

Sex

1.0

Généré par Doxygen 1.8.13

Table des matières

1	Une énumération pour représenter le sexe d'une personne	1
2	Index des espaces de nommage	1
2.1	Liste des espaces de nommage	1
3	Index des fichiers	2
3.1	Liste des fichiers	2
4	Documentation des espaces de nommage	2
4.1	Référence de l'espace de nommage <code>nvs</code>	2
4.1.1	Description détaillée	2
4.1.2	Documentation du type de l'énumération	2
4.1.3	Documentation des fonctions	3
5	Documentation des fichiers	4
5.1	Référence du fichier <code>formoption.h</code>	4
5.1.1	Description détaillée	5
5.2	Référence du fichier <code>sex.h</code>	5
5.2.1	Description détaillée	5
Index		7

1 Une énumération pour représenter le sexe d'une personne

Un bon point d'entrée est celui de la documentation de l'espace de nom `nvs`.

2 Index des espaces de nommage

2.1 Liste des espaces de nommage

Liste de tous les espaces de nommage documentés avec une brève description :

`nvs`

Espace de nom de Nicolas Vansteenkiste

2

3 Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

formoption.h	
Définition de l'énumération fortement typée nvs : :FormOption	4
sex.h	
Définition de l'énumération fortement typée nvs : :Sex et de diverses fonctions relatives	5

4 Documentation des espaces de nommage

4.1 Référence de l'espace de nommage nvs

Espace de nom de Nicolas Vansteenkiste.

Énumérations

- enum **Sex** {
 Sex : :UNKNOWN, **Sex : :FEMALE**, **Sex : :MALE**, **Sex : :VOID**,
 Sex : :OTHER }
 Énumération fortement typée pour représenter le sexe d'une personne.
- enum **FormOption** { **FormOption : :NONE** = 0b0000, **FormOption : :SHORT** = 0b0001, **FormOption : :LONG** = 0b0010 }
 Énumération pour spécifier les options de mise en forme d'impression ou de conversion binaire → chaîne de caractères.

Fonctions

- **std : :ostream & operator<<** (**std : :ostream** &out, **Sex** in)
 *Opérateur d'injection d'une variable ou valeur **nvs : :Sex** dans un flux en sortie.*
- **std : :string to_string** (**Sex** sex, **FormOption** fo=**FormOption : :SHORT**)
 *Fonction de conversion d'une variable / valeur **nvs : :Sex** en **std : :string**.*

4.1.1 Description détaillée

Espace de nom de Nicolas Vansteenkiste.

4.1.2 Documentation du type de l'énumération

4.1.2.1 FormOption

```
enum nvs::FormOption [strong]
```

Énumération pour spécifier les options de mise en forme d'impression ou de conversion binaire → chaîne de caractères.

Valeurs énumérées

NONE	Absence de spécification.
SHORT	Forme courte.
LONG	Forme longue.

4.1.2.2 Sex

```
enum nvs::Sex [strong]
```

Énumération fortement typée pour représenter le sexe d'une personne.

Valeurs énumérées

UNKNOWN	Indique que la valeur du sexe est inconnue.
FEMALE	Indique que la valeur du sexe est celle d'une femme.
MALE	Indique que la valeur du sexe est celle d'un homme.
VOID	Indique l'absence de valeur pour le sexe.
OTHER	Indique que la valeur du sexe n'est ni Sex : :UNKNOWN , ni Sex : :FEMALE , ni Sex : :MALE , ni Sex : :VOID .

4.1.3 Documentation des fonctions

4.1.3.1 operator<<()

```
std::ostream & nvs::operator<< (  
    std::ostream & out,  
    Sex in ) [inline]
```

Opérateur d'injection d'une variable ou valeur [nvs : :Sex](#) dans un flux en sortie.

La valeur injectée correspond à la *forme courte* retournée par la fonction [to_string\(Sex, FormOption\)](#).

Paramètres

<i>out</i>	le flux dans lequel réaliser l'injection.
<i>in</i>	la variable ou la valeur à injecter.

Renvoie

le flux après injection.

4.1.3.2 to_string()

```
std::string nvs::to_string (
    Sex sex,
    FormOption fo = FormOption::SHORT )
```

Fonction de conversion d'une variable / valeur `nvs::Sex` en `std::string`.

En forme courte (`fo` différent de `FormOption::LONG`), la `std::string` retournée est :

- `u` pour la valeur `Sex::UNKNOWN`;
- `f` pour la valeur `Sex::FEMALE`;
- `m` pour la valeur `Sex::MALE`;
- `v` pour la valeur `Sex::VOID`;
- `o` pour la valeur `Sex::OTHER`.

C'est la même forme que celle injectée par l'opérateur d'injection dans un flux en sortie.

En forme longue (`fo` explicitement égal à `FormOption::LONG`), la `std::string` retournée est :

- `unknown` pour la valeur `Sex::UNKNOWN`;
- `female` pour la valeur `Sex::FEMALE`;
- `male` pour la valeur `Sex::MALE`;
- `void` pour la valeur `Sex::VOID`;
- `other` pour la valeur `Sex::OTHER`.

Paramètres

<code>sex</code>	la valeur de <code>nvs::Sex</code> à convertir.
<code>fo</code>	pour contrôler la forme courte ou longue de la <code>std::string</code> retournée.

Renvoie

une représentation de la valeur de `sex` sous la forme d'une `std::string`.

5 Documentation des fichiers

5.1 Référence du fichier formoption.h

Définition de l'énumération fortement typée `nvs::FormOption`.

Espaces de nommage

- `nvs`
Espace de nom de Nicolas Vansteenkiste.

Énumérations

- enum `nvs::FormOption` { `nvs::FormOption::NONE` = 0b0000, `nvs::FormOption::SHORT` = 0b0001, `nvs::FormOption::LONG` = 0b0010 }
Énumération pour spécifier les options de mise en forme d'impression ou de conversion binaire → chaîne de caractères.

5.1.1 Description détaillée

Définition de l'énumération fortement typée `nvs : :FormOption`.

5.2 Référence du fichier sex.h

Définition de l'énumération fortement typée `nvs : :Sex` et de diverses fonctions relatives.

```
#include <ostream>
#include <string>
#include "../misc/formoption.h"
```

Espaces de nommage

- `nvs`
Espace de nom de Nicolas Vansteenkiste.

Énumérations

- enum `nvs : :Sex` {
 `nvs : :Sex : :UNKNOWN`, `nvs : :Sex : :FEMALE`, `nvs : :Sex : :MALE`, `nvs : :Sex : :VOID`,
 `nvs : :Sex : :OTHER` }
 Énumération fortement typée pour représenter le sexe d'une personne.

Fonctions

- `std : :ostream & nvs : :operator<< (std : :ostream &out, Sex in)`
 Opérateur d'injection d'une variable ou valeur `nvs : :Sex` dans un flux en sortie.
- `std : :string nvs : :to_string (Sex sex, FormOption fo=FormOption : :SHORT)`
 Fonction de conversion d'une variable / valeur `nvs : :Sex` en `std : :string`.

5.2.1 Description détaillée

Définition de l'énumération fortement typée `nvs : :Sex` et de diverses fonctions relatives.

Index

FormOption

nvs, [2](#)

formoption.h, [4](#)

nvs, [2](#)

FormOption, [2](#)

operator<<, [3](#)

Sex, [3](#)

to_string, [3](#)

operator<<

nvs, [3](#)

Sex

nvs, [3](#)

sex.h, [5](#)

to_string

nvs, [3](#)