

SchoolTools - Schedule
live

Wygenerowano przez Doxygen 1.6.3

Tue Jun 1 21:53:32 2010

Spis treści

1	Struktura katalogów	1
1.1	Katalogi	1
2	Indeks przestrzeni nazw	3
2.1	Lista przestrzeni nazw	3
3	Indeks klas	5
3.1	Hierarchia klas	5
4	Indeks klas	7
4.1	Lista klas	7
5	Indeks plików	9
5.1	Lista plików	9
6	Dokumentacja katalogów	11
6.1	Dokumentacja katalogu build/	11
6.2	Dokumentacja katalogu build/CMakeFiles/	12
6.3	Dokumentacja katalogu build/CMakeFiles/CompilerIdC/	13
6.4	Dokumentacja katalogu build/CMakeFiles/CompilerIdCXX/	14
6.5	Dokumentacja katalogu include/	15
6.6	Dokumentacja katalogu src/obsolete/	16
6.7	Dokumentacja katalogu include/obsolete/	17
6.8	Dokumentacja katalogu src/	18
7	Dokumentacja przestrzeni nazw	19
7.1	Dokumentacja przestrzeni nazw obsolete	19
7.1.1	Dokumentacja funkcji	20
7.1.1.1	operator<<	20
7.1.1.2	operator>>	20

8 Dokumentacja klas	21
8.1 Dokumentacja klasy <code>obsolete::AbstractID</code>	21
8.1.1 Opis szczegółowy	22
8.1.2 Dokumentacja konstruktora i destruktora	23
8.1.2.1 <code>AbstractID</code>	23
8.1.3 Dokumentacja przyjaciół i funkcji związanych	23
8.1.3.1 operator<<	23
8.1.3.2 operator>>	23
8.1.4 Dokumentacja atrybutów składowych	23
8.1.4.1 ID	23
8.2 Dokumentacja klasy <code>obsolete::AbstractIDGenerator</code>	24
8.2.1 Opis szczegółowy	26
8.2.2 Dokumentacja konstruktora i destruktora	26
8.2.2.1 <code>AbstractIDGenerator</code>	26
8.2.3 Dokumentacja funkcji składowych	26
8.2.3.1 <code>createID</code>	26
8.2.3.2 <code>createVoidID</code>	26
8.2.4 Dokumentacja atrybutów składowych	26
8.2.4.1 ID	26
8.3 Dokumentacja klasy <code>obsolete::AbstractIncrementalID</code>	27
8.3.1 Opis szczegółowy	29
8.3.2 Dokumentacja konstruktora i destruktora	29
8.3.2.1 <code>AbstractIncrementalID</code>	29
8.3.2.2 <code>AbstractIncrementalID</code>	29
8.3.3 Dokumentacja atrybutów składowych	29
8.3.3.1 ID	29
8.4 Dokumentacja klasy <code>obsolete::AbstractReservedID</code>	30
8.4.1 Opis szczegółowy	32
8.4.2 Dokumentacja konstruktora i destruktora	32
8.4.2.1 <code>AbstractReservedID</code>	32
8.4.2.2 <code>AbstractReservedID</code>	32
8.4.3 Dokumentacja funkcji składowych	32
8.4.3.1 <code>createID</code>	32
8.4.4 Dokumentacja atrybutów składowych	32
8.4.4.1 ID	32
8.5 Dokumentacja klasy <code>Event</code>	33

8.5.1	Opis szczegółowy	34
8.5.2	Dokumentacja konstruktora i destruktor	34
8.5.2.1	Event	34
8.5.2.2	Event	35
8.5.3	Dokumentacja przyjaciół i funkcji związanych	35
8.5.3.1	operator<<	35
8.5.3.2	operator>>	35
8.5.4	Dokumentacja atrybutów składowych	35
8.5.4.1	subjectID	35
8.5.4.2	terms	36
8.5.4.3	typeID	36
8.6	Dokumentacja klasy obsolete::EventID	37
8.6.1	Opis szczegółowy	39
8.6.2	Dokumentacja konstruktora i destruktor	39
8.6.2.1	EventID	39
8.6.2.2	EventID	39
8.6.3	Dokumentacja atrybutów składowych	39
8.6.3.1	ID	39
8.7	Dokumentacja klasy obsolete::EventIDGenerator	40
8.7.1	Opis szczegółowy	42
8.7.2	Dokumentacja funkcji składowych	42
8.7.2.1	createID	42
8.7.2.2	createVoidID	42
8.7.3	Dokumentacja atrybutów składowych	42
8.7.3.1	ID	42
8.8	Dokumentacja klasy Events	43
8.8.1	Opis szczegółowy	44
8.8.2	Dokumentacja konstruktora i destruktor	45
8.8.2.1	Events	45
8.8.3	Dokumentacja funkcji składowych	45
8.8.3.1	addEvent	45
8.8.3.2	create	45
8.8.4	Dokumentacja przyjaciół i funkcji związanych	46
8.8.4.1	operator<<	46
8.8.4.2	operator>>	46
8.8.5	Dokumentacja atrybutów składowych	46

8.8.5.1	eventIDGenerator	46
8.8.5.2	events	47
8.8.5.3	existOne	47
8.8.5.4	initialized	47
8.9	Dokumentacja klasy GeneratorIDGenerator	48
8.9.1	Opis szczegółowy	49
8.9.2	Dokumentacja konstruktora i destruktor	49
8.9.2.1	GeneratorIDGenerator	49
8.9.3	Dokumentacja funkcji składowych	49
8.9.3.1	nextIDGenerator	49
8.9.3.2	voidIDGenerator	49
8.9.4	Dokumentacja atrybutów składowych	50
8.9.4.1	freeIDGenerator	50
8.10	Dokumentacja klasy IDGenerator::ID	51
8.10.1	Opis szczegółowy	52
8.10.2	Dokumentacja konstruktora i destruktor	52
8.10.2.1	ID	52
8.10.2.2	ID	52
8.10.3	Dokumentacja funkcji składowych	53
8.10.3.1	operator=	53
8.10.4	Dokumentacja przyjaciół i funkcji związanych	53
8.10.4.1	IDGenerator	53
8.10.4.2	operator<<	53
8.10.4.3	operator>>	53
8.10.5	Dokumentacja atrybutów składowych	54
8.10.5.1	id	54
8.10.5.2	type	54
8.11	Dokumentacja klasy obsolete::ID	55
8.11.1	Opis szczegółowy	57
8.11.2	Dokumentacja funkcji składowych	57
8.11.2.1	createID	57
8.11.2.2	createID	57
8.11.3	Dokumentacja atrybutów składowych	57
8.11.3.1	ID	57
8.11.3.2	ID	57
8.11.3.3	ID	58

8.11.3.4	ID	58
8.12	Dokumentacja klasy IDGenerator	59
8.12.1	Opis szczegółowy	60
8.12.2	Dokumentacja konstruktora i destruktora	61
8.12.2.1	IDGenerator	61
8.12.3	Dokumentacja funkcji składowych	61
8.12.3.1	initialize	61
8.12.3.2	nextID	61
8.12.3.3	operator=	62
8.12.3.4	voidID	62
8.12.4	Dokumentacja przyjaciół i funkcji związanych	62
8.12.4.1	GeneratorIDGenerator	62
8.12.4.2	operator<<	62
8.12.4.3	operator>>	63
8.12.5	Dokumentacja atrybutów składowych	63
8.12.5.1	freeID	63
8.12.5.2	type	63
8.13	Dokumentacja klasy obsolete::MarkID	64
8.13.1	Opis szczegółowy	66
8.13.2	Dokumentacja konstruktora i destruktora	66
8.13.2.1	MarkID	66
8.13.2.2	MarkID	66
8.13.3	Dokumentacja atrybutów składowych	66
8.13.3.1	ID	66
8.14	Dokumentacja klasy obsolete::MarkIDGenerator	67
8.14.1	Opis szczegółowy	69
8.14.2	Dokumentacja funkcji składowych	69
8.14.2.1	createID	69
8.14.2.2	createVoidID	69
8.14.3	Dokumentacja atrybutów składowych	69
8.14.3.1	ID	69
8.15	Dokumentacja klasy QDate	70
8.16	Dokumentacja klasy obsolete::State	71
8.16.1	Opis szczegółowy	72
8.16.2	Dokumentacja konstruktora i destruktora	72
8.16.2.1	State	72

8.16.3	Dokumentacja atrybutów składowych	72
8.16.3.1	subjectIDGenerator	72
8.16.3.2	subjectMapping	72
8.16.3.3	subjects	72
8.17	Dokumentacja klasy Subject	73
8.17.1	Opis szczegółowy	74
8.17.2	Dokumentacja atrybutów składowych	74
8.17.2.1	exist	74
8.17.2.2	name	74
8.17.2.3	typeIDGenerator	74
8.17.2.4	types	74
8.17.2.5	typesMapping	74
8.18	Dokumentacja klasy obsolete::SubjectID	75
8.18.1	Opis szczegółowy	77
8.18.2	Dokumentacja funkcji składowych	77
8.18.2.1	createID	77
8.18.3	Dokumentacja atrybutów składowych	77
8.18.3.1	ID	77
8.19	Dokumentacja klasy obsolete::SubjectIDGenerator	78
8.19.1	Opis szczegółowy	80
8.19.2	Dokumentacja funkcji składowych	80
8.19.2.1	createID	80
8.19.2.2	createVoidID	80
8.19.3	Dokumentacja atrybutów składowych	80
8.19.3.1	ID	80
8.20	Dokumentacja klasy Subjects	81
8.20.1	Opis szczegółowy	82
8.20.2	Dokumentacja funkcji składowych	82
8.20.2.1	create	82
8.20.3	Dokumentacja atrybutów składowych	83
8.20.3.1	existOne	83
8.20.3.2	initialized	83
8.20.3.3	subjectIDGenerator	83
8.20.3.4	subjectMapping	83
8.20.3.5	subjects	83
8.21	Dokumentacja klasy Term	84

8.21.1	Opis szczegółowy	86
8.21.2	Dokumentacja konstruktora i destruktora	86
8.21.2.1	Term	86
8.21.3	Dokumentacja przyjaciół i funkcji związanych	86
8.21.3.1	operator<<	86
8.21.3.2	operator>>	87
8.21.4	Dokumentacja atrybutów składowych	87
8.21.4.1	eventID	87
8.21.4.2	markID	87
8.21.4.3	revoke	87
8.22	Dokumentacja klasy Type	88
8.22.1	Opis szczegółowy	88
8.22.2	Dokumentacja atrybutów składowych	88
8.22.2.1	exist	88
8.22.2.2	name	89
8.23	Dokumentacja klasy obsolete::TypeID	90
8.23.1	Opis szczegółowy	92
8.23.2	Dokumentacja funkcji składowych	92
8.23.2.1	createID	92
8.23.3	Dokumentacja atrybutów składowych	92
8.23.3.1	ID	92
8.24	Dokumentacja klasy obsolete::TypeIDGenerator	93
8.24.1	Opis szczegółowy	95
8.24.2	Dokumentacja funkcji składowych	95
8.24.2.1	createID	95
8.24.2.2	createVoidID	95
8.24.3	Dokumentacja atrybutów składowych	95
8.24.3.1	ID	95
9	Dokumentacja plików	97
9.1	Dokumentacja pliku build/CMakeFiles/CompilerIdC/CMakeCCompilerId.c	97
9.1.1	Dokumentacja definicji	97
9.1.1.1	ARCHITECTURE_ID	97
9.1.1.2	COMPILER_ID	97
9.1.1.3	PLATFORM_ID	97
9.1.2	Dokumentacja funkcji	98
9.1.2.1	main	98

9.1.3	Dokumentacja zmiennych	98
9.1.3.1	info_arch	98
9.1.3.2	info_compiler	98
9.1.3.3	info_platform	98
9.2	Dokumentacja pliku build/CMakeFiles/CompilerIdCXX/CMakeCXXCompilerId.cpp	99
9.2.1	Dokumentacja definicji	99
9.2.1.1	ARCHITECTURE_ID	99
9.2.1.2	COMPILER_ID	99
9.2.1.3	PLATFORM_ID	99
9.2.2	Dokumentacja funkcji	99
9.2.2.1	main	99
9.2.3	Dokumentacja zmiennych	100
9.2.3.1	info_arch	100
9.2.3.2	info_compiler	100
9.2.3.3	info_platform	100
9.3	Dokumentacja pliku include/event.h	101
9.3.1	Dokumentacja funkcji	101
9.3.1.1	operator<<	101
9.3.1.2	operator>>	102
9.4	Dokumentacja pliku include/events.h	103
9.4.1	Dokumentacja funkcji	104
9.4.1.1	operator<<	104
9.4.1.2	operator>>	104
9.5	Dokumentacja pliku include/generatoridgenerator.h	105
9.6	Dokumentacja pliku include/idgenerator.h	106
9.6.1	Dokumentacja funkcji	107
9.6.1.1	operator<<	107
9.6.1.2	operator<<	107
9.6.1.3	operator>>	107
9.6.1.4	operator>>	108
9.7	Dokumentacja pliku include/obsolete/abstractid.h	109
9.8	Dokumentacja pliku include/obsolete/abstractidgenerator.h	110
9.9	Dokumentacja pliku include/obsolete/abstractincrementalid.h	111
9.10	Dokumentacja pliku include/obsolete/abstractreservedid.h	112
9.11	Dokumentacja pliku include/obsolete/eventid.h	113
9.12	Dokumentacja pliku include/obsolete/eventidgenerator.h	114

9.13	Dokumentacja pliku include/obsolete/id.h	115
9.14	Dokumentacja pliku include/obsolete/markid.h	116
9.15	Dokumentacja pliku include/obsolete/markidgenerator.h	117
9.16	Dokumentacja pliku include/obsolete/state.h	118
9.17	Dokumentacja pliku include/obsolete/subjectid.h	119
9.18	Dokumentacja pliku include/obsolete/subjectidgenerator.h	120
9.19	Dokumentacja pliku include/obsolete/typeid.h	121
9.20	Dokumentacja pliku include/obsolete/typeidgenerator.h	122
9.21	Dokumentacja pliku include/subject.h	123
9.22	Dokumentacja pliku include/subjects.h	124
9.23	Dokumentacja pliku include/term.h	125
9.23.1	Dokumentacja funkcji	125
9.23.1.1	operator<<	125
9.23.1.2	operator>>	126
9.24	Dokumentacja pliku include/type.h	127
9.25	Dokumentacja pliku main.cpp	128
9.25.1	Dokumentacja funkcji	128
9.25.1.1	main	128
9.26	Dokumentacja pliku src/event.cpp	130
9.26.1	Dokumentacja funkcji	130
9.26.1.1	operator<<	130
9.26.1.2	operator>>	130
9.27	Dokumentacja pliku src/events.cpp	132
9.27.1	Dokumentacja funkcji	132
9.27.1.1	operator<<	132
9.27.1.2	operator>>	133
9.28	Dokumentacja pliku src/generatoridgenerator.cpp	134
9.29	Dokumentacja pliku src/idgenerator.cpp	135
9.29.1	Dokumentacja funkcji	135
9.29.1.1	operator<<	135
9.29.1.2	operator<<	136
9.29.1.3	operator>>	136
9.29.1.4	operator>>	136
9.30	Dokumentacja pliku src/obsolete/abstractid.cpp	138
9.30.1	Dokumentacja funkcji	138
9.30.1.1	operator<<	138

9.30.1.2 operator>>	138
9.31 Dokumentacja pliku src/obsolete/abstractidgenerator.cpp	139
9.32 Dokumentacja pliku src/obsolete/abstractincrementalid.cpp	140
9.33 Dokumentacja pliku src/obsolete/abstractreservedid.cpp	141
9.34 Dokumentacja pliku src/obsolete/eventid.cpp	142
9.35 Dokumentacja pliku src/obsolete/eventidgenerator.cpp	143
9.36 Dokumentacja pliku src/obsolete/id.cpp	144
9.37 Dokumentacja pliku src/obsolete/markid.cpp	145
9.38 Dokumentacja pliku src/obsolete/markidgenerator.cpp	146
9.39 Dokumentacja pliku src/obsolete/state.cpp	147
9.40 Dokumentacja pliku src/obsolete/subjectid.cpp	148
9.41 Dokumentacja pliku src/obsolete/subjectidgenerator.cpp	149
9.42 Dokumentacja pliku src/obsolete/typeid.cpp	150
9.43 Dokumentacja pliku src/obsolete/typeidgenerator.cpp	151
9.44 Dokumentacja pliku src/subject.cpp	152
9.45 Dokumentacja pliku src/subjects.cpp	153
9.46 Dokumentacja pliku src/term.cpp	154
9.46.1 Dokumentacja funkcji	154
9.46.1.1 operator<<	154
9.46.1.2 operator>>	154
9.47 Dokumentacja pliku src/type.cpp	156

Rozdział 1

Struktura katalogów

1.1 Katalogi

Ta struktura katalogów jest posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

build	11
CMakeFiles	12
CompilerIdC	13
CompilerIdCXX	14
include	15
obsolete	17
src	18
obsolete	16

Rozdział 2

Indeks przestrzeni nazw

2.1 Lista przestrzeni nazw

Tutaj znajdują się wszystkie przestrzenie nazw wraz z ich krótkimi opisami:

obsolete	19
------------------------------------	----

Rozdział 3

Indeks klas

3.1 Hierarchia klas

Ta lista dziedziczenia posortowana jest z grubsza, choć nie całkowicie, alfabetycznie:

obsolete::AbstractID	21
obsolete::AbstractIDGenerator	24
obsolete::EventIDGenerator	40
obsolete::MarkIDGenerator	67
obsolete::SubjectIDGenerator	78
obsolete::TypeIDGenerator	93
obsolete::AbstractIncrementalID	27
obsolete::EventID	37
obsolete::ID	55
obsolete::MarkID	64
obsolete::ID	55
obsolete::AbstractReservedID	30
obsolete::SubjectID	75
obsolete::ID	55
obsolete::TypeID	90
obsolete::ID	55
Event	33
Events	43
GeneratorIDGenerator	48
IDGenerator::ID	51
IDGenerator	59
QDate	70
Term	84
obsolete::State	71
Subject	73
Subjects	81
Type	88

Rozdział 4

Indeks klas

4.1 Lista klas

Tutaj znajdują się klasy, struktury, unie i interfejsy wraz z ich krótkimi opisami:

obsolete::AbstractID (Klasa bazowa dla wszystkich typów identyfikatorowych)	21
obsolete::AbstractIDGenerator (Klasa bazowa obiektów-generatorów identyfikatorów)	24
obsolete::AbstractIncrementalID (Podstawa dla wszystkich klas identyfikatorów automatycznych)	27
obsolete::AbstractReservedID (Klasa reprezentująca identyfikatory rezerwowane z góry)	30
Event (Reprezentuje zdarzenie w postaci zdefiniowanej przez użytkownika)	33
obsolete::EventID (Klasa reprezentująca identyfikator zdarzenia)	37
obsolete::EventIDGenerator (Klasa generatorów generatorów identyfikatorów wydarzeń)	40
Events (Klasa przechowuje wszystkie wydarzenia i zarządza nimi)	43
GeneratorIDGenerator (Ta klasa odpowiada za generowanie generatorów identyfikatorów)	48
IDGenerator::ID (Obiekt reprezentuje dany identyfikator określonego typu)	51
obsolete::ID (Korzeń nowej idei uniwersalnego ID)	55
IDGenerator (Generator identyfikatorów określonego typu)	59
obsolete::MarkID (Klasa reprezentująca identyfikator oceny)	64
obsolete::MarkIDGenerator (Klasa generatorów generatorów identyfikatorów skutków wydarzeń)	67
QDate	70
obsolete::State (Klasa przechowująca wszelkie informacje o stanie aplikacji, bazy danych i danych)	71
Subject (Reprezentuje informacje o pojedynczym przedmiocie)	73
obsolete::SubjectID (Klasa reprezentująca identyfikator przedmiotu)	75
obsolete::SubjectIDGenerator (Klasa generatorów generatorów identyfikatorów przedmiotów)	78
Subjects (Obiekt klasy reprezentuje wszystkie przedmioty)	81
Term (Reprezentacja egzekucji wydarzenia)	84
Type	88
obsolete::TypeID (Klasa reprezentująca identyfikator typu wydarzenia)	90
obsolete::TypeIDGenerator (Generator generatorów obiektów TypeID)	93

Rozdział 5

Indeks plików

5.1 Lista plików

Tutaj znajduje się lista wszystkich plików z ich krótkimi opisami:

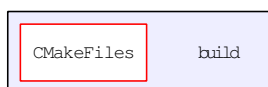
main.cpp	128
build/CMakeFiles/CompilerIdC/CMakeCCompilerId.c	97
build/CMakeFiles/CompilerIdCXX/CMakeCXXCompilerId.cpp	99
include/event.h	101
include/events.h	103
include/generatoridgenerator.h	105
include/idgenerator.h	106
include/subject.h	123
include/subjects.h	124
include/term.h	125
include/type.h	127
include/obsolete/abstractid.h	109
include/obsolete/abstractidgenerator.h	110
include/obsolete/abstractincrementalid.h	111
include/obsolete/abstractreservedid.h	112
include/obsolete/eventid.h	113
include/obsolete/eventidgenerator.h	114
include/obsolete/id.h	115
include/obsolete/markid.h	116
include/obsolete/markidgenerator.h	117
include/obsolete/state.h	118
include/obsolete/subjectid.h	119
include/obsolete/subjectidgenerator.h	120
include/obsolete/typeid.h	121
include/obsolete/typeidgenerator.h	122
src/event.cpp	130
src/events.cpp	132
src/generatoridgenerator.cpp	134
src/idgenerator.cpp	135
src/subject.cpp	152
src/subjects.cpp	153
src/term.cpp	154
src/type.cpp	156

src/obsolete/abstractid.cpp	138
src/obsolete/abstractidgenerator.cpp	139
src/obsolete/abstractincrementalid.cpp	140
src/obsolete/abstractreservedid.cpp	141
src/obsolete/eventid.cpp	142
src/obsolete/eventidgenerator.cpp	143
src/obsolete/id.cpp	144
src/obsolete/markid.cpp	145
src/obsolete/markidgenerator.cpp	146
src/obsolete/state.cpp	147
src/obsolete/subjectid.cpp	148
src/obsolete/subjectidgenerator.cpp	149
src/obsolete/typeid.cpp	150
src/obsolete/typeidgenerator.cpp	151

Rozdział 6

Dokumentacja katalogów

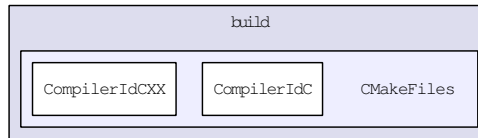
6.1 Dokumentacja katalogu build/



Katalogi

- katalog [CMakeFiles](#)

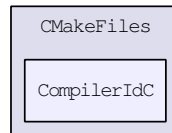
6.2 Dokumentacja katalogu build/CMakeFiles/



Katalogi

- katalog [CompilerIdC](#)
- katalog [CompilerIdCXX](#)

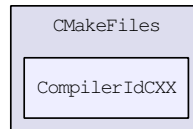
6.3 Dokumentacja katalogu build/CMakeFiles/CompilerIdC/



Pliki

- plik [CMakeCCompilerId.c](#)

6.4 Dokumentacja katalogu build/CMakeFiles/CompilerIdCXX/



Pliki

- plik [CMakeCXXCompilerId.cpp](#)

6.5 Dokumentacja katalogu include/



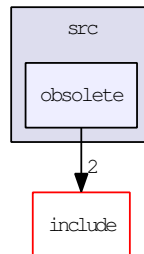
Katalogi

- katalog [obsolete](#)

Pliki

- plik [event.h](#)
- plik [events.h](#)
- plik [generatoridgenerator.h](#)
- plik [idgenerator.h](#)
- plik [subject.h](#)
- plik [subjects.h](#)
- plik [term.h](#)
- plik [type.h](#)

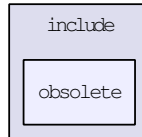
6.6 Dokumentacja katalogu src/obsolete/



Pliki

- plik [abstractid.cpp](#)
- plik [abstractidgenerator.cpp](#)
- plik [abstractincrementalid.cpp](#)
- plik [abstractreservedid.cpp](#)
- plik [eventid.cpp](#)
- plik [eventidgenerator.cpp](#)
- plik [id.cpp](#)
- plik [markid.cpp](#)
- plik [markidgenerator.cpp](#)
- plik [state.cpp](#)
- plik [subjectid.cpp](#)
- plik [subjectidgenerator.cpp](#)
- plik [typeid.cpp](#)
- plik [typeidgenerator.cpp](#)

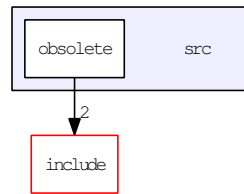
6.7 Dokumentacja katalogu include/obsolete/



Pliki

- plik [abstractid.h](#)
- plik [abstractidgenerator.h](#)
- plik [abstractincrementalid.h](#)
- plik [abstractreservedid.h](#)
- plik [eventid.h](#)
- plik [eventidgenerator.h](#)
- plik [id.h](#)
- plik [markid.h](#)
- plik [markidgenerator.h](#)
- plik [state.h](#)
- plik [subjectid.h](#)
- plik [subjectidgenerator.h](#)
- plik [typeid.h](#)
- plik [typeidgenerator.h](#)

6.8 Dokumentacja katalogu src/



Katalogi

- katalog [obsolete](#)

Pliki

- plik [event.cpp](#)
- plik [events.cpp](#)
- plik [generatoridgenerator.cpp](#)
- plik [idgenerator.cpp](#)
- plik [subject.cpp](#)
- plik [subjects.cpp](#)
- plik [term.cpp](#)
- plik [type.cpp](#)

Rozdział 7

Dokumentacja przestrzeni nazw

7.1 Dokumentacja przestrzeni nazw obsolete

Komponenty

- class [AbstractID](#)
Klasa bazowa dla wszystkich typów identyfikatorowych.
- class [AbstractIDGenerator](#)
Klasa bazowa obiektów-generatorów identyfikatorów.
- class [AbstractIncrementalID](#)
Podstawa dla wszystkich klas identyfikatorów automatycznych.
- class [AbstractReservedID](#)
Klasa reprezentująca identyfikatory rezerwowane z góry.
- class [EventID](#)
Klasa reprezentująca identyfikator zdarzenia.
- class [EventIDGenerator](#)
Klasa generatorów generatorów identyfikatorów wydarzeń.
- class [ID](#)
Korzeń nowej idei uniwersalnego [ID](#).
- class [MarkID](#)
Klasa reprezentująca identyfikator oceny.
- class [MarkIDGenerator](#)
Klasa generatorów generatorów identyfikatorów skutków wydarzeń.
- class [State](#)
Klasa przechowująca wszelkie informacje o stanie aplikacji, bazy danych i danych.

