProcessor</code>	28
7.2.5	Field Documentation	29
7.2.5.1	<code>bad_args</code>	29
7.2.5.2	<code>frozen_name</code>	29
7.2.5.3	<code>invalid_name</code>	29
<b>8</b>	<b>File Documentation</b>	<b>31</b>
8.1	<code>named_item.cc</code> File Reference	31
8.1.1	Detailed Description	31
8.2	<code>named_item.hh</code> File Reference	31
8.2.1	Detailed Description	32
8.3	<code>named_item_demangle.cc</code> File Reference	32
8.3.1	Detailed Description	33
8.4	<code>named_item_messages.cc</code> File Reference	33
8.4.1	Detailed Description	33
8.5	<code>named_item_messages.hh</code> File Reference	33
8.5.1	Detailed Description	33
<b>Index</b>		<b>34</b>



# 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:

<a href="#">jeod</a>	Namespace jeod . . . . .	<a href="#">13</a>
----------------------	--------------------------	--------------------



## Chapter 3

# Data Structure Index

### 3.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">jeod::NamedItem</a>	
Provides a set of static methods for constructing dot-conjoined names	15
<a href="#">jeod::NamedItemMessages</a>	
Specifies the message IDs used in the named_item model	28



## Chapter 4

# File Index

### 4.1 File List

Here is a list of all files with brief descriptions:

<a href="#">named_item.cc</a>	Construct the name of a NamedItem object by conjoining the passed parameters with a dot . . .	31
<a href="#">named_item.hh</a>	Define the NamedItem utility class . . . . .	31
<a href="#">named_item_demangle.cc</a>	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 . . . . .	32
<a href="#">named_item_messages.cc</a>	Implement the class NamedItemMessages . . . . .	33
<a href="#">named_item_messages.hh</a>	Define the class NamedItemMessages, the class that specifies the message IDs used in the named item model . . . . .	33



## 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/"`

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

Definition at line 33 of file `named_item_demangle.cc`.

##### 5.3.2.2 `#define MAX_NAME_ITEMS 8`

Definition at line 48 of file `named_item.cc`.

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

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

Definition at line 37 of file `named_item_messages.cc`.

### 5.3.3 Function Documentation

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

Comparison to string.

Definition at line 209 of file `named_item.cc`.

References `jeod::NamedItem::get_name()`.

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

Definition at line 214 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` (`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.*
- 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 90 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 99 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 439 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 107 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 119 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 135 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 154 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 176 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 201 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 229 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 323 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 340 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 60 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 467 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 538 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 455 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 431 of file `named_item.hh`.

References `name`.

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

**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 392 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 402 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 413 of file `named_item.hh`.

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

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

Comparison of names.

Definition at line 423 of file `named_item.hh`.

References `get_name( )`, and `name`.

**7.1.4.20** `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 492 of file named\_item.hh.

References construct\_name\_string(), name, and verify\_unfrozen\_name().

**7.1.4.21** `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 507 of file named\_item.hh.

References construct\_name\_string(), name, and verify\_unfrozen\_name().

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

Getter for the length of the name.

Definition at line 447 of file named\_item.hh.

References name.

**7.1.4.23** `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 143 of file named\_item.cc.

References name.

Referenced by suffix().

**7.1.4.24** `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 480 of file `named_item.hh`.

References `name`, and `suffix()`.

**7.1.4.25** `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 550 of file `named_item.hh`.

References `is_frozen`.

**7.1.4.26** `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 75 of file `named_item.cc`.

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

Referenced by `vconstruct_name()`.

**7.1.4.27** `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 <b>FILE</b>
in	<i>line</i>	Usually <b>LINE</b>
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 169 of file named\_item.cc.

References jeod::NamedItemMessages::invalid\_name.

Referenced by construct\_name\_string(), and validate\_name().

**7.1.4.28** 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 <b>FILE</b>
in	<i>line</i>	Usually <b>LINE</b>
in	<i>variable_type</i>	Variable description
in	<i>variable_name</i>	Variable name

Definition at line 526 of file named\_item.hh.

References name, and validate\_name().

**7.1.4.29** 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 (const char \*)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 55 of file named\_item.cc.