- class [SubjectID](#)
Klasa reprezentująca identyfikator przedmiotu.
- class [SubjectIDGenerator](#)
Klasa generatorów generatorów identyfikatorów przedmiotów.
- class [TypeID](#)
Klasa reprezentująca identyfikator typu wydarzenia.
- class [TypeIDGenerator](#)
Generator generatorów obiektów [TypeID](#).

Funkcje

- `QDataStream & operator<< (QDataStream &stream, const AbstractID &ID)`
Zapisuje dany identyfikator do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, AbstractID &ID)`
Odczytuje dany identyfikator z danego strumienia.

7.1.1 Dokumentacja funkcji

7.1.1.1 `QDataStream& obsolete::operator<< (QDataStream & stream, const AbstractID & ID)`

Zapisuje dany identyfikator do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany identyfikator.

ID Identyfikator który zostanie zapisany.

Zwraca

Ten sam strumień.

7.1.1.2 `QDataStream& obsolete::operator>> (QDataStream & stream, AbstractID & ID)`

Odczytuje dany identyfikator z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany identyfikator.

ID Identyfikator który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Rozdział 8

Dokumentacja klas

8.1 Dokumentacja klasy `obsolete::AbstractID`

Klasa bazowa dla wszystkich typów identyfikatorowych.

```
#include <abstractid.h>
```

Diagram dziedziczenia dla `obsolete::AbstractID`

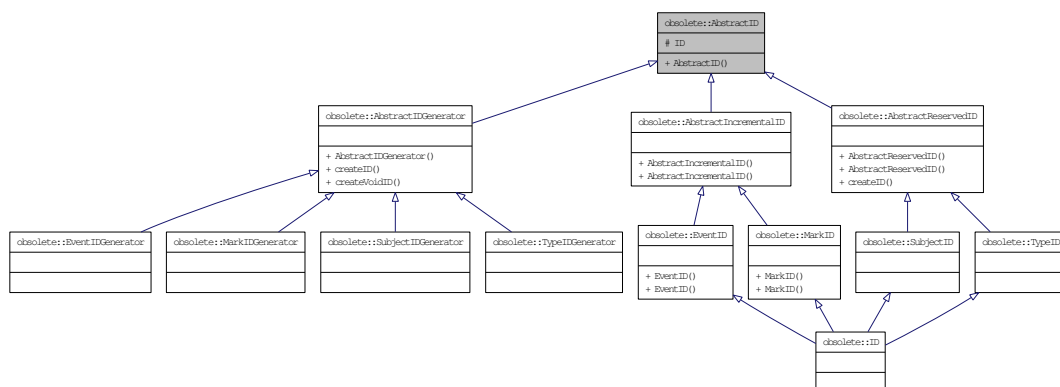
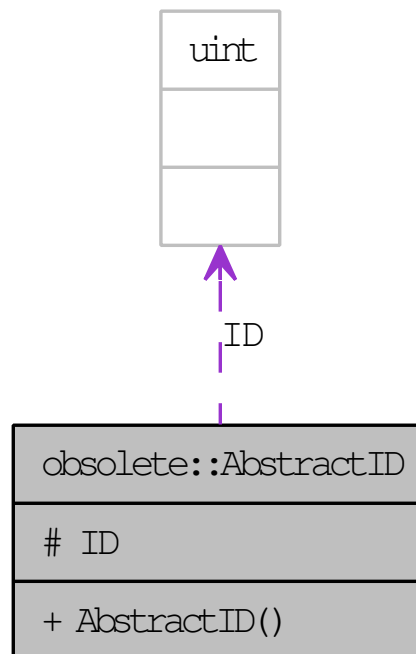


Diagram współpracy dla `obsolete::AbstractID`:



Metody publiczne

- `AbstractID (uint ID=0)`
Inicjalizuje nowy obiekt.

Atrybuty chronione

- `uint ID`
Reprezentowany `ID`.

Przyjaciele

- `QDataStream & operator<< (QDataStream &stream, const AbstractID &ID)`
Zapisuje dany identyfikator do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, AbstractID &ID)`
Odczytuje dany identyfikator z danego strumienia.

8.1.1 Opis szczegółowy

Klasa bazowa dla wszystkich typów identyfikatorowych.

Definicja w linii 11 pliku `abstractid.h`.

8.1.2 Dokumentacja konstruktora i destruktor

8.1.2.1 `obsolete::AbstractID::AbstractID (uint ID = 0) [inline]`

Inicjalizuje nowy obiekt.

Użytkownik powinien użyć metody `createdID()`.

Definicja w linii 31 pliku `abstractid.h`.

```
31 : ID ( ID ) {}
```

8.1.3 Dokumentacja przyjaciół i funkcji związanych

8.1.3.1 `QDataStream& operator<< (QDataStream & stream, const AbstractID & ID) [friend]`

Zapisuje dany identyfikator do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany identyfikator.

ID Identyfikator który zostanie zapisany.

Zwraca

Ten sam strumień.

8.1.3.2 `QDataStream& operator>> (QDataStream & stream, AbstractID & ID) [friend]`

Odczytuje dany identyfikator z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany identyfikator.

ID Identyfikator który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

8.1.4 Dokumentacja atrybutów składowych

8.1.4.1 `uint obsolete::AbstractID::ID [protected]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [include/obsolete/abstractid.h](#)

8.2 Dokumentacja klasy `obsolete::AbstractIDGenerator`

Klasa bazowa obiektów-generatorów identyfikatorów.

```
#include <abstractidgenerator.h>
```

Diagram dziedziczenia dla `obsolete::AbstractIDGenerator`

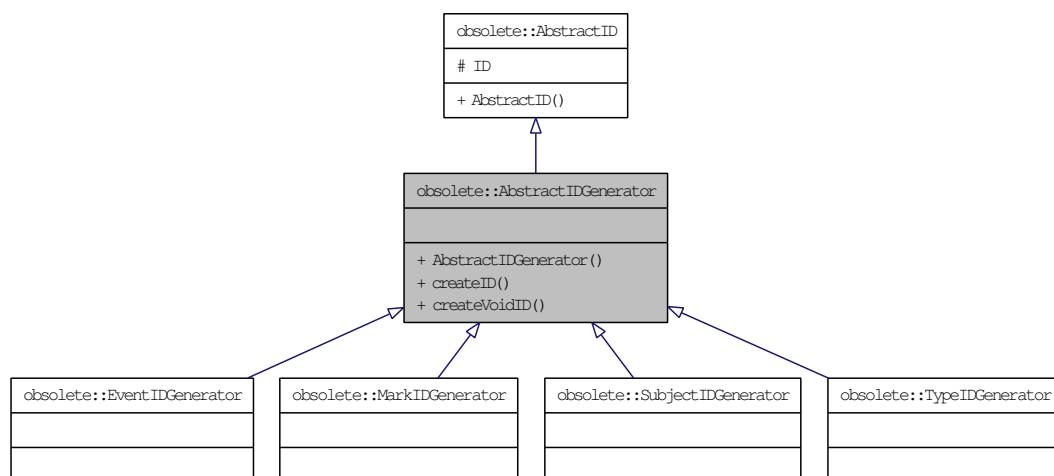
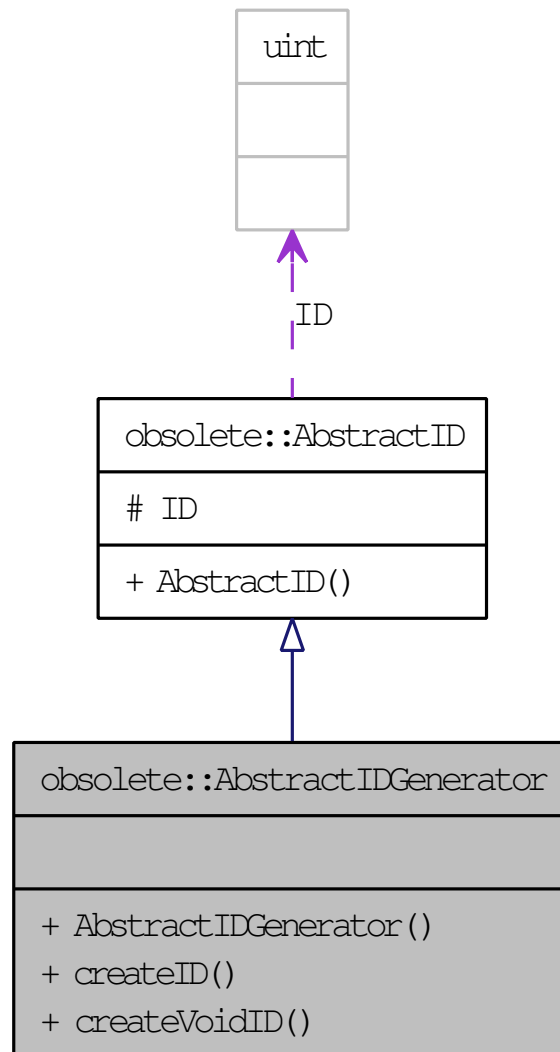


Diagram współpracy dla `obsolete::AbstractIDGenerator`:



Metody publiczne

- `AbstractIDGenerator (uint ID=1)`
Inicjalizuje wartość początkową generatora identyfikatorów.
- `virtual ID createID ()`
Tworzy nowy identyfikator.

Statyczne metody publiczne

- `static ID createVoidID ()`
Tworzy niezainicjowany identyfikator.

Atrybuty chronione

- `uint` [ID](#)
Reprezentowany [ID](#).

8.2.1 Opis szczegółowy

Klasa bazowa obiektów-generatorów identyfikatorów.

Definicja w linii 9 pliku `abstractidgenerator.h`.

8.2.2 Dokumentacja konstruktora i destruktor

8.2.2.1 `obsolete::AbstractIDGenerator::AbstractIDGenerator (uint ID = 1) [inline]`

Inicjalizuje wartość początkową generatora identyfikatorów.

Definicja w linii 8 pliku `abstractidgenerator.h`.

8.2.3 Dokumentacja funkcji składowych

8.2.3.1 `virtual ID obsolete::AbstractIDGenerator::createID () [virtual]`

Tworzy nowy identyfikator.

8.2.3.2 `static ID obsolete::AbstractIDGenerator::createVoidID () [static]`

Tworzy niezainicjowany identyfikator.

8.2.4 Dokumentacja atrybutów składowych

8.2.4.1 `uint obsolete::AbstractID::ID [protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `include/obsolete/abstractidgenerator.h`

8.3 Dokumentacja klasy `obsolete::AbstractIncrementalID`

Podstawa dla wszystkich klas identyfikatorów automatycznych.

```
#include <abstractincrementalid.h>
```

Diagram dziedziczenia dla `obsolete::AbstractIncrementalID`

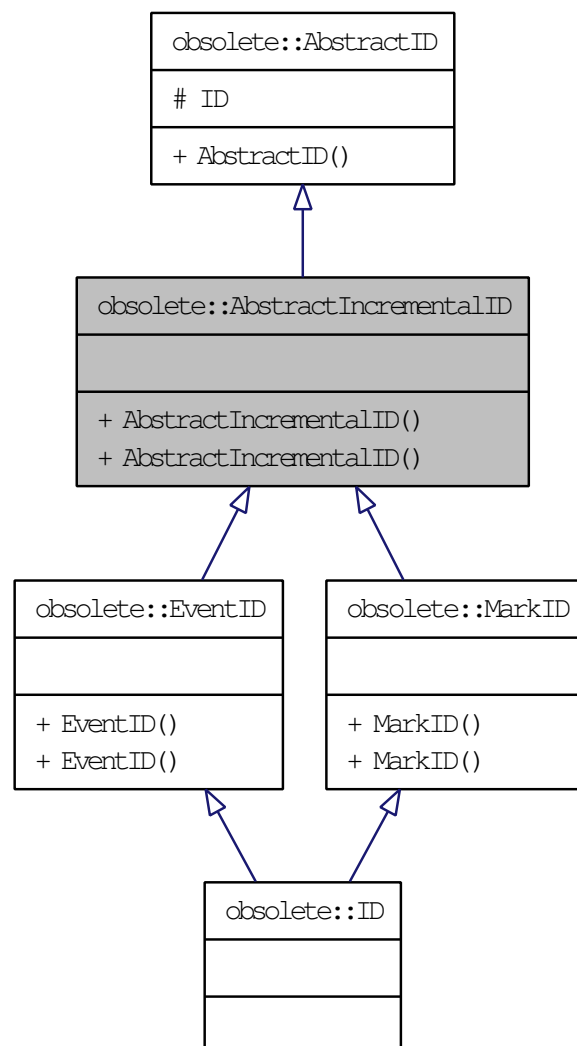
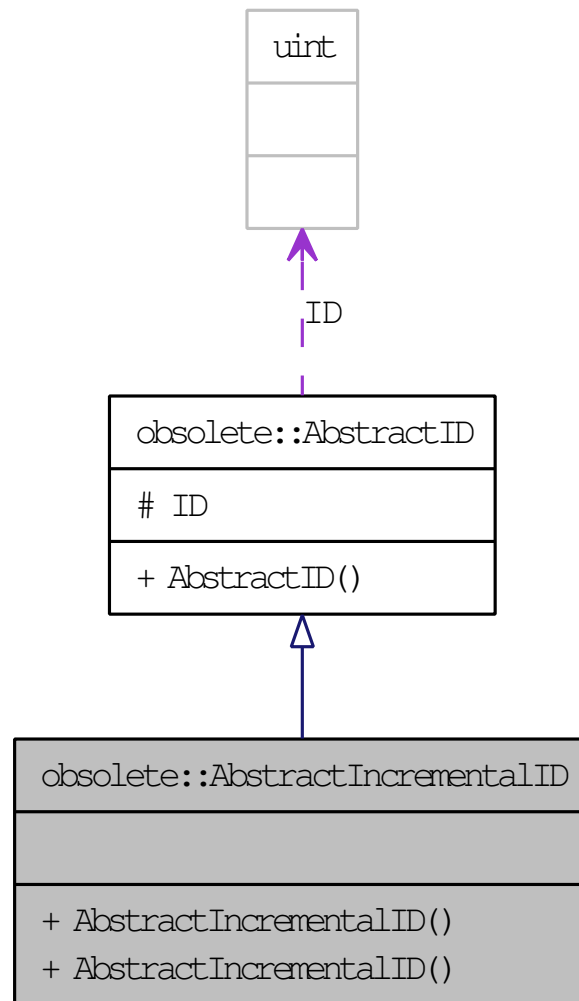


Diagram współpracy dla `obsolete::AbstractIncrementalID`:



Metody publiczne

- [AbstractIncrementalID \(uint ID\)](#)

Wykorzystuje konstruktor klasy nadrzędnej.

- [AbstractIncrementalID \(\)](#)

Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Atrybuty chronione

- `uint ID`

Reprezentowany [ID](#).

8.3.1 Opis szczegółowy

Podstawa dla wszystkich klas identyfikatorów automatycznych.

Definicja w linii 10 pliku `abstractincrementalid.h`.

8.3.2 Dokumentacja konstruktora i destruktora

8.3.2.1 `obsolete::AbstractIncrementalID::AbstractIncrementalID (uint ID) [inline]`

Wykorzystuje konstruktor klasy nadrzędnej.

Definicja w linii 13 pliku `abstractincrementalid.h`.

8.3.2.2 `obsolete::AbstractIncrementalID::AbstractIncrementalID () [inline]`

Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Definicja w linii 14 pliku `abstractincrementalid.h`.

8.3.3 Dokumentacja atrybutów składowych

8.3.3.1 `uint obsolete::AbstractID::ID [protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [include/obsolete/abstractincrementalid.h](#)

8.4 Dokumentacja klasy `obsolete::AbstractReservedID`

Klasa reprezentująca identyfikatory rezerwowane z góry.

```
#include <abstractreservedid.h>
```

Diagram dziedziczenia dla `obsolete::AbstractReservedID`

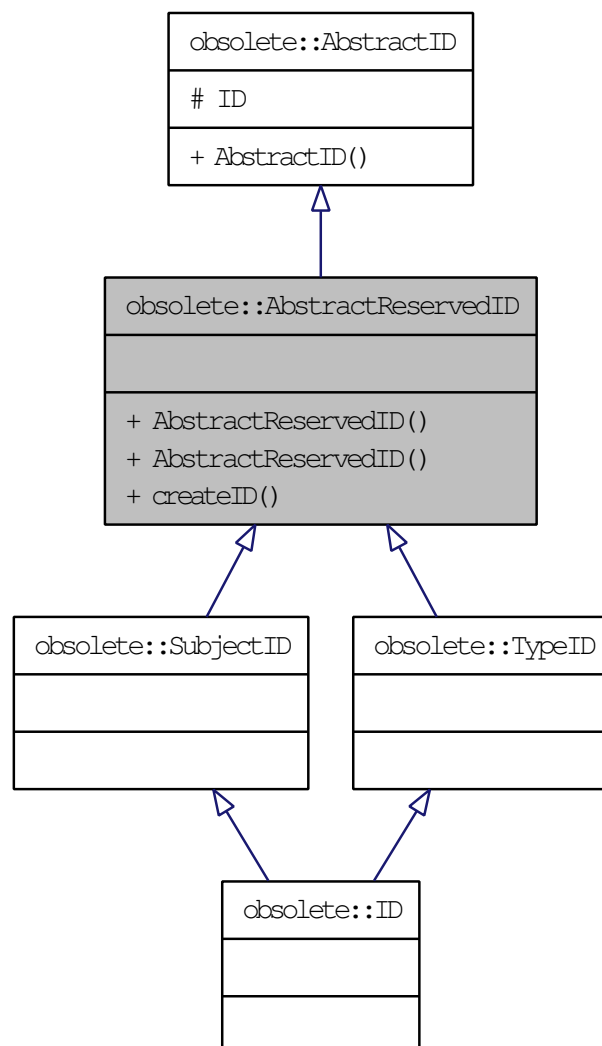
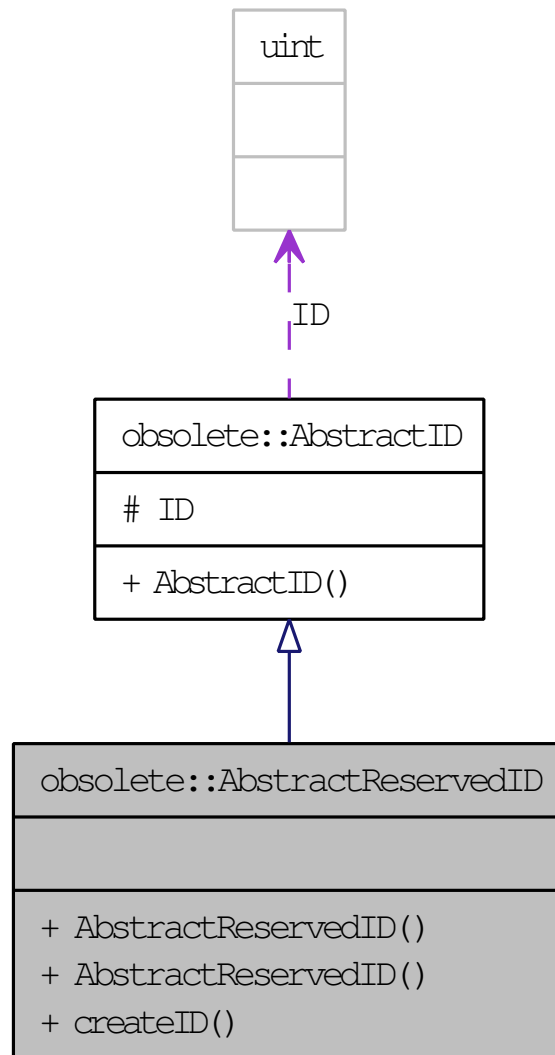


Diagram współpracy dla `obsolete::AbstractReservedID`:



Metody publiczne

- [AbstractReservedID](#) (`uint ID`)
Wykorzystuje konstruktor klasy nadrzędnej.
- [AbstractReservedID](#) ()
Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Statyczne metody publiczne

- `static` [AbstractReservedID createID](#) (`uint ID`)
Opakowuje numer porządkowy zadaną liczbą.

Atrybuty chronione

- uint [ID](#)
Reprezentowany [ID](#).

8.4.1 Opis szczegółowy

Klasa reprezentująca identyfikatory rezerwowane z góry. (coś w stylu haszowanie doskonałe)

Definicja w linii 10 pliku abstractreservedid.h.

8.4.2 Dokumentacja konstruktora i destruktora

8.4.2.1 `obsolete::AbstractReservedID::AbstractReservedID (uint ID)` [`inline`]

Wykorzystuje konstruktor klasy nadrzędnej.

Definicja w linii 13 pliku abstractreservedid.h.

8.4.2.2 `obsolete::AbstractReservedID::AbstractReservedID ()` [`inline`]

Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Definicja w linii 14 pliku abstractreservedid.h.

8.4.3 Dokumentacja funkcji składowych

8.4.3.1 `static AbstractReservedID obsolete::AbstractReservedID::createID (uint ID)` [`inline`, `static`]

Opakowuje numer porządkowy zadaną liczbą.

8.4.4 Dokumentacja atrybutów składowych

8.4.4.1 `uint obsolete::AbstractID::ID` [`protected`, `inherited`]

Reprezentowany [ID](#).

Definicja w linii 34 pliku abstractid.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

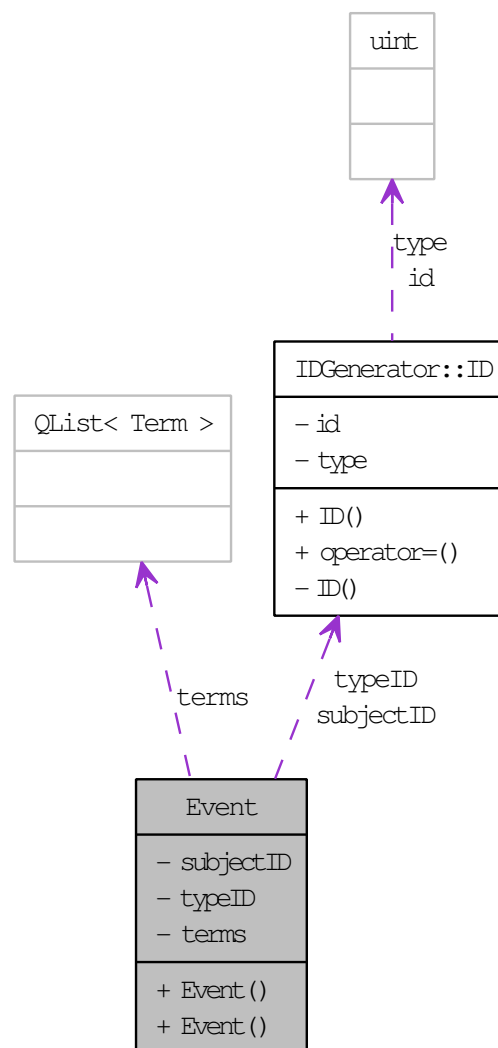
- `include/obsolete/abstractreservedid.h`

8.5 Dokumentacja klasy Event

Reprezentuje zdarzenie w postaci zdefiniowanej przez użytkownika.

```
#include <event.h>
```

Diagram współpracy dla Event:



Metody publiczne

- **Event** ([IDGenerator::ID](#) subjectID, [IDGenerator::ID](#) typeID, [QList< Term >](#) term=[QList< Term >\(\)](#))

Inicjalizuje dane wydarzenie niezbędnymi informacjami.

- **Event** ()

Umożliwia tworzenie niezainicjalizowanych obiektów.

Atrybuty prywatne

- `IDGenerator::ID subjectID`
Identyfikator przedmiotu skojarzonego z wydarzeniem.
- `IDGenerator::ID typeID`
Identyfikator typu wydarzenia.
- `QList< Term > terms`
Lista terminów danego wydarzenia.

Przyjaciele

- `QDataStream & operator<< (QDataStream &stream, const Event &event)`
Zapisuje informacje o danym wydarzeniu do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, Event &event)`
Odczytuje informacje o danym wydarzeniu z danego strumienia.

8.5.1 Opis szczegółowy

Reprezentuje zdarzenie w postaci zdefiniowanej przez użytkownika. Posiada określony typ (zdefiniowany przez użytkownika z dostępnych możliwości (zdefiniowanych przez użytkownika)). Powiązane z terminami.

Pozwala ustawić ocenę.

Definicja w linii 15 pliku event.h.

8.5.2 Dokumentacja konstruktora i destruktor

8.5.2.1 `Event::Event (IDGenerator::ID subjectID, IDGenerator::ID typeID, QList< Term > term = QList<Term> ()) [inline]`

Inicjalizuje dane wydarzenie niezbędnymi informacjami.

Parametry

subjectID Identyfikator przedmiotu związanego z wydarzeniem

typeID Identyfikator typu wydarzenia

term Lista terminów tego wydarzenia. Może być pusta.

Definicja w linii 30 pliku event.h.

8.5.2.2 Event::Event () [inline]

Umożliwia tworzenie niezainicjalizowanych obiektów.

Definicja w linii 33 pliku event.h.

8.5.3 Dokumentacja przyjaciół i funkcji związanych

8.5.3.1 QDataStream& operator<< (QDataStream & *stream*, const Event & *event*) [friend]

Zapisuje informacje o danym wydarzeniu do danego strumienia.

Parametry

stream Strumień do którego będą zapisywane dane.

event Wydarzenie które będzie zapisywane.

Zwraca

Ten sam strumień.

Definicja w linii 3 pliku event.cpp.

```
4 {  
5     return stream<<event.subjectID<<event.typeID<<event.terms;  
6 }
```

8.5.3.2 QDataStream& operator>> (QDataStream & *stream*, Event & *event*) [friend]

Odczytuje informacje o danym wydarzeniu z danego strumienia.

Parametry

stream Strumień z którego będą odczytywane dane.

event Wydarzenie które zostanie zainicjalizowane wczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 8 pliku event.cpp.

```
9 {  
10     return stream>>event.subjectID<<event.typeID<<event.terms;  
11 }
```

8.5.4 Dokumentacja atrybutów składowych

8.5.4.1 IDGenerator::ID Event::subjectID [private]

Identyfikator przedmiotu skojarzonego z wydarzeniem.

Definicja w linii 35 pliku event.h.

8.5.4.2 QList<Term> Event::terms [private]

Lista terminów danego wydarzenia.

Definicja w linii 37 pliku event.h.

8.5.4.3 IDGenerator::ID Event::typeID [private]

Identyfikator typu wydarzenia.

Definicja w linii 36 pliku event.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `include/event.h`

8.6 Dokumentacja klasy `obsolete::EventID`

Klasa reprezentująca identyfikator zdarzenia.

```
#include <eventid.h>
```

Diagram dziedziczenia dla `obsolete::EventID`

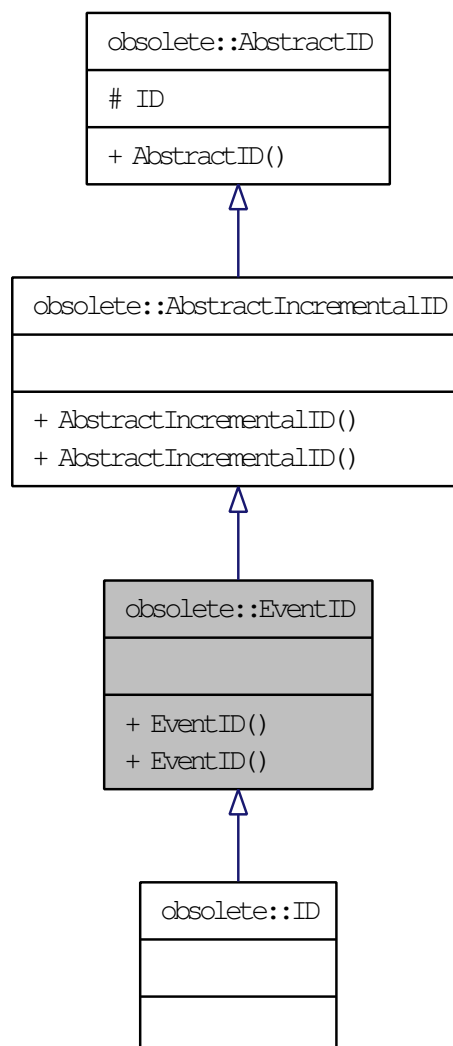
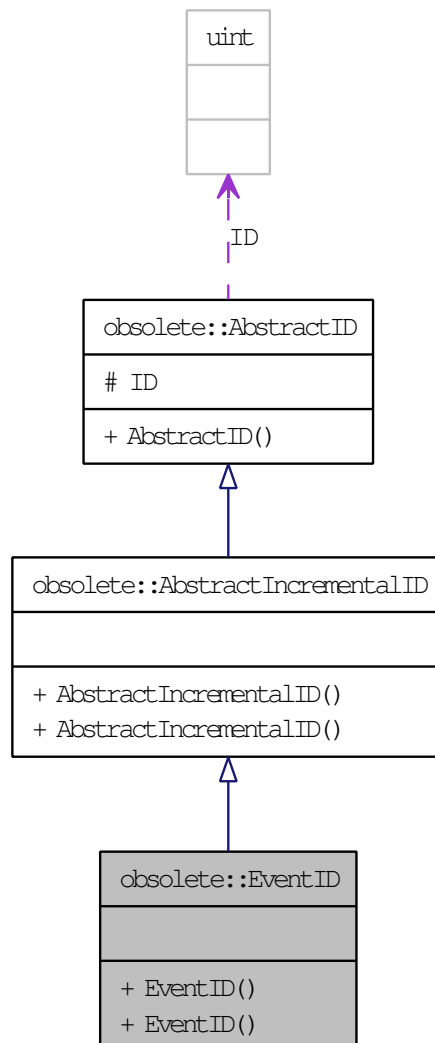


Diagram współpracy dla `obsolete::EventID`:



Metody publiczne

- `EventID (uint ID)`

Wykorzystuje konstruktor klasy nadrzędnej.

- `EventID ()`

Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Atrybuty chronione

- `uint ID`

Reprezentowany `ID`.

8.6.1 Opis szczegółowy

Klasa reprezentująca identyfikator zdarzenia.

Definicja w linii 10 pliku `eventid.h`.

8.6.2 Dokumentacja konstruktora i destruktor

8.6.2.1 `obsolete::EventID::EventID (uint ID) [inline]`

Wykorzystuje konstruktor klasy nadrzędnej.

Definicja w linii 13 pliku `eventid.h`.

8.6.2.2 `obsolete::EventID::EventID () [inline]`

Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Definicja w linii 14 pliku `eventid.h`.

8.6.3 Dokumentacja atrybutów składowych

8.6.3.1 `uint obsolete::AbstractID::ID [protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `include/obsolete/eventid.h`

8.7 Dokumentacja klasy `obsolete::EventIDGenerator`

Klasa generatorów generatorów identyfikatorów wydarzeń.

```
#include <eventidgenerator.h>
```

Diagram dziedziczenia dla `obsolete::EventIDGenerator`

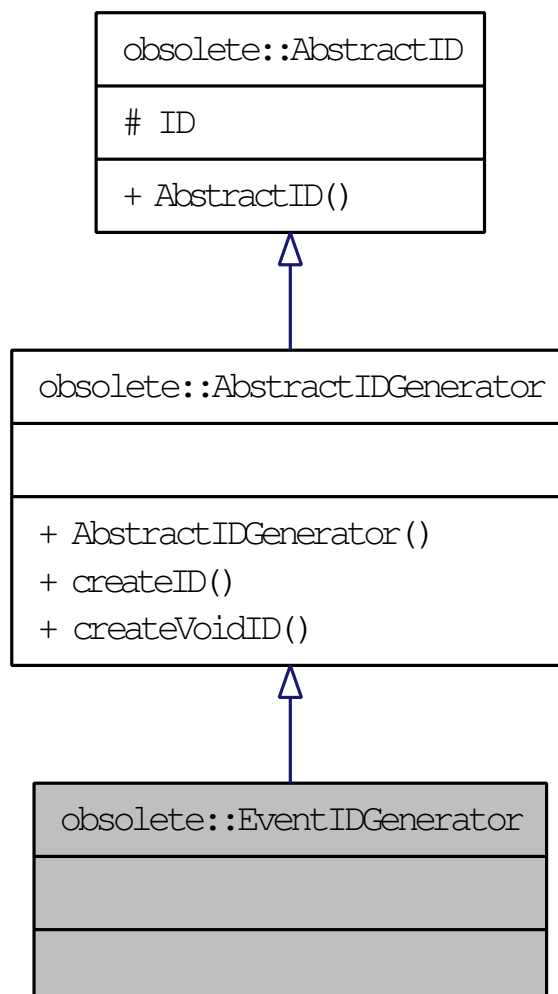
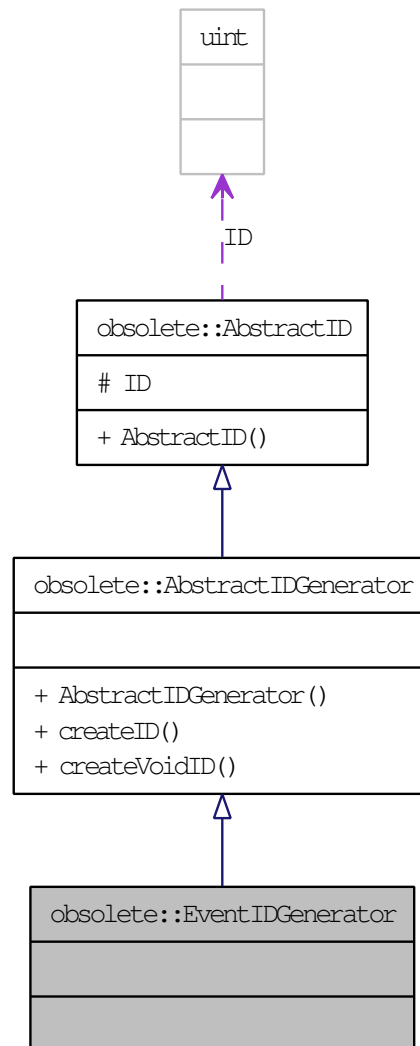


Diagram współpracy dla `obsolete::EventIDGenerator`:



Metody publiczne

- virtual `ID createID()`

Tworzy nowy identyfikator.

Statyczne metody publiczne

- static `ID createVoidID()`

Tworzy niezainicjowany identyfikator.

Atrybuty chronione

- uint [ID](#)
Reprezentowany [ID](#).

8.7.1 Opis szczegółowy

Klasa generatorów generatorów identyfikatorów wydarzeń.

Definicja w linii 8 pliku eventidgenerator.h.

8.7.2 Dokumentacja funkcji składowych

8.7.2.1 virtual ID obsolete::AbstractIDGenerator::createID () [virtual, inherited]

Tworzy nowy identyfikator.

8.7.2.2 static ID obsolete::AbstractIDGenerator::createVoidID () [static, inherited]

Tworzy niezainicjowany identyfikator.

8.7.3 Dokumentacja atrybutów składowych

8.7.3.1 uint obsolete::AbstractID::ID [protected, inherited]

Reprezentowany [ID](#).

Definicja w linii 34 pliku abstractid.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

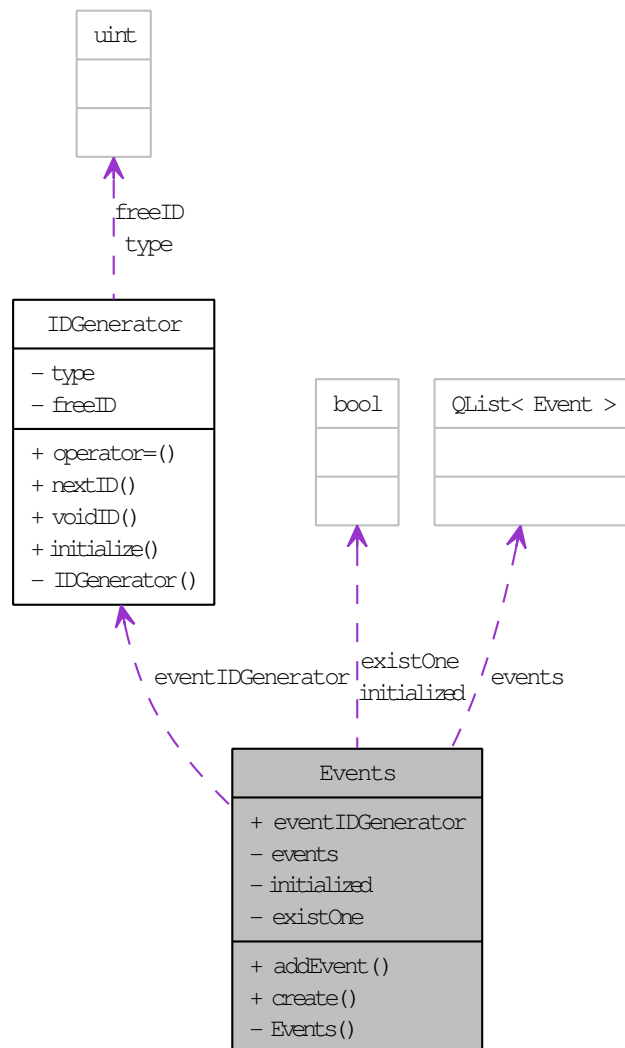
- include/obsolete/[eventidgenerator.h](#)

8.8 Dokumentacja klasy Events

Klasa przechowuje wszystkie wydarzenia i zarządza nimi.

```
#include <events.h>
```

Diagram współpracy dla Events:



Metody publiczne

- void `addEvent` (`Event` newEvent)

Dodaje nowe wydarzenie.

Statyczne metody publiczne

- static `Events` `create` () throw (char*)

Tworzy instancje klasy.

Atrybuty publiczne

- [IDGenerator](#) [eventIDGenerator](#)

Generator identyfikatorów wydarzeń.

Metody prywatne

- [Events](#) ()

Oznacza nowy obiekt jako niezainicjalizowany.

Atrybuty prywatne

- [QList< Event > events](#)

Lista wszystkich wydarzeń. Będzie się powiększać.

- bool [initialized](#)

Czy obiekt tej klasy został zainicjalizowany.

Statyczne atrybuty prywatne

- static bool [existOne](#)

Czy istnieje jakiś obiekt tej klasy.

Przyjaciele

- [QDataStream & operator<<](#) ([QDataStream &stream](#), const [Events &events](#))

Zapisuje informacje o wydarzeniach do danego strumienia.

- [QDataStream & operator>>](#) ([QDataStream &stream](#), [Events &events](#))

Odczytuje informacje o wydarzeniach z danego strumienia.

8.8.1 Opis szczegółowy

Klasa przechowuje wszystkie wydarzenia i zarządza nimi.

Definicja w linii 11 pliku events.h.

8.8.2 Dokumentacja konstruktora i destruktor

8.8.2.1 Events::Events () [inline, private]

Oznacza nowy obiekt jako niezainicjalizowany.

Definicja w linii 47 pliku events.h.

Odwołania w create().

50 :

Oto graf wywołań tej funkcji:



8.8.3 Dokumentacja funkcji składowych

8.8.3.1 void Events::addEvent (Event *newEvent*)

Dodaje nowe wydarzenie.

Parametry

newEvent Wydarzenie które ma zostać dodane do bazy danych.

Definicja w linii 15 pliku events.cpp.

Odwołuje się do events.

```
16 {  
17     Events::events<<newEvent;  
18 }
```

8.8.3.2 Events Events::create () throw (char*) [static]

Tworzy instancje klasy.

Wyjątki

*char** Jeżeli już wywołano tę metodę

Zwraca

Nowy obiekt tej klasy

Definicja w linii 20 pliku events.cpp.

Odwołuje się do Events() i existOne.

```
21 {  
22     if (existOne)  
23         throw "Obiekt już istnieje!";  
24     existOne=true;  
25     return Events();  
26 }
```

Oto graf wywołań dla tej funkcji:



8.8.4 Dokumentacja przyjaciół i funkcji związanych

8.8.4.1 QDataStream& operator<< (QDataStream & *stream*, const Events & *events*) [friend]

Zapisuje informacje o wydarzeniach do danego strumienia.

Parametry

stream Strumień do którego będą zapisywane dane.

events Wydarzenia które będzie zapisywane.

Zwraca

Ten sam strumień.

Definicja w linii 5 pliku events.cpp.

```

6 {
7     return stream<<events.eventIDGenerator<<events.events;
8 }
```

8.8.4.2 QDataStream& operator>> (QDataStream & *stream*, Events & *events*) [friend]

Odczytuje informacje o wydarzeniach z danego strumienia.

Parametry

stream Strumień z którego będą odczytywane dane.

events Obiekt klasy [Events](#) który zostanie zainicjalizowany wczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 10 pliku events.cpp.

```

11 {
12     return stream>>events.eventIDGenerator>>events.events;
13 }
```

8.8.5 Dokumentacja atrybutów składowych

8.8.5.1 IDGenerator Events::eventIDGenerator

Generator identyfikatorów wydarzeń.

Definicja w linii 24 pliku events.h.

Odwołania w operator<<() i operator>>().

8.8.5.2 QList<Event> Events::events [private]

Lista wszystkich wydarzeń. Będzie się powiększać.

Definicja w linii 26 pliku events.h.

Odwołania w addEvent(), operator<<() i operator>>().

8.8.5.3 bool Events::existOne [static, private]

Czy istnieje jakiś obiekt tej klasy.

Definicja w linii 29 pliku events.h.

Odwołania w create().

8.8.5.4 bool Events::initialized [private]

Czy obiekt tej klasy został zainicjalizowany.

Definicja w linii 28 pliku events.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

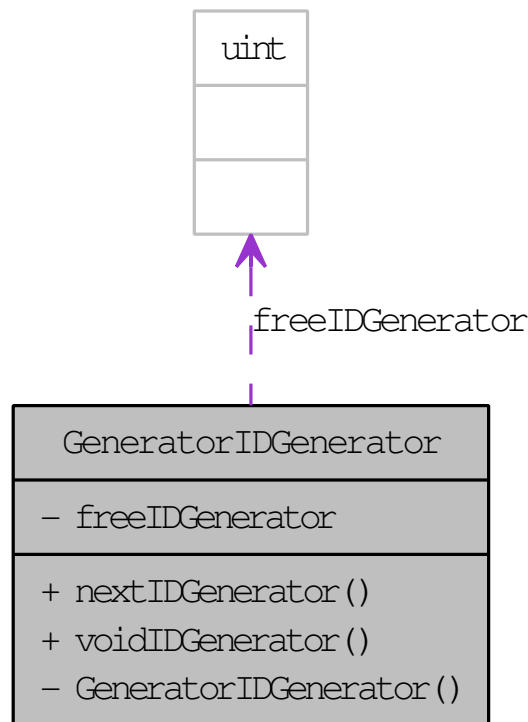
- [include/events.h](#)
- [src/events.cpp](#)

8.9 Dokumentacja klasy GeneratorIDGenerator

Ta klasa odpowiada za generowanie generatorów identyfikatorów.

```
#include <generatoridgenerator.h>
```

Diagram współpracy dla GeneratorIDGenerator:



Statyczne metody publiczne

- static `IDGenerator nextIDGenerator()`
Generuje następny generator identyfikatorów.
- static `IDGenerator voidIDGenerator()`
Generuje niezainicjalizowany (błędny) generator identyfikatorów.

Metody prywatne

- `GeneratorIDGenerator()`
Nie pozwalam na tworzenie obiektów tej klasy.

Statyczne atrybuty prywatne

- static uint `freeIDGenerator` = 1

Pierwszy dostępny typ Generators identyfikatorów.

8.9.1 Opis szczegółowy

Ta klasa odpowiada za generowanie generatorów identyfikatorów.

Definicja w linii 7 pliku generatoridgenerator.h.

8.9.2 Dokumentacja konstruktora i destruktor

8.9.2.1 GeneratorIDGenerator::GeneratorIDGenerator () [inline, private]

Nie pozwalam na tworzenie obiektów tej klasy.

Definicja w linii 20 pliku generatoridgenerator.h.

```
20 {}
```

8.9.3 Dokumentacja funkcji składowych

8.9.3.1 IDGenerator GeneratorIDGenerator::nextIDGenerator () [static]

Generuje następny generator identyfikatorów.

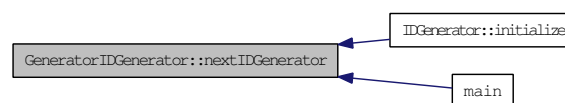
Definicja w linii 5 pliku generatoridgenerator.cpp.

Odwołuje się do freeIDGenerator.

Odwołania w IDGenerator::initialize() i main().

```
6 {  
7     return IDGenerator(freeIDGenerator++);  
8 }
```

Oto graf wywołań tej funkcji:



8.9.3.2 IDGenerator GeneratorIDGenerator::voidIDGenerator () [static]

Generuje niezainicjalizowany (błędny) generator identyfikatorów.