References name, and va\_construct\_name().

Referenced by construct\_name().

**7.1.4.30** void jeod::NamedItem::verify\_unfrozen\_name ( ) const

Verify that the name is not frozen.

Definition at line 197 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 92 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

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

Definition at line 363 of file `named_item.hh`.

7.1.6.2 `jeod::NamedItem::is_frozen`

**Initial value:**

```
{frozen_in}
{ }
```

```
NamedItem (const NamedItem&) = default
```

Definition at line 364 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 566 of file `named_item.hh`.

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

The item's name.

`trick_units(-)`

Definition at line 561 of file `named_item.hh`.

Referenced by `c_str()`, `ends_with()`, `get_name()`, `operator=()`, `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 79 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 82 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 92 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 102 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 97 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

#### 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.

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.*

## 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.

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, [27](#)
- bad\_args
  - jeod::NamedItemMessages, [29](#)
- 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, [29](#)
- get\_is\_frozen
  - jeod::NamedItem, [22](#)
- get\_name
  - jeod::NamedItem, [22](#)
- init\_attrjeod\_\_NamedItem
  - jeod::NamedItem, [27](#)
- init\_attrjeod\_\_NamedItemMessages
  - jeod::NamedItemMessages, [28](#)
- InputProcessor
  - jeod::NamedItem, [27](#)
  - jeod::NamedItemMessages, [28](#)
- invalid\_name
  - jeod::NamedItemMessages, [29](#)
- is\_frozen
  - jeod::NamedItem, [27](#)
- jeod, [13](#)
- jeod::NamedItem, [15](#)
  - ~NamedItem, [17](#)
  - \_\_pad0\_\_, [27](#)
  - 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, [27](#)
  - InputProcessor, [27](#)
  - is\_frozen, [27](#)
  - name, [27](#)
  - NamedItem, [17](#)
  - operator=, [22](#)
  - operator==, [22](#)
  - set\_name, [22, 24](#)
  - size, [24](#)
  - size\_type, [17](#)
  - suffix, [24](#)
  - unfreeze\_name, [25](#)
  - va\_construct\_name, [25](#)
  - validate\_name, [25, 26](#)
  - vconstruct\_name, [26](#)
  - verify\_unfrozen\_name, [26](#)
- jeod::NamedItemMessages, [28](#)
  - bad\_args, [29](#)
  - frozen\_name, [29](#)
  - init\_attrjeod\_\_NamedItemMessages, [28](#)
  - InputProcessor, [28](#)
  - invalid\_name, [29](#)
  - NamedItemMessages, [28](#)
  - operator=, [28](#)
- MAX\_NAME\_ITEMS
  - NamedItem, [11](#)
- Models, [9](#)
- name
  - jeod::NamedItem, [27](#)
- named\_item.cc, [31](#)
- named\_item.hh, [31](#)
- named\_item\_demangle.cc, [32](#)
- named\_item\_messages.cc, [33](#)
- named\_item\_messages.hh, [33](#)
- NamedItem, [11](#)
  - \_\_has\_include, [11](#)
  - jeod::NamedItem, [17](#)
  - MAX\_NAME\_ITEMS, [11](#)
  - operator==, [12](#)
  - PATH, [11](#)
- NamedItemMessages
  - jeod::NamedItemMessages, [28](#)



operator=  
    jeod::NamedItem, [22](#)  
    jeod::NamedItemMessages, [28](#)  
operator==  
    jeod::NamedItem, [22](#)  
    NamedItem, [12](#)

PATH  
    NamedItem, [11](#)

set\_name  
    jeod::NamedItem, [22](#), [24](#)  
size  
    jeod::NamedItem, [24](#)  
size\_type  
    jeod::NamedItem, [17](#)  
suffix  
    jeod::NamedItem, [24](#)

unfreeze\_name  
    jeod::NamedItem, [25](#)  
Utils, [10](#)

va\_construct\_name  
    jeod::NamedItem, [25](#)  
validate\_name  
    jeod::NamedItem, [25](#), [26](#)  
vconstruct\_name  
    jeod::NamedItem, [26](#)  
verify\_unfrozen\_name  
    jeod::NamedItem, [26](#)