Definicja w linii 10 pliku generatoridgenerator.cpp.

```
11 {  
12     return IDGenerator(0);  
13 }
```

8.9.4 Dokumentacja atrybutów składowych

8.9.4.1 `uint GeneratorIDGenerator::freeIDGenerator = 1` `[static, private]`

Pierwszy dostępny typ Generатора identyfikatorów.

Definicja w linii 11 pliku `generatoridgenerator.h`.

Odwołania w `nextIDGenerator()`.

Dokumentacja dla tej klasy została wygenerowana z plików:

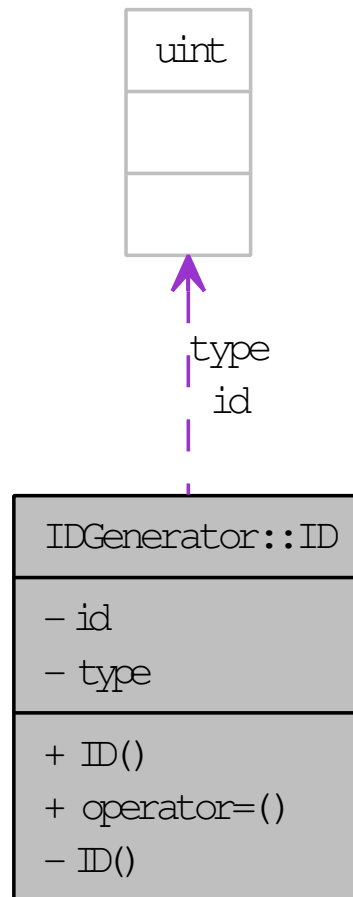
- [include/generatoridgenerator.h](#)
- [src/generatoridgenerator.cpp](#)

8.10 Dokumentacja klasy IDGenerator::ID

Obiekt reprezentuje dany identyfikator określonego typu.

```
#include <idgenerator.h>
```

Diagram współpracy dla IDGenerator::ID:



Metody publiczne

- `ID ()`
Konstruktor domyślny.
- `ID & operator= (const ID &o) throw (char*)`
Przypisuje wartość identyfikatora do pustego identyfikatora.

Metody prywatne

- `ID (uint id, uint type)`
Inicjalizuje identyfikator.

Atrybuty prywatne

- uint `id`
Reprezentowany ID.
- uint `type`
Typ tego identyfikatora.

Przyjaciele

- class `IDGenerator`
Identyfikator jest zaprzyjaźniony ze swoim stwórcą.
- `QDataStream & operator<< (QDataStream &stream, const ID &id)`
Zapisuje dany identyfikator do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, ID &id)`
Odczytuje dany identyfikator z danego strumienia.

8.10.1 Opis szczegółowy

Obiekt reprezentuje dany identyfikator określonego typu.

Definicja w linii 29 pliku `idgenerator.h`.

8.10.2 Dokumentacja konstruktora i destruktor

8.10.2.1 `IDGenerator::ID::ID () [inline]`

Konstruktor domyślny.

Definicja w linii 54 pliku `idgenerator.h`.

```
54 : id(0), type(0) {}
```

8.10.2.2 `IDGenerator::ID::ID (uint id, uint type) [inline, private]`

Inicjalizuje identyfikator.

Definicja w linii 61 pliku `idgenerator.h`.

```
61 : id(id), type(type) {}
```


8.10.3 Dokumentacja funkcji składowych

8.10.3.1 IDGenerator::ID & IDGenerator::ID::operator= (const ID & o) throw (char*)

Przypisuje wartość identyfikatora do pustego identyfikatora.

Definicja w linii 40 pliku idgenerator.cpp.

Odwołuje się do id.

```
41 {  
42     if (type!=0)  
43         throw "Nie wolno nadpisywać identyfikatorów!";  
44     id=o.id;  
45     type=o.type;  
46     return *this;  
47 }
```

8.10.4 Dokumentacja przyjaciół i funkcji związanych

8.10.4.1 friend class IDGenerator [friend]

Identyfikator jest zaprzyjaźniony ze swoim stwórcą.

Definicja w linii 32 pliku idgenerator.h.

8.10.4.2 QDataStream& operator<< (QDataStream & stream, const ID & id) [friend]

Zapisuje dany identyfikator do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany identyfikator.

id Identyfikator który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 20 pliku idgenerator.cpp.

```
21 {  
22     return stream<<id.type<<id.id;  
23 }
```

8.10.4.3 QDataStream& operator>> (QDataStream & stream, IDGenerator::ID & id) [friend]

Odczytuje dany identyfikator z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany identyfikator.

id Identyfikator który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 25 pliku idgenerator.cpp.

```
26 {  
27     return stream>>id.type>>id.id;  
28 }
```

8.10.5 Dokumentacja atrybutów składowych

8.10.5.1 uint IDGenerator::ID::id [private]

Reprezentowany [ID](#).

Definicja w linii 50 pliku idgenerator.h.

Odwołania w operator=().

8.10.5.2 uint IDGenerator::ID::type [private]

Typ tego identyfikatora.

Definicja w linii 51 pliku idgenerator.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [include/idgenerator.h](#)
- [src/idgenerator.cpp](#)

8.11 Dokumentacja klasy `obsolete::ID`

Korzeń nowej idei uniwersalnego [ID](#).

```
#include <id.h>
```

Diagram dziedziczenia dla `obsolete::ID`

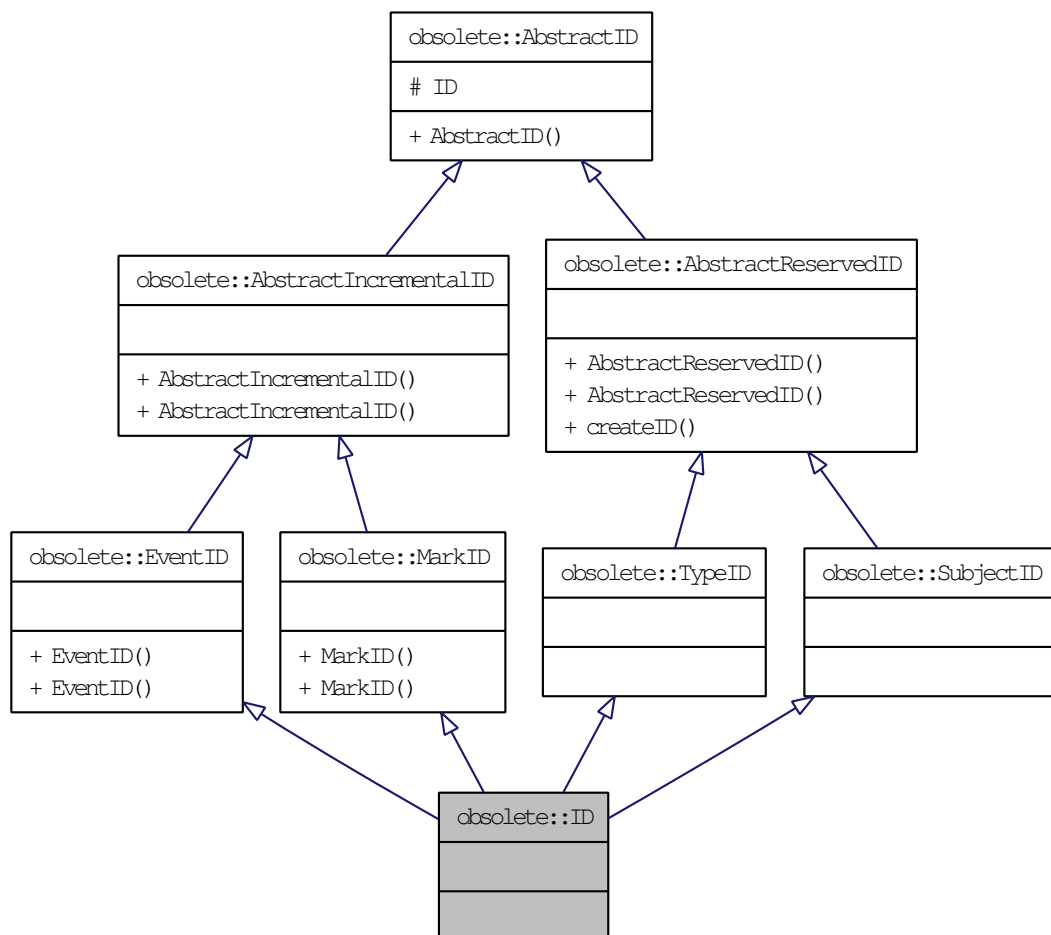
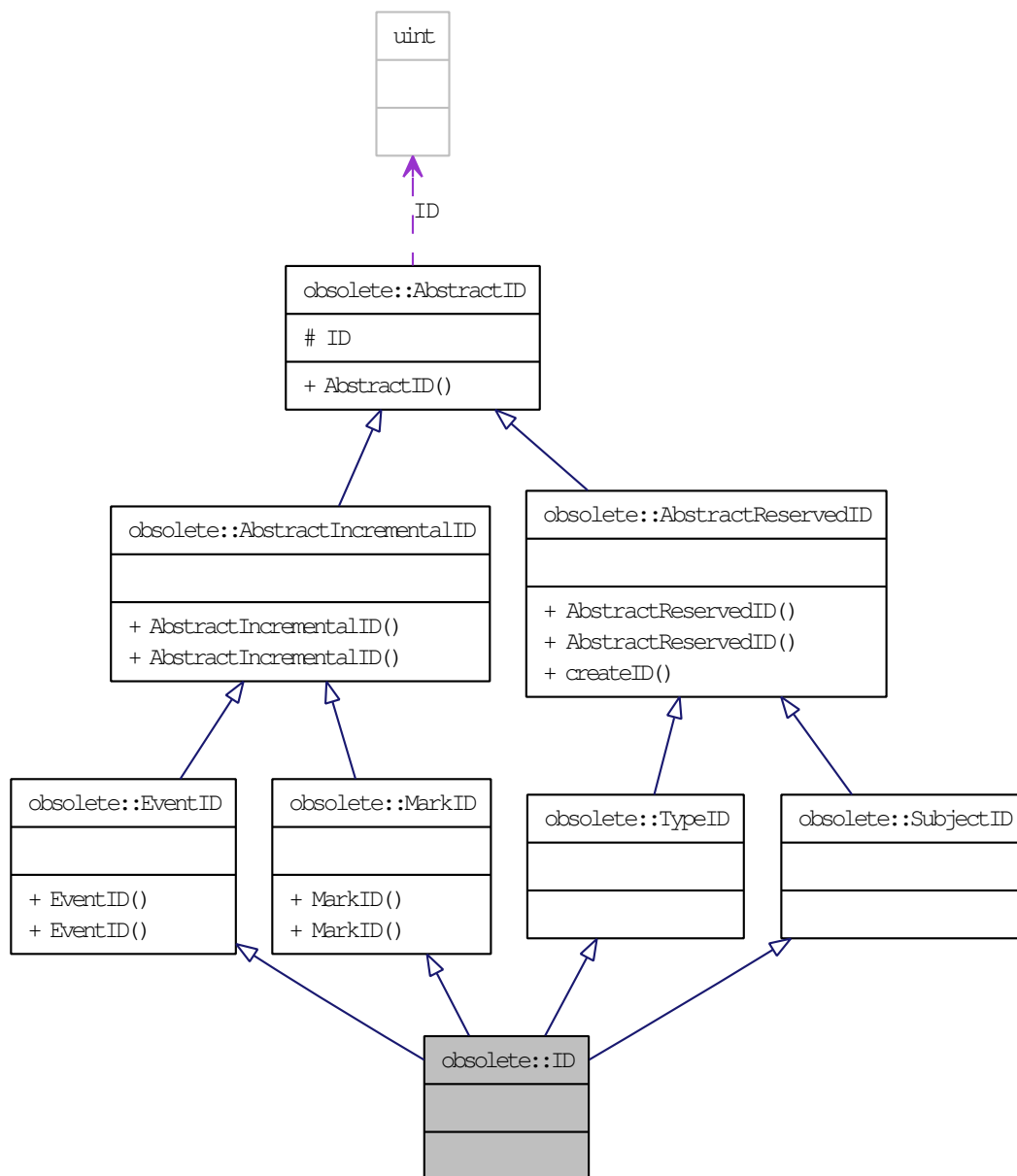


Diagram współpracy dla `obsolete::ID`:



Statyczne metody publiczne

- static `AbstractReservedID createID (uint ID)`
Opakowuje numer porządkowy zadaną liczbą.
- static `AbstractReservedID createID (uint ID)`
Opakowuje numer porządkowy zadaną liczbą.

Atrybuty chronione

- `uint ID`
Reprezentowany [ID](#).
- `uint ID`
Reprezentowany [ID](#).
- `uint ID`
Reprezentowany [ID](#).
- `uint ID`
Reprezentowany [ID](#).

8.11.1 Opis szczegółowy

Korzeń nowej idei uniwersalnego [ID](#).

Definicja w linii 13 pliku `id.h`.

8.11.2 Dokumentacja funkcji składowych

8.11.2.1 `static AbstractReservedID obsolete::AbstractReservedID::createID (uint ID)` `[inline, static, inherited]`

Opakowuje numer porządkowy zadaną liczbą.

8.11.2.2 `static AbstractReservedID obsolete::AbstractReservedID::createID (uint ID)` `[inline, static, inherited]`

Opakowuje numer porządkowy zadaną liczbą.

8.11.3 Dokumentacja atrybutów składowych

8.11.3.1 `uint obsolete::AbstractID::ID` `[protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

8.11.3.2 `uint obsolete::AbstractID::ID` `[protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

8.11.3.3 uint obsolete::AbstractID::ID [protected, inherited]

Reprezentowany [ID](#).

Definicja w linii 34 pliku abstractid.h.

8.11.3.4 uint obsolete::AbstractID::ID [protected, inherited]

Reprezentowany [ID](#).

Definicja w linii 34 pliku abstractid.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

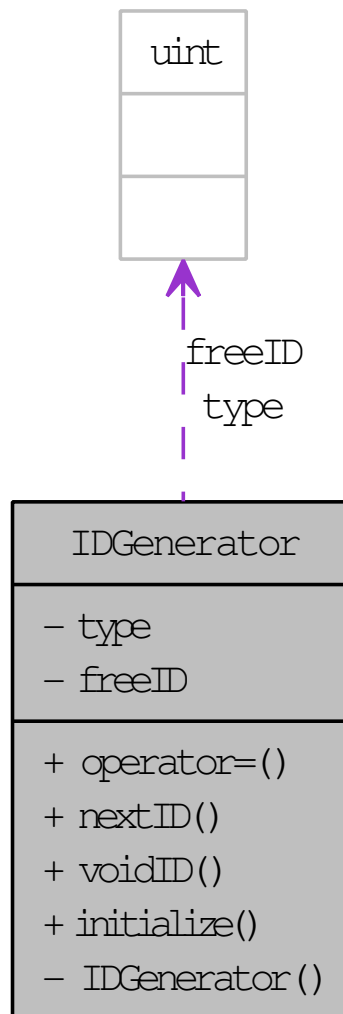
- [include/obsolete/id.h](#)

8.12 Dokumentacja klasy IDGenerator

Generator identyfikatorów określonego typu.

```
#include <idgenerator.h>
```

Diagram współpracy dla IDGenerator:



Komponenty

- class `ID`

Obiekt reprezentuje dany identyfikator określonego typu.

Metody publiczne

- `IDGenerator & operator= (const IDGenerator &o) throw (char*)`

Przeładowany operator przypisania dla upewnienia się, że żaden generator nie zostanie zniszczony.

- `ID nextID ()`

Generuje następny identyfikator.

- `ID voidID ()`

Generuje niezainicjalizowany identyfikator.

- `void initialize () throw (char*)`

Inicjalizuje dany generator identyfikatorów.

Metody prywatne

- `IDGenerator (uint T)`

Inicjalizuje dany generator.

Atrybuty prywatne

- `uint type`

Typ identyfikatorów generowanych przez dany generator.

- `uint freeID`

*Pierwszy wolny *ID*.*

Przyjaciele

- `class GeneratorIDGenerator`

Każdy generator identyfikatorów jest zaprzyjaźniony ze swoim stwórcą.

- `QDataStream & operator<< (QDataStream &stream, const IDGenerator &generator)`

Zapisuje dany generator identyfikatorów do danego strumienia.

- `QDataStream & operator>> (QDataStream &stream, IDGenerator &generator)`

Odczytuje dany generator identyfikatorów z danego strumienia.

8.12.1 Opis szczegółowy

Generator identyfikatorów określonego typu.

Definicja w linii 8 pliku idgenerator.h.

8.12.2 Dokumentacja konstruktora i destruktora

8.12.2.1 IDGenerator::IDGenerator (uint T) [inline, private]

Inicjalizuje dany generator.

Parametry

T Typ identyfikatorów generowanych przez ten generator.

Definicja w linii 91 pliku idgenerator.h.

```
91 : type(T), freeID(1) {}
```

8.12.3 Dokumentacja funkcji składowych

8.12.3.1 void IDGenerator::initialize () throw (char*)

Inicjalizuje dany generator identyfikatorów.

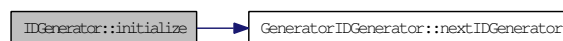
Inicjalizowany identyfikator MUSI być niezainicjalizowany!

Definicja w linii 4 pliku idgenerator.cpp.

Odwołuje się do GeneratorIDGenerator::nextIDGenerator() i type.

```
5 {  
6     if (type!=0)  
7         throw "Error";  
8     *this=GeneratorIDGenerator::nextIDGenerator();  
9 }
```

Oto graf wywołań dla tej funkcji:



8.12.3.2 ID IDGenerator::nextID () [inline]

Generuje następny identyfikator.

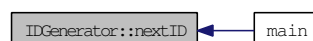
Definicja w linii 72 pliku idgenerator.h.

Odwołuje się do freeID i type.

Odwołania w main().

```
73 {  
74     return ID(freeID++,type);  
75 }
```

Oto graf wywoływań tej funkcji:



8.12.3.3 IDGenerator & IDGenerator::operator= (const IDGenerator & o) throw (char*)

Przeładowany operator przypisania dla upewnienia się, że żaden generator nie zostanie zniszczony.

Definicja w linii 11 pliku idgenerator.cpp.

Odwołuje się do type.

```
12 {
13     if (type!=0)
14         throw "Error";
15     type=o.type;
16     freeID=o.freeID;
17     return *this;
18 }
```

8.12.3.4 ID IDGenerator::voidID () [inline]

Generuje niezainicjalizowany identyfikator.

Definicja w linii 78 pliku idgenerator.h.

Odwołuje się do type.

```
79     {
80         return ID(0,type);
81     }
```

8.12.4 Dokumentacja przyjaciół i funkcji związanych

8.12.4.1 friend class GeneratorIDGenerator [friend]

Każdy generator identyfikatorów jest zaprzyjaźniony ze swoim stwórcą.

Definicja w linii 11 pliku idgenerator.h.

8.12.4.2 QDataStream& operator<< (QDataStream & stream, const IDGenerator & generator) [friend]

Zapisuje dany generator identyfikatorów do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany generator identyfikatorów.

generator Generator identyfikatorów który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 30 pliku idgenerator.cpp.

```
31 {
32     return stream<<generator.type<<generator.freeID;
33 }
```

8.12.4.3 QDataStream& operator>> (QDataStream & *stream*, IDGenerator & *generator*) [friend]

Odczytuje dane generator identyfikatorów z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany generator identyfikatorów.

generator Generator identyfikatorów który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 35 pliku idgenerator.cpp.

```
36 {  
37     return stream>>generator.type>>generator.freeID;  
38 }
```

8.12.5 Dokumentacja atrybutów składowych

8.12.5.1 uint IDGenerator::freeID [private]

Pierwszy wolny [ID](#).

Definicja w linii 65 pliku idgenerator.h.

Odwołania w nextID(), operator<<() i operator>>().

8.12.5.2 uint IDGenerator::type [private]

Typ identyfikatorów generowanych przez dany generator.

Definicja w linii 64 pliku idgenerator.h.

Odwołania w initialize(), nextID(), operator<<(), operator=(), operator>>() i voidID().

Dokumentacja dla tej klasy została wygenerowana z plików:

- [include/idgenerator.h](#)
- [src/idgenerator.cpp](#)

8.13 Dokumentacja klasy `obsolete::MarkID`

Klasa reprezentująca identyfikator oceny.

```
#include <markid.h>
```

Diagram dziedziczenia dla `obsolete::MarkID`

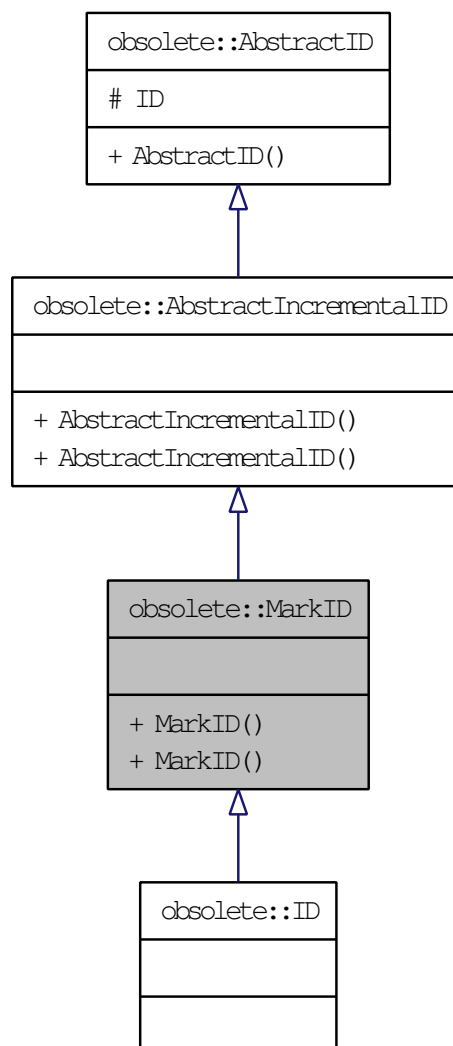
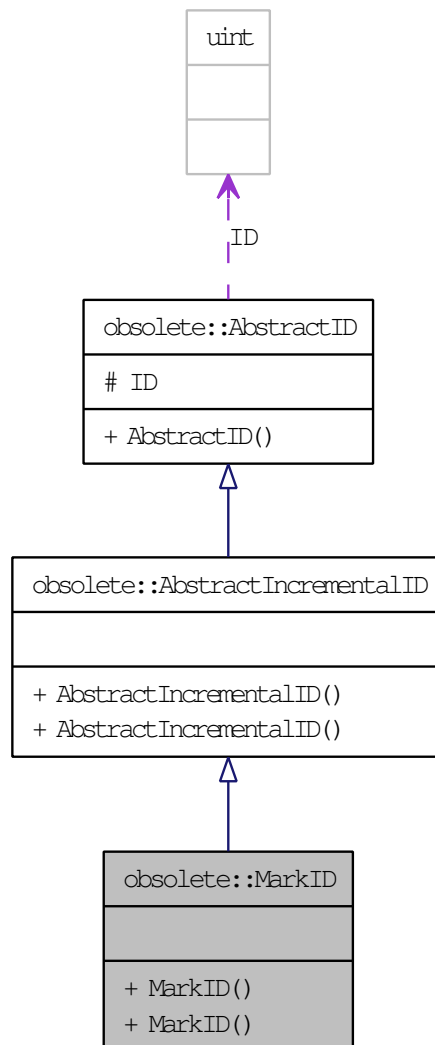


Diagram współpracy dla `obsolete::MarkID`:



Metody publiczne

- `MarkID (uint ID)`

Wykorzystuje konstruktor klasy nadrzędnej.

- `MarkID ()`

Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Atrybuty chronione

- `uint ID`

Reprezentowany `ID`.

8.13.1 Opis szczegółowy

Klasa reprezentująca identyfikator oceny.

Definicja w linii 10 pliku markid.h.

8.13.2 Dokumentacja konstruktora i destruktor

8.13.2.1 `obsolete::MarkID::MarkID (uint ID) [inline]`

Wykorzystuje konstruktor klasy nadrzędnej.

Definicja w linii 13 pliku markid.h.

8.13.2.2 `obsolete::MarkID::MarkID () [inline]`

Pozwalam również na tworzenie niezainicjalizowanych obiektów.

Definicja w linii 14 pliku markid.h.

8.13.3 Dokumentacja atrybutów składowych

8.13.3.1 `uint obsolete::AbstractID::ID [protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku abstractid.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `include/obsolete/markid.h`

8.14 Dokumentacja klasy `obsolete::MarkIDGenerator`

Klasa generatorów generatorów identyfikatorów skutków wydarzeń.

```
#include <markidgenerator.h>
```

Diagram dziedziczenia dla `obsolete::MarkIDGenerator`

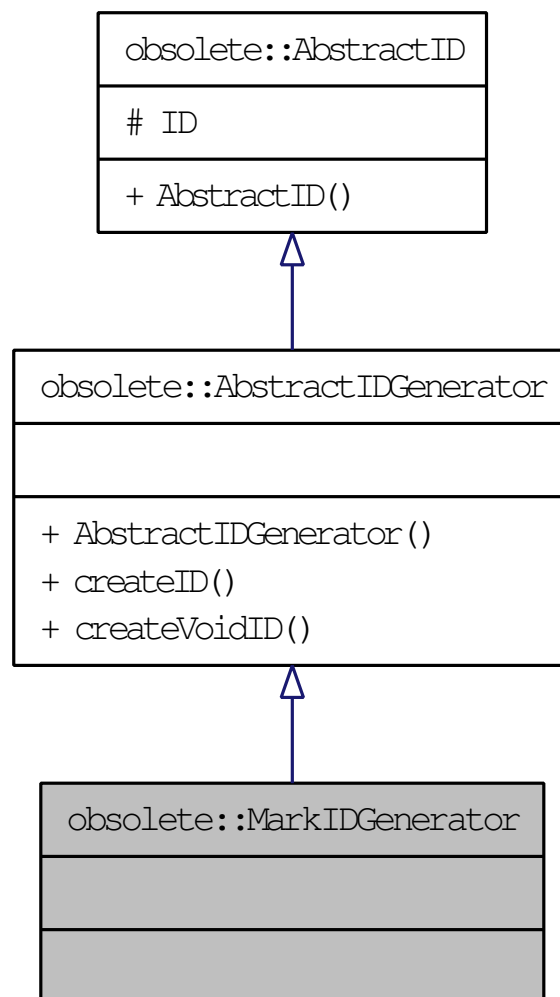
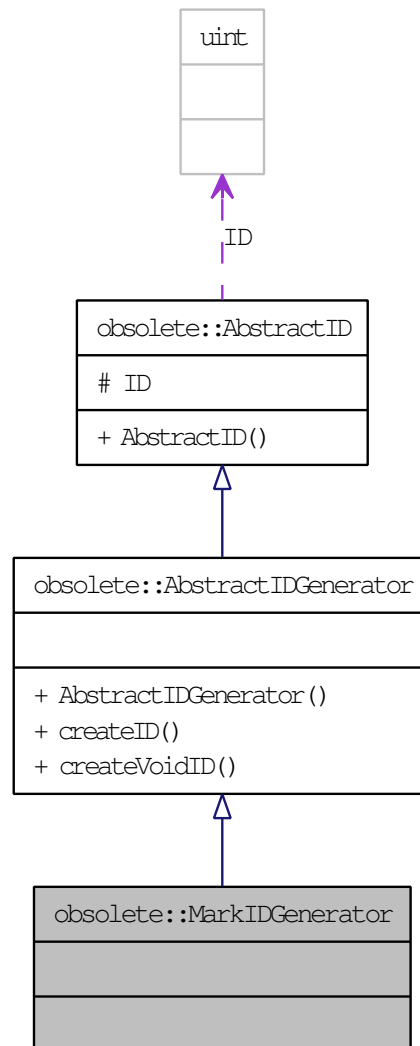


Diagram współpracy dla `obsolete::MarkIDGenerator`:



Metody publiczne

- virtual `ID createID ()`

Tworzy nowy identyfikator.

Statyczne metody publiczne

- static `ID createVoidID ()`

Tworzy niezainicjowany identyfikator.

Atrybuty chronione

- `uint ID`
Reprezentowany [ID](#).

8.14.1 Opis szczegółowy

Klasa generatorów generatorów identyfikatorów skutków wydarzeń.

Definicja w linii 10 pliku `markidgenerator.h`.

8.14.2 Dokumentacja funkcji składowych

8.14.2.1 `virtual ID obsolete::AbstractIDGenerator::createID ()` [`virtual`, `inherited`]

Tworzy nowy identyfikator.

8.14.2.2 `static ID obsolete::AbstractIDGenerator::createVoidID ()` [`static`, `inherited`]

Tworzy niezainicjowany identyfikator.

8.14.3 Dokumentacja atrybutów składowych

8.14.3.1 `uint obsolete::AbstractID::ID` [`protected`, `inherited`]

Reprezentowany [ID](#).

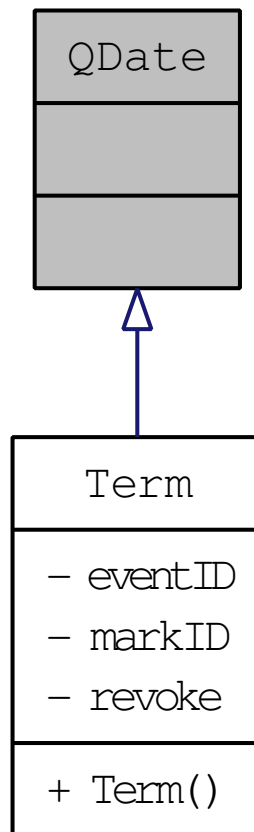
Definicja w linii 34 pliku `abstractid.h`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `include/obsolete/markidgenerator.h`

8.15 Dokumentacja klasy QDate

Diagram dziedziczenia dla QDate



Dokumentacja dla tej klasy została wygenerowana z pliku:

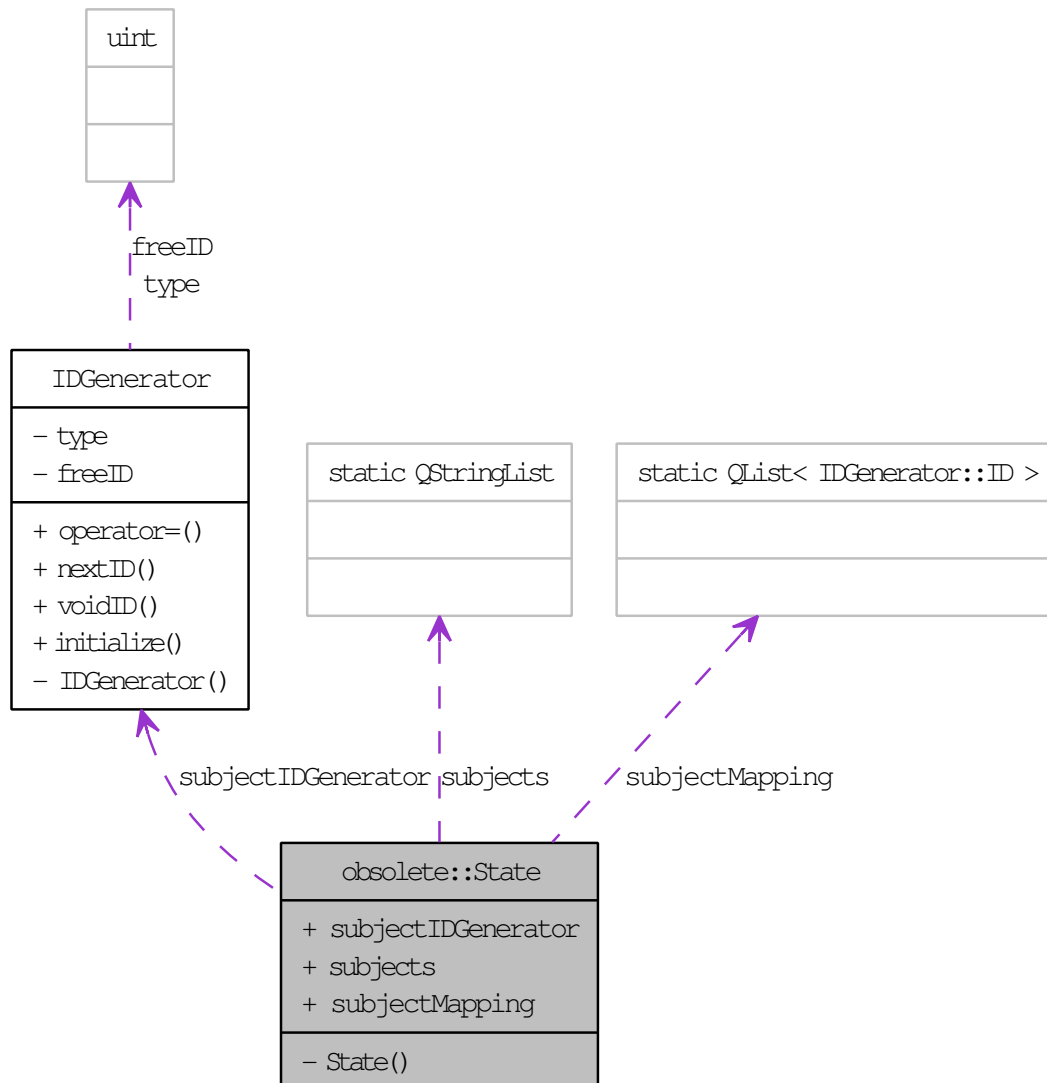
- `include/term.h`

8.16 Dokumentacja klasy obsolete::State

Klasa przechowująca wszelkie informacje o stanie aplikacji, bazy danych i danych.

```
#include <state.h>
```

Diagram współpracy dla obsolete::State:



Statyczne atrybuty publiczne

- static [IDGenerator subjectIDGenerator](#)
Generator identyfikatorów przedmiotów.
- static [QStringList subjects](#)
Lista przedmiotów. subjects[SubjectID] = Przedmiot o zadanym identyfikatorze.
- static [QList< IDGenerator::ID > subjectMapping](#)

Lista odnośników [pozycja w ComboBox]-->[pozycja w subjects].

Metody prywatne

- [State \(\)](#)

Nie wolno tworzyć obiektów tej klasy.

8.16.1 Opis szczegółowy

Klasa przechowująca wszelkie informacje o stanie aplikacji, bazy danych i danych. Jest to kontener na informacje o przedmiotach.

Nie wolno tworzyć obiektów tej klasy, wszystkie pola są statyczne.

Definicja w linii 15 pliku state.h.

8.16.2 Dokumentacja konstruktora i destruktora

8.16.2.1 `obsolete::State::State () [inline, private]`

Nie wolno tworzyć obiektów tej klasy.

Definicja w linii 23 pliku state.h.

8.16.3 Dokumentacja atrybutów składowych

8.16.3.1 `IDGenerator obsolete::State::subjectIDGenerator [static]`

Generator identyfikatorów przedmiotów.

Definicja w linii 18 pliku state.h.

8.16.3.2 `QList<IDGenerator::ID> obsolete::State::subjectMapping [static]`

Lista odnośników [pozycja w ComboBox]-->[pozycja w subjects].

Definicja w linii 20 pliku state.h.

8.16.3.3 `QStringList obsolete::State::subjects [static]`

Lista przedmiotów. `subjects[SubjectID]` = Przedmiot o zadanym identyfikatorze.

Definicja w linii 19 pliku state.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

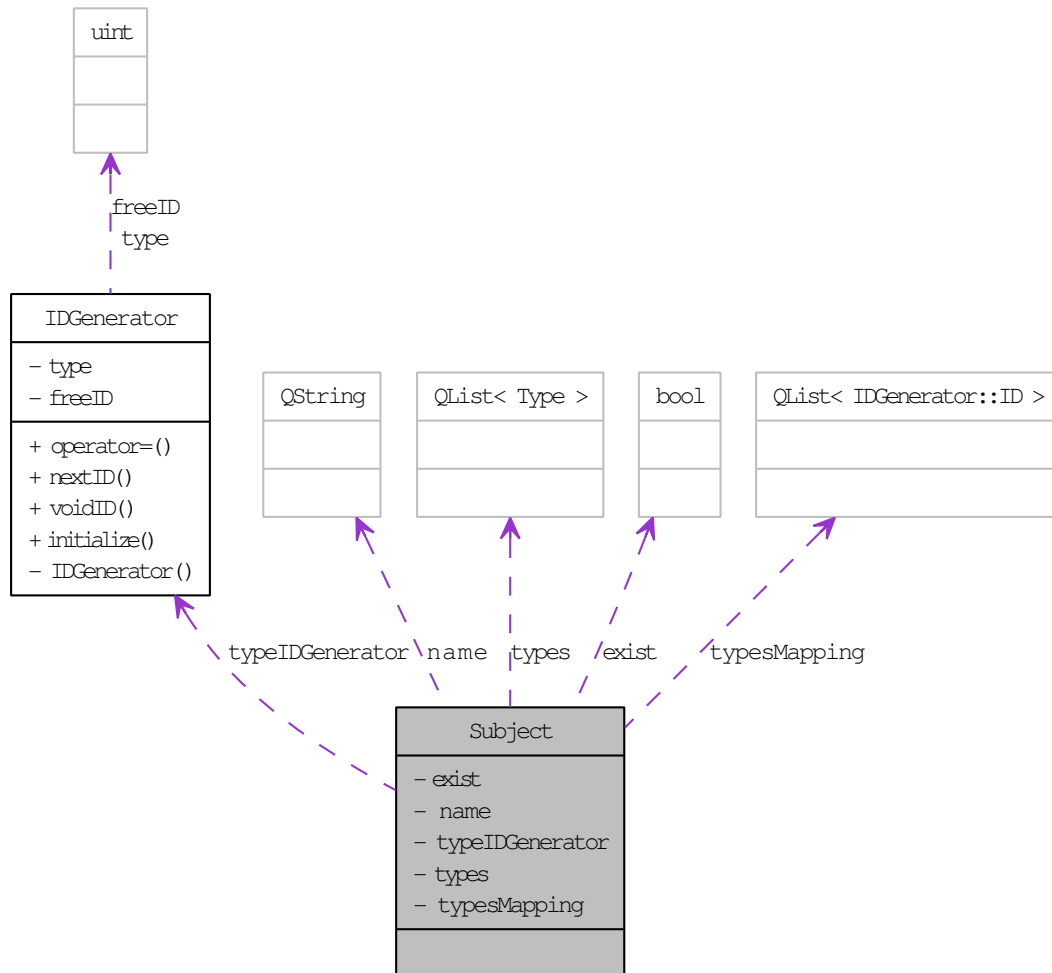
- `include/obsolete/state.h`

8.17 Dokumentacja klasy Subject

Reprezentuje informacje o pojedynczym przedmiocie.

```
#include <subject.h>
```

Diagram współpracy dla Subject:



Atrybuty prywatne

- `bool` **exist**
Czy ten obiekt reprezentuje dane.
- `QString` **name**
Nazwa przedmiotu.
- `IDGenerator` **typeIDGenerator**
Generator identyfikatorów typów wydarzeń.
- `QList< Type >` **types**

Typy wydarzeń. Indeksowane types[TypeID] = wydarzenie TypeID.

- `QList< IDGenerator::ID > typesMapping`

Lista odnośników [pozycja w ComboBox]-->[pozycja w types].

8.17.1 Opis szczegółowy

Reprezentuje informacje o pojedynczym przedmiocie.

Definicja w linii 11 pliku subject.h.

8.17.2 Dokumentacja atrybutów składowych

8.17.2.1 `bool Subject::exist [private]`

Czy ten obiekt reprezentuje dane.

Definicja w linii 12 pliku subject.h.

8.17.2.2 `QString Subject::name [private]`

Nazwa przedmiotu.

Definicja w linii 13 pliku subject.h.

8.17.2.3 `IDGenerator Subject::typeIDGenerator [private]`

Generator identyfikatorów typów wydarzeń.

Definicja w linii 14 pliku subject.h.

8.17.2.4 `QList<Type> Subject::types [private]`

Typy wydarzeń. Indeksowane types[TypeID] = wydarzenie TypeID.

Definicja w linii 15 pliku subject.h.

8.17.2.5 `QList<IDGenerator::ID> Subject::typesMapping [private]`

Lista odnośników [pozycja w ComboBox]-->[pozycja w types].

Definicja w linii 16 pliku subject.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `include/subject.h`

8.18 Dokumentacja klasy `obsolete::SubjectID`

Klasa reprezentująca identyfikator przedmiotu.

```
#include <subjectid.h>
```

Diagram dziedziczenia dla `obsolete::SubjectID`

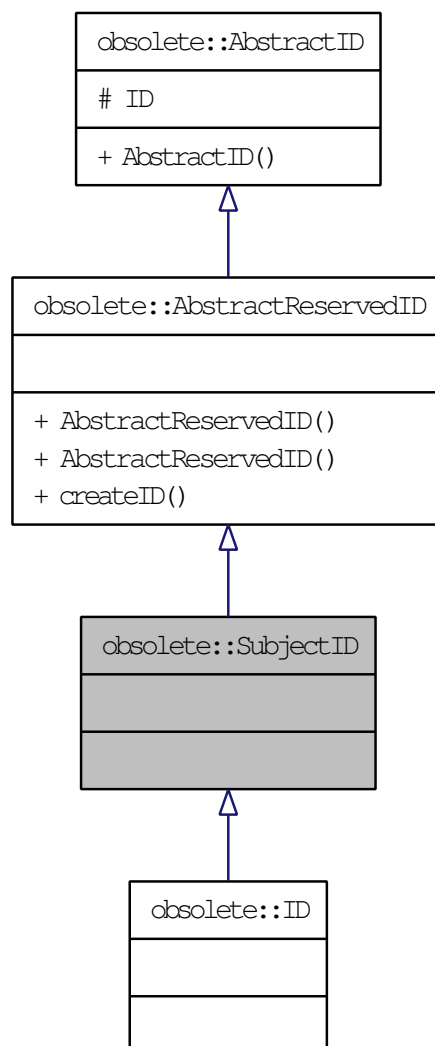
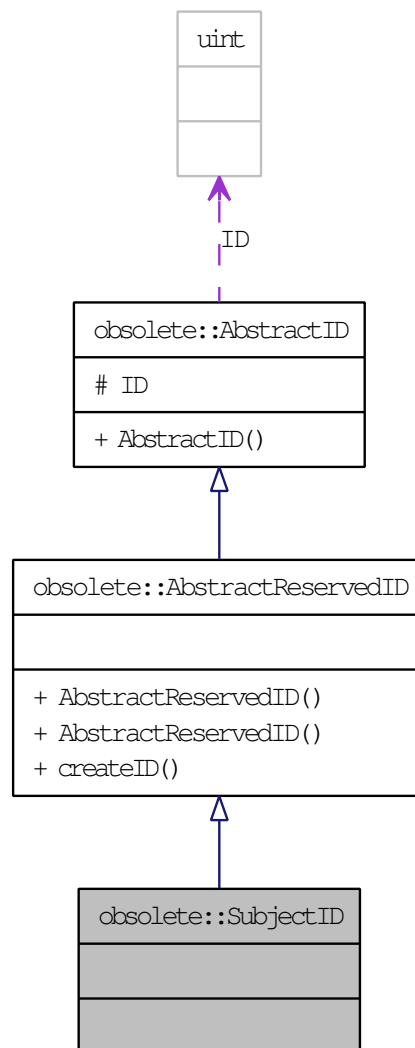


Diagram współpracy dla `obsolete::SubjectID`:



Statyczne metody publiczne

- static `AbstractReservedID createID (uint ID)`

Opakowuje numer porządkowy zadaną liczbą.

Atrybuty chronione

- uint `ID`

Reprezentowany `ID`.

8.18.1 Opis szczegółowy

Klasa reprezentująca identyfikator przedmiotu.

Definicja w linii 10 pliku `subjectid.h`.

8.18.2 Dokumentacja funkcji składowych

8.18.2.1 `static AbstractReservedID obsolete::AbstractReservedID::createID (uint ID)` `[inline, static, inherited]`

Opakowuje numer porządkowy zadaną liczbą.

8.18.3 Dokumentacja atrybutów składowych

8.18.3.1 `uint obsolete::AbstractID::ID` `[protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [include/obsolete/subjectid.h](#)

8.19 Dokumentacja klasy `obsolete::SubjectIDGenerator`

Klasa generatorów generatorów identyfikatorów przedmiotów.

```
#include <subjectidgenerator.h>
```

Diagram dziedziczenia dla `obsolete::SubjectIDGenerator`

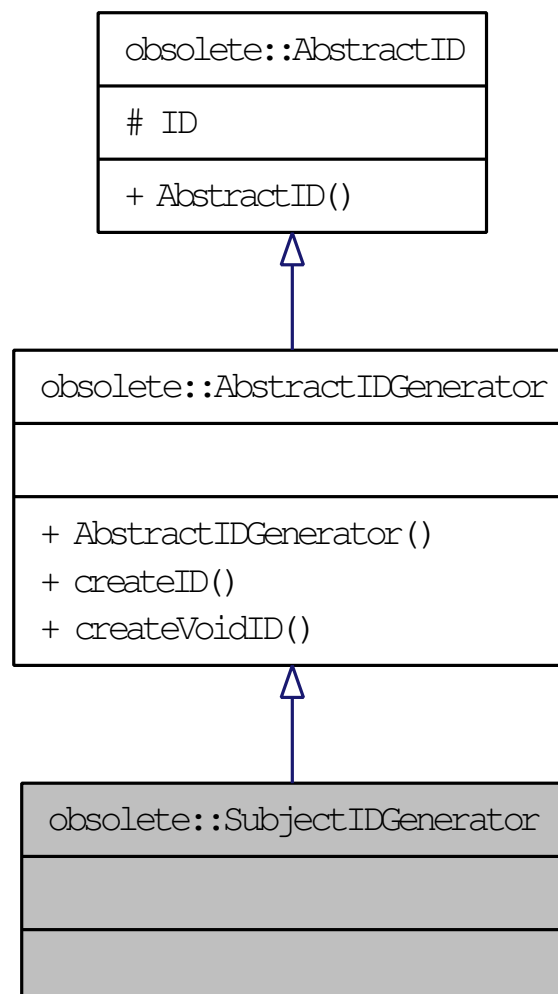
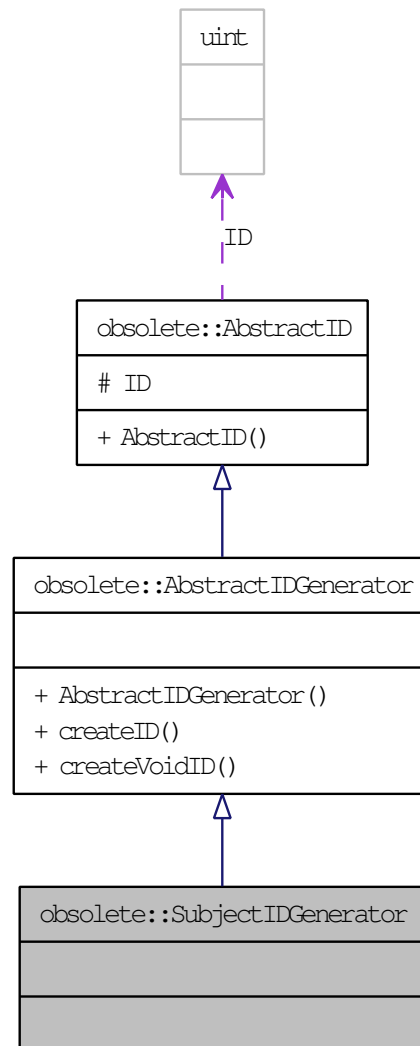


Diagram współpracy dla obsolete::SubjectIDGenerator:



Metody publiczne

- virtual `ID createID ()`

Tworzy nowy identyfikator.

Statyczne metody publiczne

- static `ID createVoidID ()`

Tworzy niezainicjowany identyfikator.

Atrybuty chronione

- uint [ID](#)
Reprezentowany [ID](#).

8.19.1 Opis szczegółowy

Klasa generatorów generatorów identyfikatorów przedmiotów.

Definicja w linii 10 pliku subjectidgenerator.h.

8.19.2 Dokumentacja funkcji składowych

8.19.2.1 virtual ID obsolete::AbstractIDGenerator::createID () [virtual, inherited]

Tworzy nowy identyfikator.

8.19.2.2 static ID obsolete::AbstractIDGenerator::createVoidID () [static, inherited]

Tworzy niezainicjowany identyfikator.

8.19.3 Dokumentacja atrybutów składowych

8.19.3.1 uint obsolete::AbstractID::ID [protected, inherited]

Reprezentowany [ID](#).

Definicja w linii 34 pliku abstractid.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

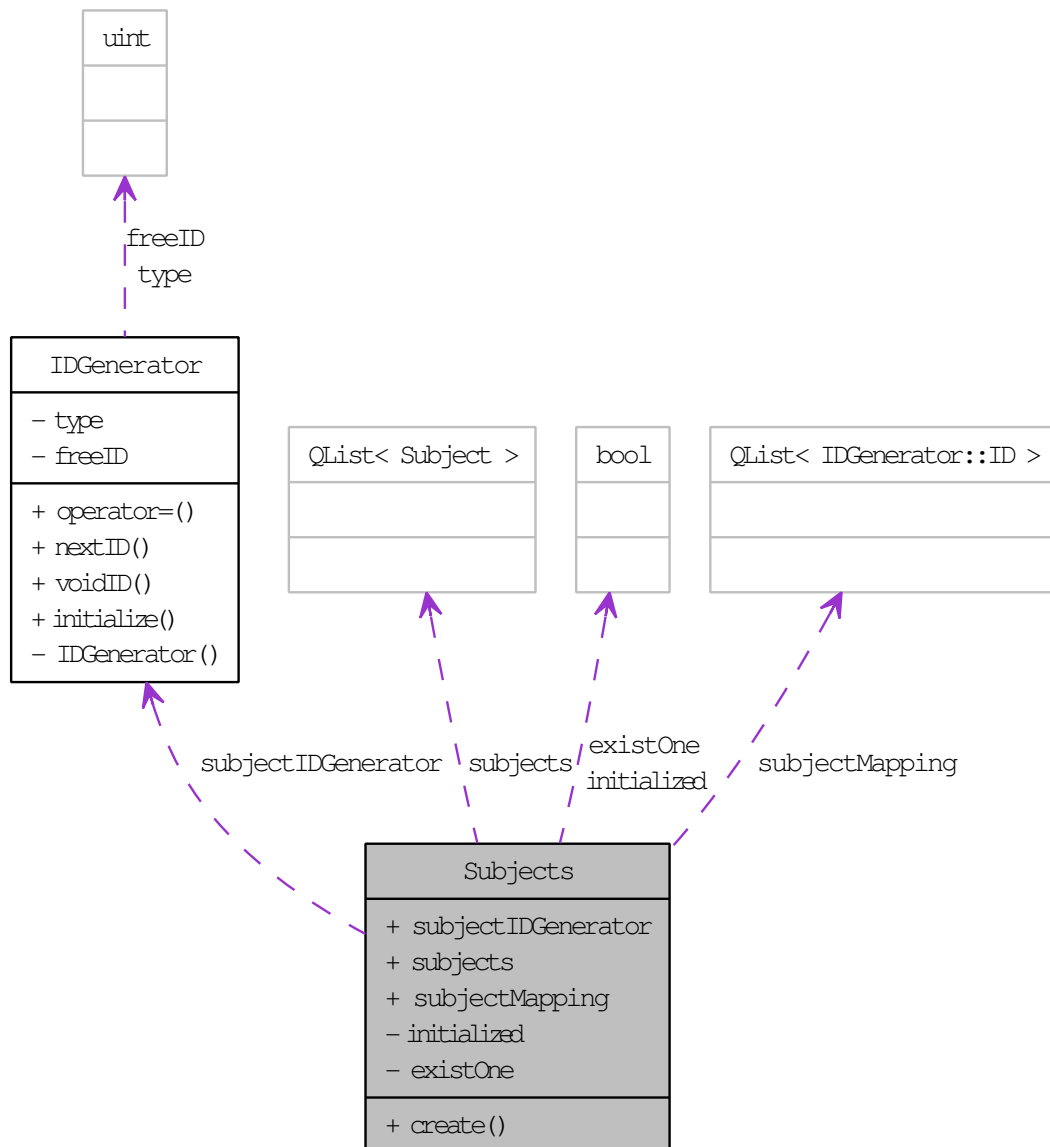
- include/obsolete/[subjectidgenerator.h](#)

8.20 Dokumentacja klasy Subjects

Obiekt klasy reprezentuje wszystkie przedmioty.

```
#include <subjects.h>
```

Diagram współpracy dla Subjects:



Statyczne metody publiczne

- static [Subjects create \(\)](#) private throw (char*)

Tworzy instancje klasy.

Atrybuty publiczne

- [IDGenerator subjectIDGenerator](#)
Generator identyfikatorów przedmiotów.
- [QList< Subject > subjects](#)
Lista przedmiotów.
- [QList< IDGenerator::ID > subjectMapping](#)
Lista odnośników [pozycja w ComboBox]-->[pozycja w subjects].

Atrybuty prywatne

- [bool initialized](#)
Czy obiekt tej klasy został zainicjalizowany.

Statyczne atrybuty prywatne

- [static bool existOne = false](#)
Czy istnieje jakiś obiekt tej klasy.

8.20.1 Opis szczegółowy

Obiekt klasy reprezentuje wszystkie przedmioty.

Definicja w linii 10 pliku subjects.h.

8.20.2 Dokumentacja funkcji składowych

8.20.2.1 Subjects Subjects::create () throw (char*) [inline, static]

Tworzy instancje klasy.

Wyjątki

*char** Jeżeli już wywołano tę metodę

Zwraca

Nowy obiekt tej klasy Oznacza nowy obiekt jako niezainicjalizowany.

Definicja w linii 23 pliku subjects.h.

8.20.3 Dokumentacja atrybutów składowych

8.20.3.1 `bool Subjects::existOne = false` `[static, private]`

Czy istnieje jakiś obiekt tej klasy.

Definicja w linii 16 pliku subjects.h.

8.20.3.2 `bool Subjects::initialized` `[private]`

Czy obiekt tej klasy został zainicjalizowany.

Definicja w linii 15 pliku subjects.h.

8.20.3.3 `IDGenerator Subjects::subjectIDGenerator`

Generator identyfikatorów przedmiotów.

Definicja w linii 10 pliku subjects.h.

8.20.3.4 `QList<IDGenerator::ID> Subjects::subjectMapping`

Lista odnośników [pozycja w ComboBox]-->[pozycja w subjects].

Definicja w linii 12 pliku subjects.h.

8.20.3.5 `QList<Subject> Subjects::subjects`

Lista przedmiotów.

Definicja w linii 11 pliku subjects.h.

Dokumentacja dla tej klasy została wygenerowana z plików:

- [include/subjects.h](#)
- [src/events.cpp](#)
- [src/subjects.cpp](#)

8.21 Dokumentacja klasy Term

Reprezentacja egzekucji wydarzenia.

```
#include <term.h>
```

Diagram dziedziczenia dla Term

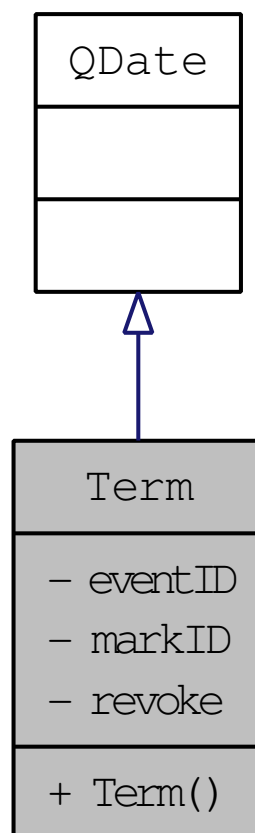
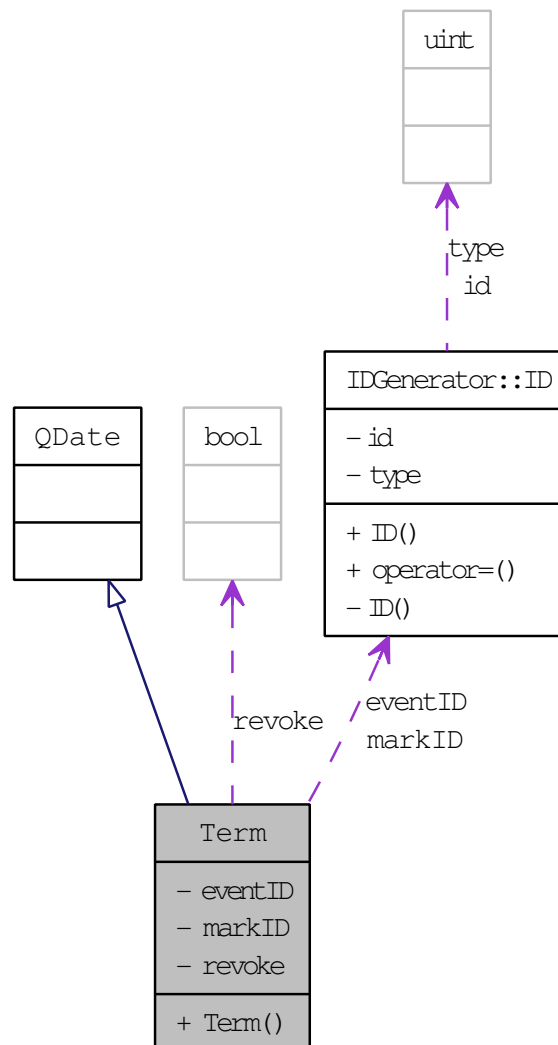


Diagram współpracy dla Term:



Metody publiczne

- **Term** ([IDGenerator::ID](#) event)

Inicjalizuje obiekt reprezentujący termin danego wydarzenia.

Atrybuty prywatne

- [IDGenerator::ID](#) **eventID**

Identyfikator wydarzenia.

- [IDGenerator::ID](#) **markID**

Identyfikator oceny - skutku egzekucji wydarzenia.

- bool [revoke](#)

Czy jest odwołany.

Przyjaciele

- QDataStream & [operator<<](#) (QDataStream &stream, const [Term](#) &term)

Zapisuje dany termin do danego strumienia.

- QDataStream & [operator>>](#) (QDataStream &stream, [Term](#) &term)

Odczytuje dany termin z danego strumienia.

8.21.1 Opis szczegółowy

Reprezentacja egzekucji wydarzenia. Jest również odnośnikiem do danego wydarzenia.

Definicja w linii 9 pliku term.h.

8.21.2 Dokumentacja konstruktora i destruktor

8.21.2.1 [Term::Term \(IDGenerator::ID event\)](#) [[inline](#)]

Inicjalizuje obiekt reprezentujący termin danego wydarzenia.

Parametry

event Identyfikator wydarzenia związanego z tym terminem

Definicja w linii 29 pliku term.h.

```
29 : eventID(event) {}
```

8.21.3 Dokumentacja przyjaciół i funkcji związanych

8.21.3.1 [QDataStream& operator<< \(QDataStream &stream, const Term &term\)](#) [[friend](#)]

Zapisuje dany termin do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany termin.

term Termin który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 3 pliku term.cpp.

```
4 {  
5     const QDate &date=term;  
6     return stream<<date<<term.eventID<<term.markID;  
7 }
```

8.21.3.2 QDataStream& operator>> (QDataStream & stream, Term & term) [friend]

Odczytuje dany termin z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany termin.

term Termin który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 9 pliku term.cpp.

```
10 {  
11     QDate &date=term;  
12     return stream>>date>>term.eventID>>term.markID;  
13 }
```

8.21.4 Dokumentacja atrybutów składowych

8.21.4.1 IDGenerator::ID Term::eventID [private]

Identyfikator wydarzenia.

Definicja w linii 31 pliku term.h.

Odwołania w operator<<() i operator>>().

8.21.4.2 IDGenerator::ID Term::markID [private]

Identyfikator oceny - skutku egzekucji wydarzenia.

Definicja w linii 32 pliku term.h.

Odwołania w operator<<() i operator>>().

8.21.4.3 bool Term::revoke [private]

Czy jest odwołany.

Definicja w linii 33 pliku term.h.

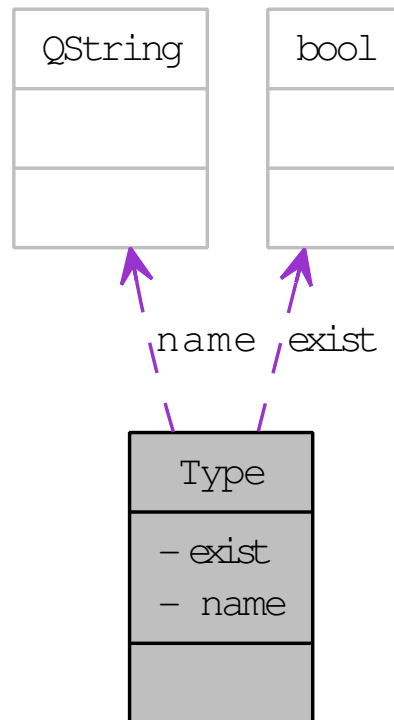
Dokumentacja dla tej klasy została wygenerowana z pliku:

- [include/term.h](#)

8.22 Dokumentacja klasy Type

```
#include <type.h>
```

Diagram współpracy dla Type:



Atrybuty prywatne

- `bool exist`
Czy ten obiekt reprezentuje dane.
- `QString name`
Nazwa typu.

8.22.1 Opis szczegółowy

Definicja w linii 6 pliku `type.h`.

8.22.2 Dokumentacja atrybutów składowych

8.22.2.1 `bool Type::exist [private]`

Czy ten obiekt reprezentuje dane.

Definicja w linii 10 pliku `type.h`.

8.22.2.2 QString Type::name [private]

Nazwa typu.

Definicja w linii 11 pliku type.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [include/type.h](#)

8.23 Dokumentacja klasy `obsolete::TypeID`

Klasa reprezentująca identyfikator typu wydarzenia.

```
#include <typeid.h>
```

Diagram dziedziczenia dla `obsolete::TypeID`

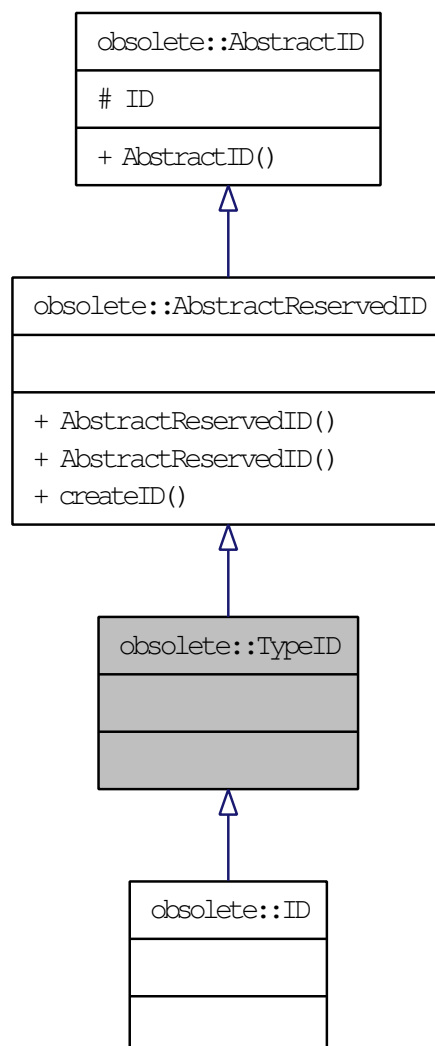
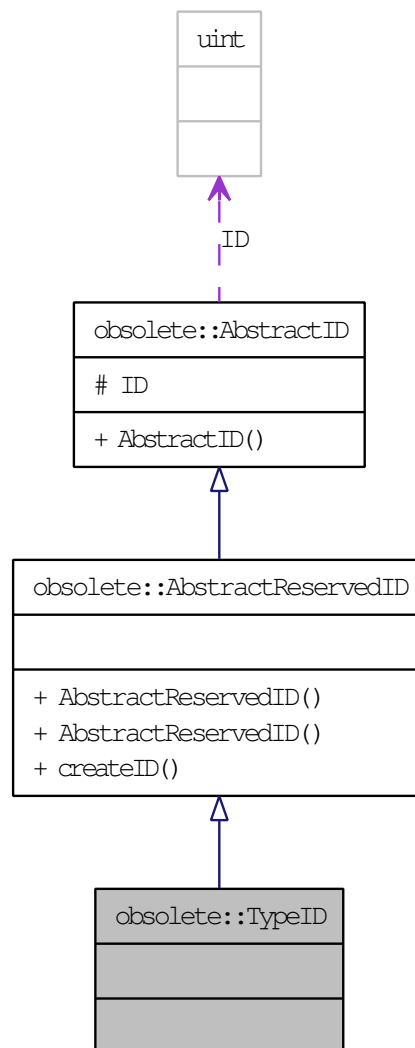


Diagram współpracy dla `obsolete::TypeID`:



Statyczne metody publiczne

- static `AbstractReservedID createID (uint ID)`

Opakowuje numer porządkowy zadaną liczbą.

Atrybuty chronione

- uint `ID`

Reprezentowany `ID`.

8.23.1 Opis szczegółowy

Klasa reprezentująca identyfikator typu wydarzenia.

Definicja w linii 10 pliku typeid.h.

8.23.2 Dokumentacja funkcji składowych

8.23.2.1 `static AbstractReservedID obsolete::AbstractReservedID::createID (uint ID)` `[inline, static, inherited]`

Opakowuje numer porządkowy zadaną liczbą.

8.23.3 Dokumentacja atrybutów składowych

8.23.3.1 `uint obsolete::AbstractID::ID` `[protected, inherited]`

Reprezentowany [ID](#).

Definicja w linii 34 pliku abstractid.h.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- [include/obsolete/typeid.h](#)

8.24 Dokumentacja klasy `obsolete::TypeIDGenerator`

Generator generatorów obiektów [TypeID](#).

```
#include <typeidgenerator.h>
```

Diagram dziedziczenia dla `obsolete::TypeIDGenerator`

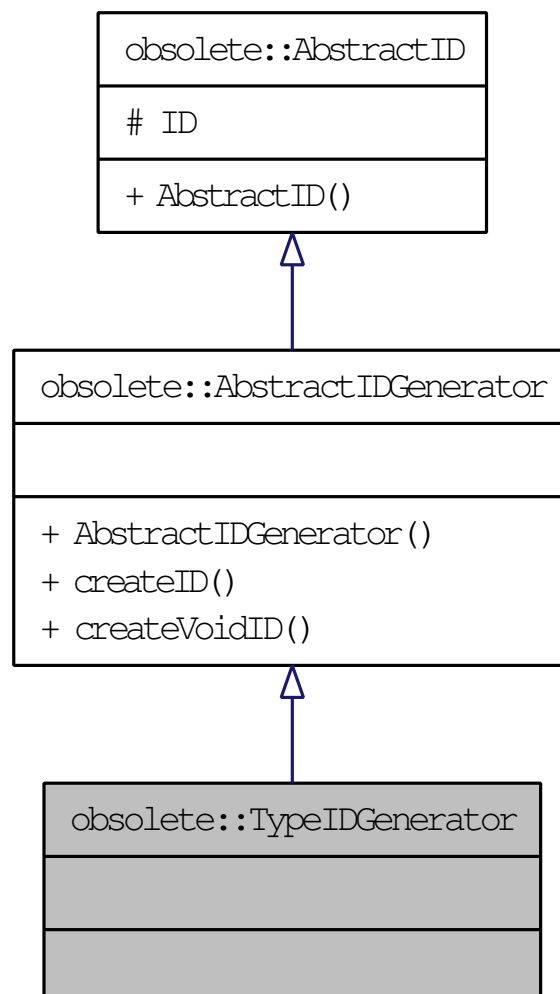
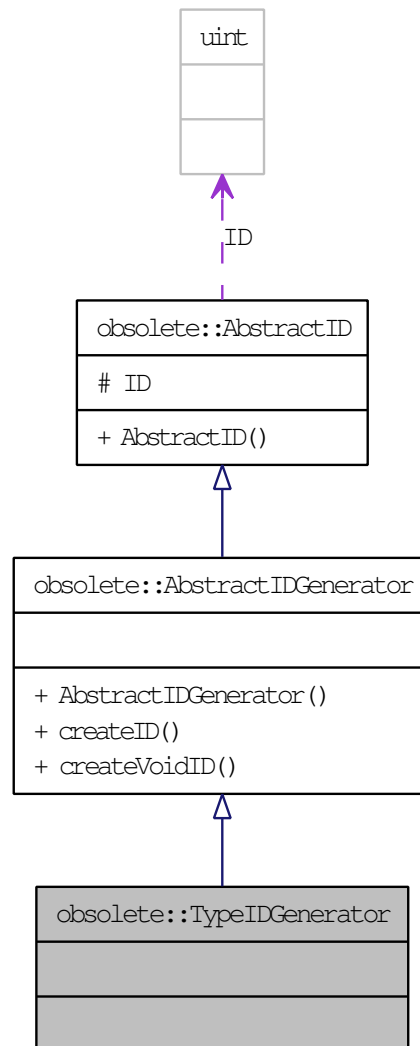


Diagram współpracy dla `obsolete::TypeIDGenerator`:



Metody publiczne

- virtual `ID createID ()`

Tworzy nowy identyfikator.

Statyczne metody publiczne

- static `ID createVoidID ()`

Tworzy niezainicjowany identyfikator.

Atrybuty chronione

- `uint ID`
Reprezentowany [ID](#).

8.24.1 Opis szczegółowy

Generator generatorów obiektów [TypeID](#).

Definicja w linii 10 pliku `typeidgenerator.h`.

8.24.2 Dokumentacja funkcji składowych

8.24.2.1 `virtual ID obsolete::AbstractIDGenerator::createID ()` [`virtual`, `inherited`]

Tworzy nowy identyfikator.

8.24.2.2 `static ID obsolete::AbstractIDGenerator::createVoidID ()` [`static`, `inherited`]

Tworzy niezainicjowany identyfikator.

8.24.3 Dokumentacja atrybutów składowych

8.24.3.1 `uint obsolete::AbstractID::ID` [`protected`, `inherited`]

Reprezentowany [ID](#).

Definicja w linii 34 pliku `abstractid.h`.

Dokumentacja dla tej klasy została wygenerowana z pliku:

- `include/obsolete/typeidgenerator.h`

Rozdział 9

Dokumentacja plików

9.1 Dokumentacja pliku build/CMakeFiles/CompilerIdC/CMakeC-CompilerId.c

Definicje

- `#define COMPILER_ID ""`
- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`

Funkcje

- `int main (int argc, char *argv[])`

Zmienne

- `char * info_compiler = "INFO" ":" "compiler[" "" "]"`
- `char * info_platform = "INFO" ":" "platform[" "" "]"`
- `char * info_arch = "INFO" ":" "arch[" "" "]"`

9.1.1 Dokumentacja definicji

9.1.1.1 `#define ARCHITECTURE_ID ""`

Definicja w linii 191 pliku CMakeCCompilerId.c.

9.1.1.2 `#define COMPILER_ID ""`

Definicja w linii 77 pliku CMakeCCompilerId.c.

9.1.1.3 `#define PLATFORM_ID ""`

Definicja w linii 167 pliku CMakeCCompilerId.c.

9.1.2 Dokumentacja funkcji

9.1.2.1 `int main (int argc, char * argv[])`

Definicja w linii 208 pliku CMakeCCompilerId.c.

Odwołuje się do `info_arch`, `info_compiler` i `info_platform`.

```
209 {
210     int require = 0;
211     require += info_compiler[argc];
212     require += info_platform[argc];
213     require += info_arch[argc];
214     (void)argv;
215     return require;
216 }
```

9.1.3 Dokumentacja zmiennych

9.1.3.1 `char* info_arch = "INFO" ":" "arch[" "" "]"`

Definicja w linii 199 pliku CMakeCCompilerId.c.

Odwołania w `main()`.

9.1.3.2 `char* info_compiler = "INFO" ":" "compiler[" "" "]"`

Definicja w linii 85 pliku CMakeCCompilerId.c.

Odwołania w `main()`.

9.1.3.3 `char* info_platform = "INFO" ":" "platform[" "" "]"`

Definicja w linii 198 pliku CMakeCCompilerId.c.

Odwołania w `main()`.

9.2 Dokumentacja pliku build/CMakeFiles/CompilerIdCXX/CMakeCXXCompilerId.cpp

Definicje

- `#define COMPILER_ID ""`
- `#define PLATFORM_ID ""`
- `#define ARCHITECTURE_ID ""`

Funkcje

- `int main (int argc, char *argv[])`

Zmienne

- `char * info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`
- `char * info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`
- `char * info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

9.2.1 Dokumentacja definicji

9.2.1.1 `#define ARCHITECTURE_ID ""`

Definicja w linii 182 pliku CMakeCXXCompilerId.cpp.

9.2.1.2 `#define COMPILER_ID ""`

Definicja w linii 68 pliku CMakeCXXCompilerId.cpp.

9.2.1.3 `#define PLATFORM_ID ""`

Definicja w linii 158 pliku CMakeCXXCompilerId.cpp.

9.2.2 Dokumentacja funkcji

9.2.2.1 `int main (int argc, char * argv[])`

Definicja w linii 196 pliku CMakeCXXCompilerId.cpp.

Odwołuje się do `info_compiler` i `info_platform`.

```
197 {
198   int require = 0;
199   require += info_compiler[argc];
200   require += info_platform[argc];
201   (void)argv;
202   return require;
203 }
```

9.2.3 Dokumentacja zmiennych

9.2.3.1 `char* info_arch = "INFO" ":" "arch[" ARCHITECTURE_ID "]"`

Definicja w linii 190 pliku CMakeCXXCompilerId.cpp.

9.2.3.2 `char* info_compiler = "INFO" ":" "compiler[" COMPILER_ID "]"`

Definicja w linii 76 pliku CMakeCXXCompilerId.cpp.

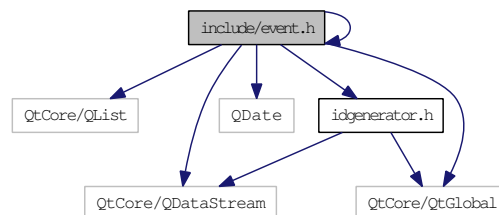
9.2.3.3 `char* info_platform = "INFO" ":" "platform[" PLATFORM_ID "]"`

Definicja w linii 189 pliku CMakeCXXCompilerId.cpp.

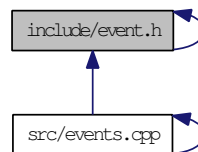
9.3 Dokumentacja pliku include/event.h

```
#include <QtCore/QList>
#include <QtCore/QDataStream>
#include "term.h"
#include <QDate>
#include <QtCore/QtGlobal>
#include "idgenerator.h"
```

Wykres zależności załączania dla event.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class [Event](#)

Reprezentuje zdarzenie w postaci zdefiniowanej przez użytkownika.

Funkcje

- `QDataStream & operator<< (QDataStream &stream, const Event &event)`

Zapisuje informacje o danym wydarzeniu do danego strumienia.

- `QDataStream & operator>> (QDataStream &stream, Event &event)`

Odczytuje informacje o danym wydarzeniu z danego strumienia.

9.3.1 Dokumentacja funkcji

9.3.1.1 QDataStream& operator<< (QDataStream & stream, const Event & event)

Zapisuje informacje o danym wydarzeniu do danego strumienia.

Parametry

stream Strumień do którego będą zapisywane dane.

event Wydarzenie które będzie zapisywane.

Zwraca

Ten sam strumień.

Definicja w linii 3 pliku event.cpp.

```
4 {  
5     return stream<<event.subjectID<<event.typeID<<event.terms;  
6 }
```

9.3.1.2 QDataStream& operator>> (QDataStream & *stream*, Event & *event*)

Odczytuje informacje o danym wydarzeniu z danego strumienia.

Parametry

stream Strumień z którego będą odczytywane dane.

event Wydarzenie które zostanie zainicjalizowane wczytanymi danymi.

Zwraca

Ten sam strumień.

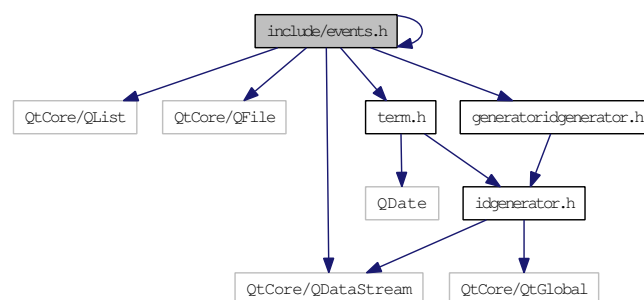
Definicja w linii 8 pliku event.cpp.

```
9 {  
10     return stream>>event.subjectID<<event.typeID<<event.terms;  
11 }
```

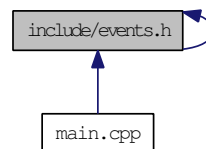
9.4 Dokumentacja pliku include/events.h

```
#include <QtCore/QList>
#include <QtCore/QFile>
#include "event.h"
#include <QtCore/QDataStream>
#include "term.h"
#include "generatoridgenerator.h"
```

Wykres zależności załączania dla events.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class [Events](#)

Klasa przechowuje wszystkie wydarzenia i zarządza nimi.

Funkcje

- `QDataStream & operator<< (QDataStream &stream, const Events &events)`

Zapisuje informacje o wydarzeniach do danego strumienia.

- `QDataStream & operator>> (QDataStream &stream, Events &events)`

Odczytuje informacje o wydarzeniach z danego strumienia.

9.4.1 Dokumentacja funkcji

9.4.1.1 QDataStream& operator<< (QDataStream & *stream*, const Events & *events*)

Zapisuje informacje o wydarzeniach do danego strumienia.

Parametry

stream Strumień do którego będą zapisywane dane.

events Wydarzenia które będzie zapisywane.

Zwraca

Ten sam strumień.

Definicja w linii 5 pliku events.cpp.

Odwołuje się do Events::eventIDGenerator i Events::events.

```
6 {  
7     return stream<<events.eventIDGenerator<<events.events;  
8 }
```

9.4.1.2 QDataStream& operator>> (QDataStream & *stream*, Events & *events*)

Odczytuje informacje o wydarzeniach z danego strumienia.

Parametry

stream Strumień z którego będą odczytywane dane.

events Obiekt klasy [Events](#) który zostanie zainicjalizowany wczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 10 pliku events.cpp.

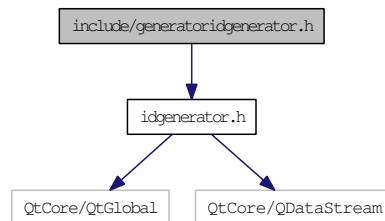
Odwołuje się do Events::eventIDGenerator i Events::events.

```
11 {  
12     return stream>>events.eventIDGenerator>>events.events;  
13 }
```

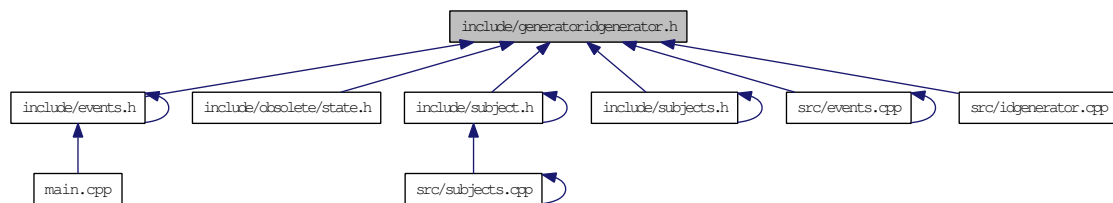
9.5 Dokumentacja pliku include/generatoridgenerator.h

```
#include "idgenerator.h"
```

Wykres zależności załączania dla generatoridgenerator.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

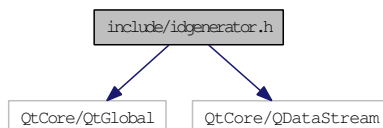
- class [GeneratorIDGenerator](#)

Ta klasa odpowiada za generowanie generatorów identyfikatorów.

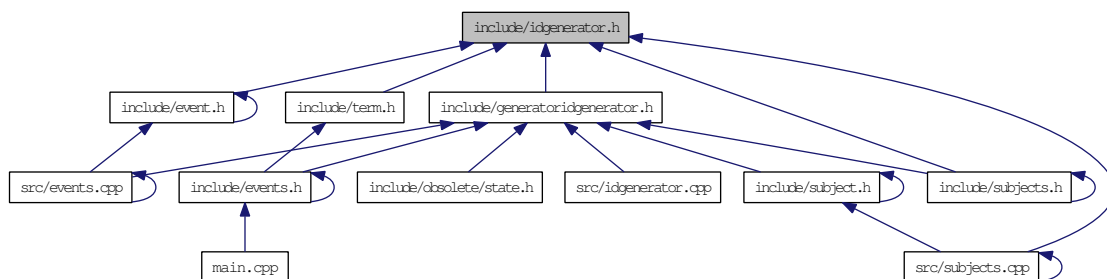
9.6 Dokumentacja pliku include/idgenerator.h

```
#include <QtCore/QtGlobal>
#include <QtCore/QDataStream>
```

Wykres zależności załączania dla idgenerator.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class **IDGenerator**
Generator identyfikatorów określonego typu.
- class **IDGenerator::ID**
Obiekt reprezentuje dany identyfikator określonego typu.

Funkcje

- `QDataStream & operator<< (QDataStream &stream, const IDGenerator::ID &id)`
Zapisuje dany identyfikator do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, IDGenerator::ID &id)`
Odczytuje dany identyfikator z danego strumienia.
- `QDataStream & operator<< (QDataStream &stream, const IDGenerator &generator)`
Zapisuje dany generator identyfikatorów do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, IDGenerator &generator)`
Odczytuje dany generator identyfikatorów z danego strumienia.

9.6.1 Dokumentacja funkcji

9.6.1.1 QDataStream& operator<< (QDataStream & *stream*, const IDGenerator & *generator*)

Zapisuje dany generator identyfikatorów do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany generator identyfikatorów.

generator Generator identyfikatorów który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 30 pliku idgenerator.cpp.

Odwołuje się do IDGenerator::freeID i IDGenerator::type.

```
31 {  
32     return stream<<generator.type<<generator.freeID;  
33 }
```

9.6.1.2 QDataStream& operator<< (QDataStream & *stream*, const IDGenerator::ID & *id*)

Zapisuje dany identyfikator do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany identyfikator.

id Identyfikator który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 20 pliku idgenerator.cpp.

```
21 {  
22     return stream<<id.type<<id.id;  
23 }
```

9.6.1.3 QDataStream& operator>> (QDataStream & *stream*, IDGenerator & *generator*)

Odczytuje dany generator identyfikatorów z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany generator identyfikatorów.

generator Generator identyfikatorów który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 35 pliku idgenerator.cpp.

Odwołuje się do IDGenerator::freeID i IDGenerator::type.

```
36 {  
37     return stream>>generator.type>>generator.freeID;  
38 }
```

9.6.1.4 QDataStream& operator>> (QDataStream & *stream*, IDGenerator::ID & *id*)

Odczytuje dany identyfikator z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany identyfikator.

id Identyfikator który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

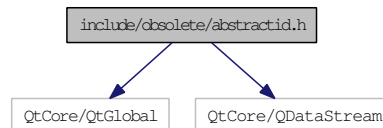
Definicja w linii 25 pliku idgenerator.cpp.

```
26 {  
27     return stream>>id.type>>id.id;  
28 }
```

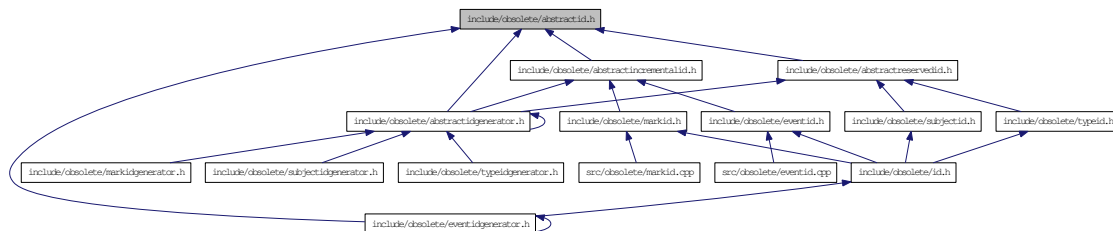

9.7 Dokumentacja pliku include/obsolete/abstractid.h

```
#include <QtCore/QtGlobal>
#include <QtCore/QDataStream>
```

Wykres zależności załączania dla abstractid.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::AbstractID`
Klasa bazowa dla wszystkich typów identyfikatorowych.

Przestrzeń nazw

- namespace `obsolete`

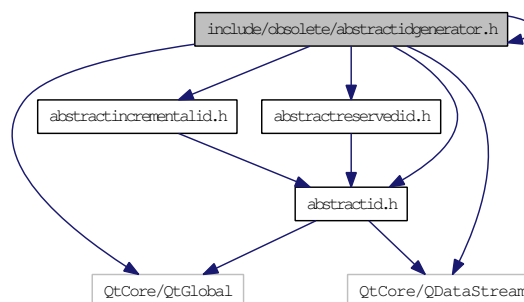
Funkcje

- `QDataStream & obsolete::operator<< (QDataStream &stream, const AbstractID &ID)`
Zapisuje dany identyfikator do danego strumienia.
- `QDataStream & obsolete::operator>> (QDataStream &stream, AbstractID &ID)`
Odczytuje dany identyfikator z danego strumienia.

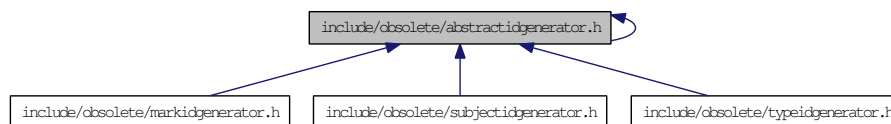
9.8 Dokumentacja pliku include/obsolete/abstractidgenerator.h

```
#include "abstractid.h"
#include <QtCore/QtGlobal>
#include <QtCore/QDataStream>
#include "abstractid.h"
#include "abstractincrementalid.h"
#include "abstractreservedid.h"
```

Wykres zależności załączania dla abstractidgenerator.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::AbstractIDGenerator`
Klasa bazowa obiektów-generatorów identyfikatorów.

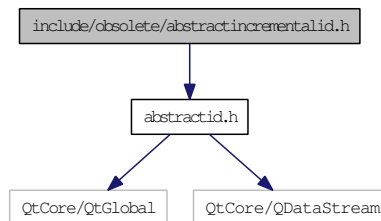
Przestrzeń nazw

- namespace `obsolete`

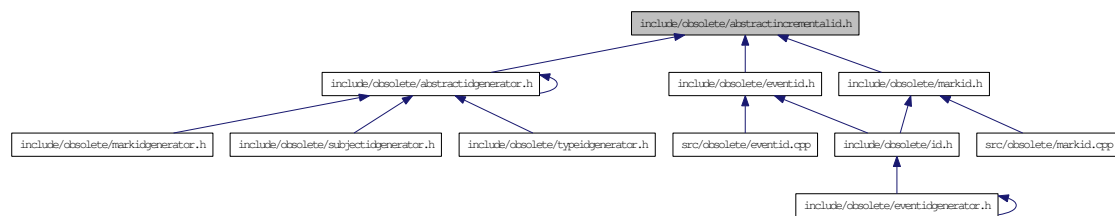
9.9 Dokumentacja pliku include/obsolete/abstractincrementalid.h

```
#include "abstractid.h"
```

Wykres zależności załączania dla abstractincrementalid.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::AbstractIncrementalID`

Podstawa dla wszystkich klas identyfikatorów automatycznych.

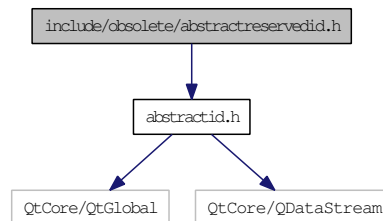
Przestrzeń nazw

- namespace `obsolete`

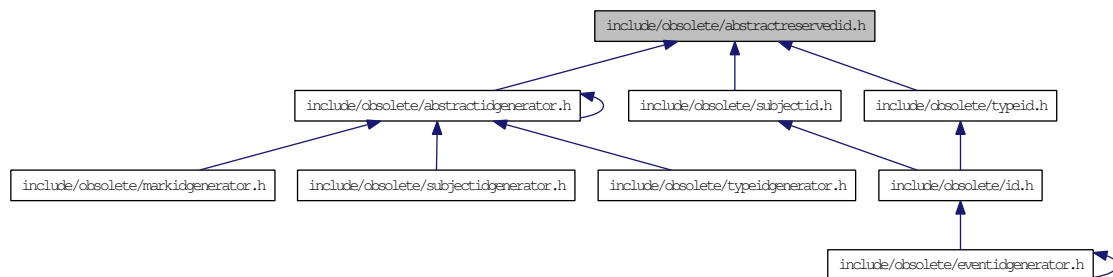
9.10 Dokumentacja pliku include/obsolete/abstractreservedid.h

```
#include "abstractid.h"
```

Wykres zależności załączania dla abstractreservedid.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::AbstractReservedID`
Klasa reprezentująca identyfikatory rezerwowane z góry.

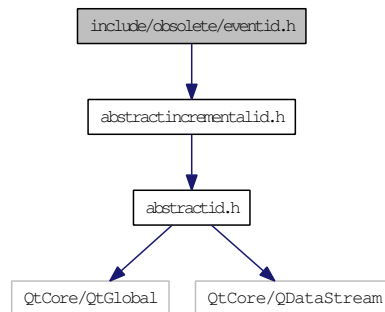
Przestrzenie nazw

- namespace `obsolete`

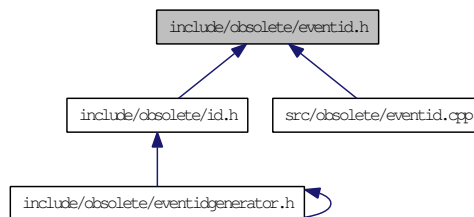
9.11 Dokumentacja pliku include/obsolete/eventid.h

```
#include "abstractincrementalid.h"
```

Wykres zależności załączania dla eventid.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::EventID`

Klasa reprezentująca identyfikator zdarzenia.

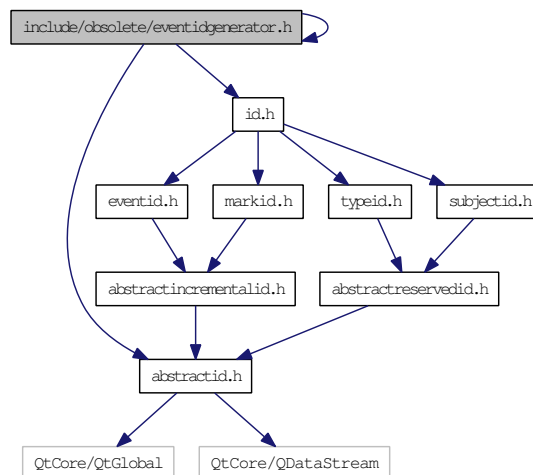
Przestrzeń nazw

- namespace `obsolete`

9.12 Dokumentacja pliku include/obsolete/eventidgenerator.h

```
#include "abstractidgenerator.h"
#include "abstractid.h"
#include "id.h"
```

Wykres zależności załączania dla eventidgenerator.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::EventIDGenerator`

Klasa generatorów generatorów identyfikatorów wydarzeń.

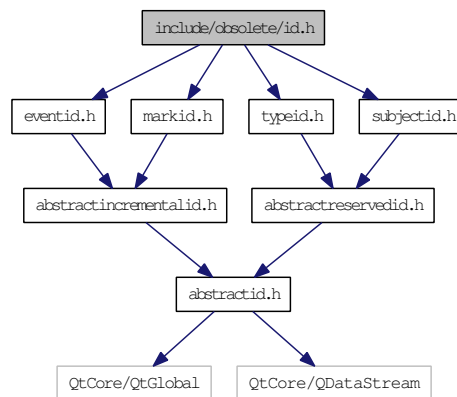
Przestrzenie nazw

- namespace `obsolete`

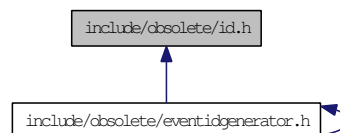
9.13 Dokumentacja pliku include/obsolete/id.h

```
#include "eventid.h"  
#include "markid.h"  
#include "typeid.h"  
#include "subjectid.h"
```

Wykres zależności załączania dla id.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::ID`
*Korzeń nowej idei uniwersalnego **ID**.*

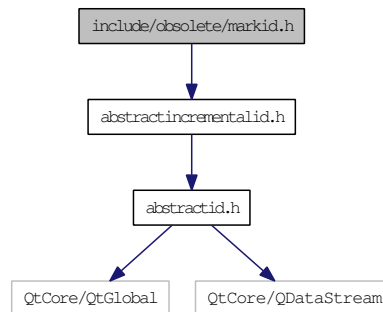
Przestrzenie nazw

- namespace `obsolete`

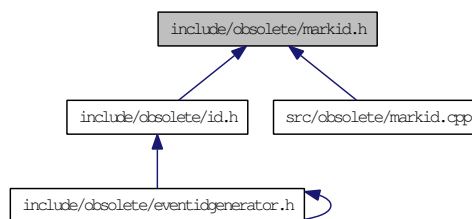
9.14 Dokumentacja pliku include/obsolete/markid.h

```
#include "abstractincrementalid.h"
```

Wykres zależności załączania dla markid.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::MarkID`

Klasa reprezentująca identyfikator oceny.

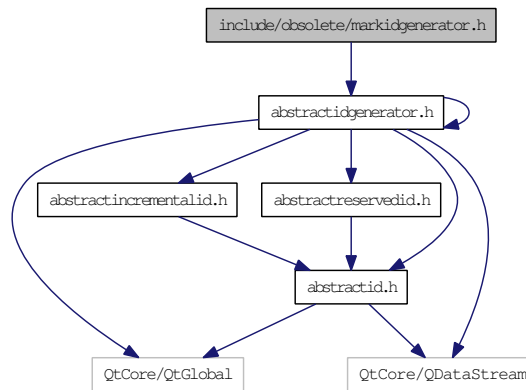
Przestrzeń nazw

- namespace `obsolete`

9.15 Dokumentacja pliku include/obsolete/markidgenerator.h

```
#include "abstractidgenerator.h"
```

Wykres zależności załączania dla markidgenerator.h:



Komponenty

- class `obsolete::MarkIDGenerator`
Klasa generatorów generatorów identyfikatorów skutków wydarzeń.

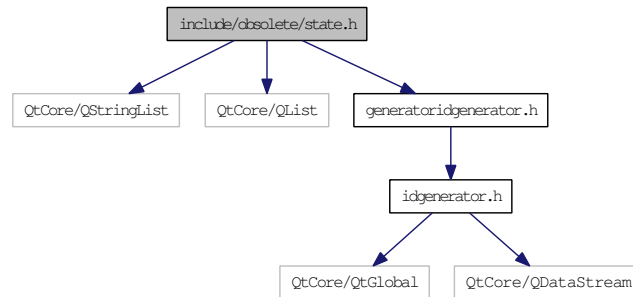
Przestrzeń nazw

- namespace `obsolete`

9.16 Dokumentacja pliku include/obsolete/state.h

```
#include <QtCore/QStringList>
#include <QtCore/QList>
#include "generatoridgenerator.h"
```

Wykres zależności załączania dla state.h:



Komponenty

- class `obsolete::State`

Klasa przechowująca wszelkie informacje o stanie aplikacji, bazy danych i danych.

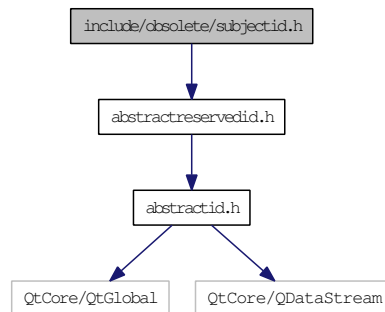
Przestrzenie nazw

- namespace `obsolete`

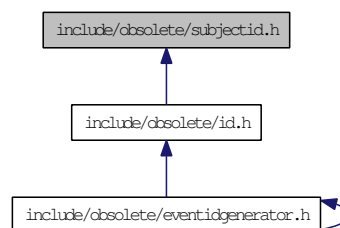
9.17 Dokumentacja pliku include/obsolete/subjectid.h

```
#include "abstractreservedid.h"
```

Wykres zależności załączania dla subjectid.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::SubjectID`
Klasa reprezentująca identyfikator przedmiotu.

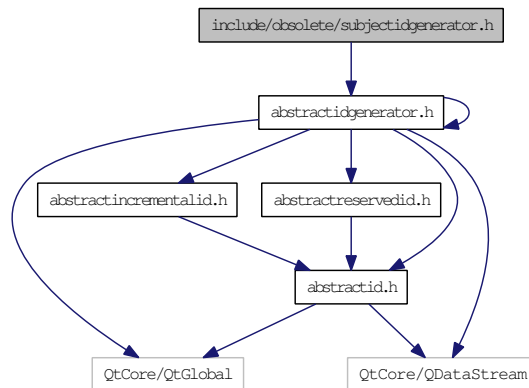
Przestrzeń nazw

- namespace `obsolete`

9.18 Dokumentacja pliku include/obsolete/subjectidgenerator.h

```
#include "abstractidgenerator.h"
```

Wykres zależności załączania dla subjectidgenerator.h:



Komponenty

- class [obsolete::SubjectIDGenerator](#)

Klasa generatorów generatorów identyfikatorów przedmiotów.

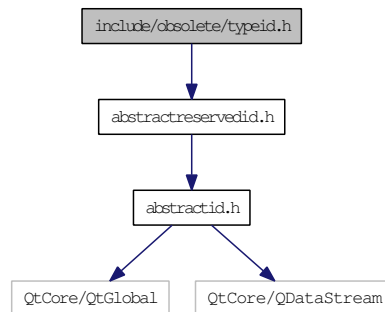
Przestrzenie nazw

- namespace [obsolete](#)

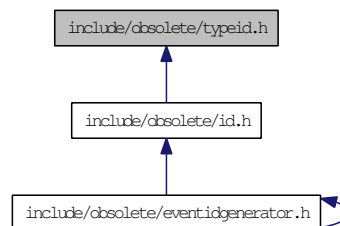
9.19 Dokumentacja pliku include/obsolete/typeid.h

```
#include "abstractreservedid.h"
```

Wykres zależności załączania dla typeid.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class `obsolete::TypeID`
Klasa reprezentująca identyfikator typu wydarzenia.

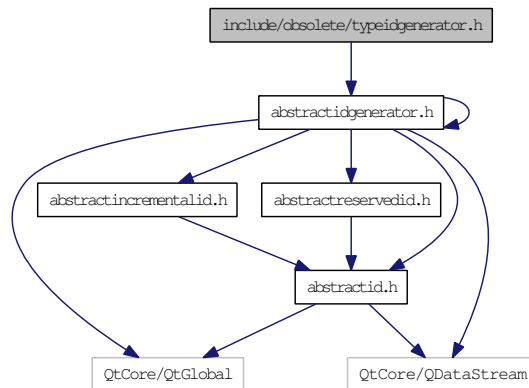
Przestrzeń nazw

- namespace `obsolete`

9.20 Dokumentacja pliku include/obsolete/typeidgenerator.h

```
#include "abstractidgenerator.h"
```

Wykres zależności załączania dla typeidgenerator.h:



Komponenty

- class `obsolete::TypeIDGenerator`
Generator generatorów obiektów `TypeID`.

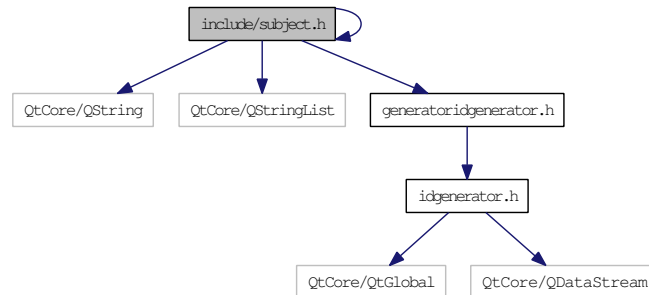
Przestrzenie nazw

- namespace `obsolete`

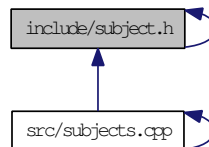
9.21 Dokumentacja pliku include/subject.h

```
#include <QtCore/QString>
#include <QtCore/QStringList>
#include "generatoridgenerator.h"
#include "type.h"
```

Wykres zależności załączania dla subject.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

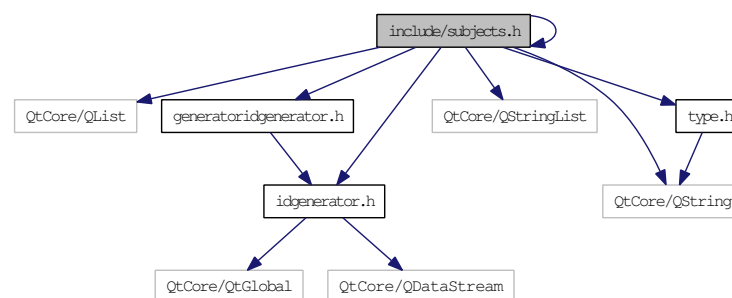
- class [Subject](#)

Reprezentuje informacje o pojedynczym przedmiocie.

9.22 Dokumentacja pliku include/subjects.h

```
#include <QtCore/QList>
#include "idgenerator.h"
#include "subject.h"
#include <QtCore/QString>
#include <QtCore/QStringList>
#include "generatoridgenerator.h"
#include "type.h"
```

Wykres zależności załączania dla subjects.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

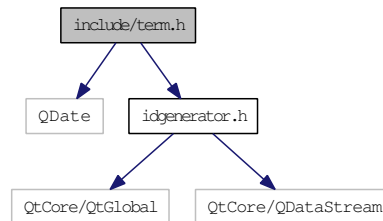
- class [Subjects](#)

Obiekt klasy reprezentuje wszystkie przedmioty.

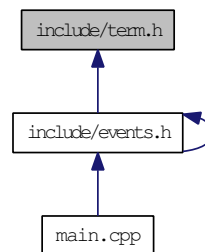
9.23 Dokumentacja pliku include/term.h

```
#include <QDate>
#include "idgenerator.h"
```

Wykres zależności załączania dla term.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

- class [Term](#)

Reprezentacja egzekucji wydarzenia.

Funkcje

- `QDataStream & operator<< (QDataStream &stream, const Term &term)`

Zapisuje dany termin do danego strumienia.

- `QDataStream & operator>> (QDataStream &stream, Term &term)`

Odczytuje dany termin z danego strumienia.

9.23.1 Dokumentacja funkcji

9.23.1.1 QDataStream& operator<< (QDataStream & stream, const Term & term)

Zapisuje dany termin do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany termin.

term Termin który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 3 pliku term.cpp.

Odwołuje się do Term::eventID i Term::markID.

```
4 {  
5     const QDate &date=term;  
6     return stream<<date<<term.eventID<<term.markID;  
7 }
```

9.23.1.2 QDataStream& operator>> (QDataStream & stream, Term & term)

Odczytuje dany termin z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany termin.

term Termin który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 9 pliku term.cpp.

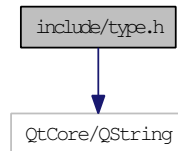
Odwołuje się do Term::eventID i Term::markID.

```
10 {  
11     QDate &date=term;  
12     return stream>>date>>term.eventID>>term.markID;  
13 }
```

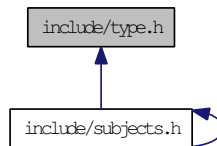
9.24 Dokumentacja pliku include/type.h

```
#include <QtCore/QString>
```

Wykres zależności załączania dla type.h:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Komponenty

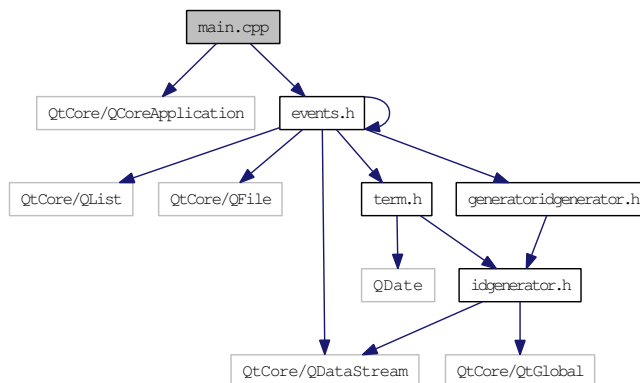
- class [Type](#)

9.25 Dokumentacja pliku main.cpp

```
#include <QtCore/QCoreApplication>
```

```
#include "events.h"
```

Wykres zależności załączania dla main.cpp:



Funkcje

- `int main (int argc, char **argv)`

9.25.1 Dokumentacja funkcji

9.25.1.1 `int main (int argc, char **argv)`

Definicja w linii 4 pliku main.cpp.

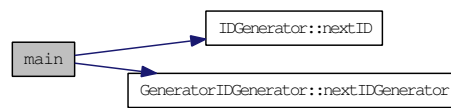
Odwołuje się do `IDGenerator::nextID()` i `GeneratorIDGenerator::nextIDGenerator()`.

```

5 {
6     QCoreApplication app(argc,argv);
7     QFile file("/tmp/tmpfile.txt");
8
9     Events::saveToFile(file);
10
11     IDGenerator eventIDGenerator=GeneratorIDGenerator::nextIDGenerator();
12     IDGenerator markIDGenerator=GeneratorIDGenerator::nextIDGenerator();
13
14     IDGenerator::ID id1=eventIDGenerator.nextID();
15     IDGenerator::ID id2=eventIDGenerator.nextID();
16
17     IDGenerator::ID id3=markIDGenerator.nextID();
18     IDGenerator::ID id4=markIDGenerator.nextID();
19
20     //file.open(QIODevice::WriteOnly);
21     //QDataStream output(&file);
22
23     //output<<id1<<id2<<id3<<id4;
24
25     return 0;
26 }

```

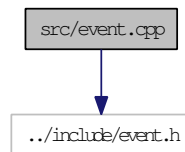
Oto graf wywołań dla tej funkcji:



9.26 Dokumentacja pliku src/event.cpp

```
#include "../include/event.h"
```

Wykres zależności załączania dla event.cpp:



Funkcje

- `QDataStream & operator<< (QDataStream &stream, const Event &event)`
Zapisuje informacje o danym wydarzeniu do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, Event &event)`
Odczytuje informacje o danym wydarzeniu z danego strumienia.

9.26.1 Dokumentacja funkcji

9.26.1.1 QDataStream& operator<< (QDataStream & stream, const Event & event)

Zapisuje informacje o danym wydarzeniu do danego strumienia.

Parametry

stream Strumień do którego będą zapisywane dane.

event Wydarzenie które będzie zapisywane.

Zwraca

Ten sam strumień.

Definicja w linii 3 pliku event.cpp.

```
4 {  
5     return stream<<event.subjectID<<event.typeID<<event.terms;  
6 }
```

9.26.1.2 QDataStream& operator>> (QDataStream & stream, Event & event)

Odczytuje informacje o danym wydarzeniu z danego strumienia.

Parametry

stream Strumień z którego będą odczytywane dane.

event Wydarzenie które zostanie zainicjalizowane wczytanymi danymi.

Zwraca

Ten sam strumień.

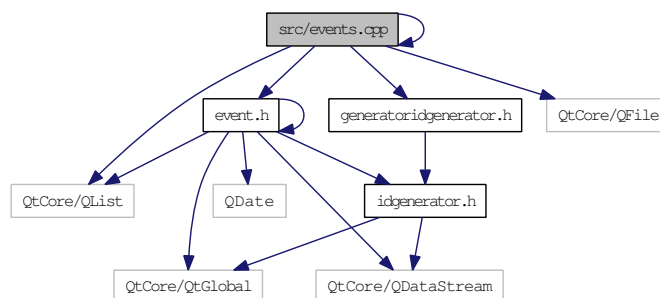
Definicja w linii 8 pliku event.cpp.

```
9 {  
10     return stream>>event.subjectID<<event.typeID<<event.terms;  
11 }
```

9.27 Dokumentacja pliku src/events.cpp

```
#include "../include/events.h"
#include <QtCore/QList>
#include <QtCore/QFile>
#include "event.h"
#include "generatoridgenerator.h"
```

Wykres zależności załączania dla events.cpp:



Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



Funkcje

- `QDataStream & operator<< (QDataStream &stream, const Events &events)`
Zapisuje informacje o wydarzeniach do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, Events &events)`
Odczytuje informacje o wydarzeniach z danego strumienia.

9.27.1 Dokumentacja funkcji

9.27.1.1 QDataStream& operator<< (QDataStream & stream, const Events & events)

Zapisuje informacje o wydarzeniach do danego strumienia.

Parametry

stream Strumień do którego będą zapisywane dane.

events Wydarzenia które będzie zapisywane.

Zwraca

Ten sam strumień.

Definicja w linii 5 pliku events.cpp.

Odwołuje się do Events::eventIDGenerator i Events::events.

```
6 {  
7     return stream<<events.eventIDGenerator<<events.events;  
8 }
```

9.27.1.2 QDataStream& operator>> (QDataStream & stream, Events & events)

Odczytuje informacje o wydarzeniach z danego strumienia.

Parametry

stream Strumień z którego będą odczytywane dane.

events Obiekt klasy [Events](#) który zostanie zainicjalizowany wczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 10 pliku events.cpp.

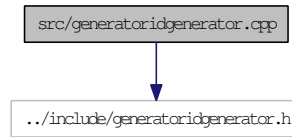
Odwołuje się do Events::eventIDGenerator i Events::events.

```
11 {  
12     return stream>>events.eventIDGenerator>>events.events;  
13 }
```

9.28 Dokumentacja pliku src/generatoridgenerator.cpp

```
#include "../include/generatoridgenerator.h"
```

Wykres zależności załączania dla generatoridgenerator.cpp:

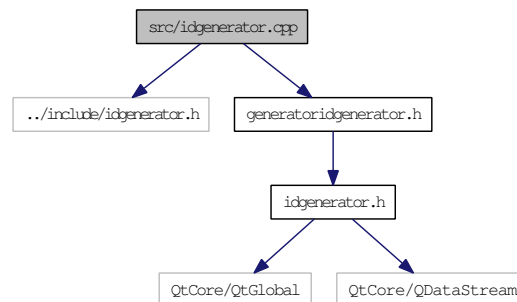


9.29 Dokumentacja pliku src/idgenerator.cpp

```
#include "../include/idgenerator.h"
```

```
#include "generatoridgenerator.h"
```

Wykres zależności załączania dla idgenerator.cpp:



Funkcje

- `QDataStream & operator<< (QDataStream &stream, const IDGenerator::ID &id)`
Zapisuje dany identyfikator do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, IDGenerator::ID &id)`
Odczytuje dany identyfikator z danego strumienia.
- `QDataStream & operator<< (QDataStream &stream, const IDGenerator &generator)`
Zapisuje dany generator identyfikatorów do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, IDGenerator &generator)`
Odczytuje dany generator identyfikatorów z danego strumienia.

9.29.1 Dokumentacja funkcji

9.29.1.1 QDataStream& operator<< (QDataStream & stream, const IDGenerator & generator)

Zapisuje dany generator identyfikatorów do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany generator identyfikatorów.

generator Generator identyfikatorów który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 30 pliku idgenerator.cpp.

Odwołuje się do IDGenerator::freeID i IDGenerator::type.

```
31 {  
32     return stream<<generator.type<<generator.freeID;  
33 }
```

9.29.1.2 QDataStream& operator<< (QDataStream & *stream*, const IDGenerator::ID & *id*)

Zapisuje dany identyfikator do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany identyfikator.

id Identyfikator który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 20 pliku idgenerator.cpp.

```
21 {  
22     return stream<<id.type<<id.id;  
23 }
```

9.29.1.3 QDataStream& operator>> (QDataStream & *stream*, IDGenerator & *generator*)

Odczytuje dany generator identyfikatorów z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany generator identyfikatorów.

generator Generator identyfikatorów który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 35 pliku idgenerator.cpp.

Odwołuje się do IDGenerator::freeID i IDGenerator::type.

```
36 {  
37     return stream>>generator.type>>generator.freeID;  
38 }
```

9.29.1.4 QDataStream& operator>> (QDataStream & *stream*, IDGenerator::ID & *id*)

Odczytuje dany identyfikator z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany identyfikator.

id Identyfikator który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

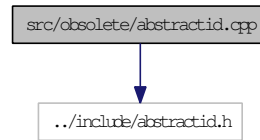
Definicja w linii 25 pliku idgenerator.cpp.

```
26 {  
27     return stream>>id.type>>id.id;  
28 }
```

9.30 Dokumentacja pliku src/obsolete/abstractid.cpp

```
#include "../include/abstractid.h"
```

Wykres zależności załączania dla abstractid.cpp:



Funkcje

- QDataStream & **operator<<** (QDataStream &stream, const AbstractID &ID)
- QDataStream & **operator>>** (QDataStream &stream, AbstractID &ID)

9.30.1 Dokumentacja funkcji

9.30.1.1 QDataStream& operator<< (QDataStream & *stream*, const AbstractID & *ID*)

Definicja w linii 3 pliku abstractid.cpp.

```
4 {  
5     return stream<<ID.ID;  
6 }
```

9.30.1.2 QDataStream& operator>> (QDataStream & *stream*, AbstractID & *ID*)

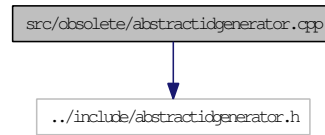
Definicja w linii 8 pliku abstractid.cpp.

```
9 {  
10     return stream>>ID.ID;  
11 }
```

9.31 Dokumentacja pliku src/obsolete/abstractidgenerator.cpp

```
#include "../include/abstractidgenerator.h"
```

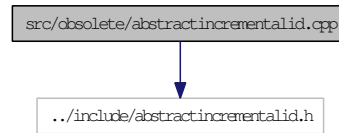
Wykres zależności załączania dla abstractidgenerator.cpp:



9.32 Dokumentacja pliku src/obsolete/abstractincrementalid.cpp

```
#include "../include/abstractincrementalid.h"
```

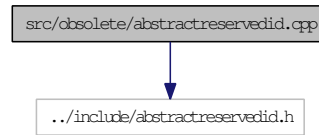
Wykres zależności załączania dla abstractincrementalid.cpp:



9.33 Dokumentacja pliku src/obsolete/abstractreservedid.cpp

```
#include "../include/abstractreservedid.h"
```

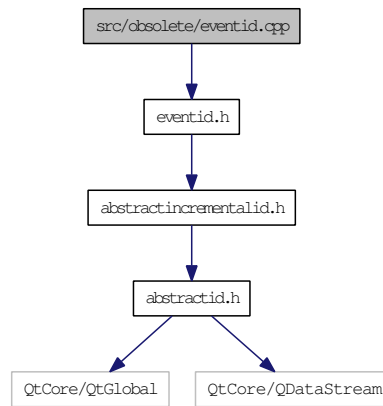
Wykres zależności załączania dla abstractreservedid.cpp:



9.34 Dokumentacja pliku src/obsolete/eventid.cpp

```
#include "eventid.h"
```

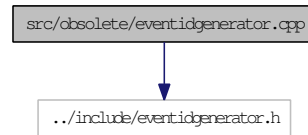
Wykres zależności załączania dla eventid.cpp:



9.35 Dokumentacja pliku src/obsolete/eventidgenerator.cpp

```
#include "../include/eventidgenerator.h"
```

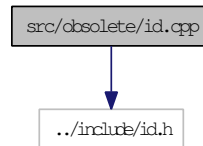
Wykres zależności załączania dla eventidgenerator.cpp:



9.36 Dokumentacja pliku src/obsolete/id.cpp

```
#include "../include/id.h"
```

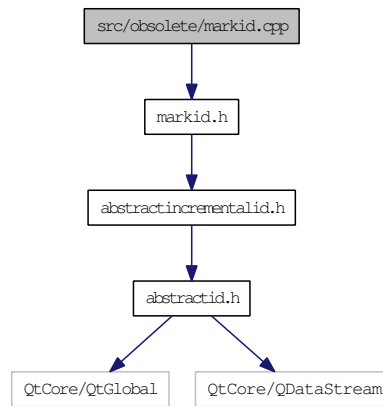
Wykres zależności załączania dla id.cpp:



9.37 Dokumentacja pliku src/obsolete/markid.cpp

```
#include "markid.h"
```

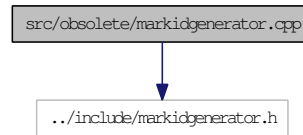
Wykres zależności załączania dla markid.cpp:



9.38 Dokumentacja pliku src/obsolete/markidgenerator.cpp

```
#include "../include/markidgenerator.h"
```

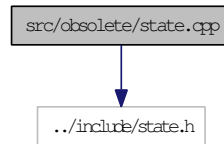
Wykres zależności załączania dla markidgenerator.cpp:



9.39 Dokumentacja pliku src/obsolete/state.cpp

```
#include "../include/state.h"
```

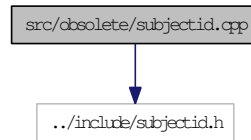
Wykres zależności załączania dla state.cpp:



9.40 Dokumentacja pliku src/obsolete/subjectid.cpp

```
#include "../include/subjectid.h"
```

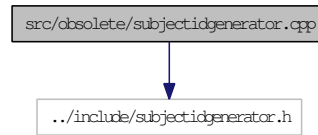
Wykres zależności załączania dla subjectid.cpp:



9.41 Dokumentacja pliku src/obsolete/subjectidgenerator.cpp

```
#include "../include/subjectidgenerator.h"
```

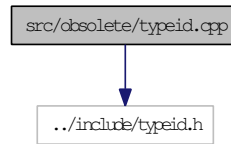
Wykres zależności załączania dla subjectidgenerator.cpp:



9.42 Dokumentacja pliku src/obsolete/typeid.cpp

```
#include "../include/typeid.h"
```

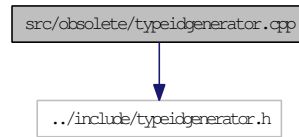
Wykres zależności załączania dla typeid.cpp:



9.43 Dokumentacja pliku src/obsolete/typeidgenerator.cpp

```
#include "../include/typeidgenerator.h"
```

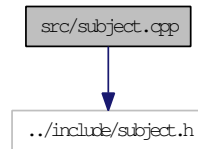
Wykres zależności załączania dla typeidgenerator.cpp:



9.44 Dokumentacja pliku src/subject.cpp

```
#include "../include/subject.h"
```

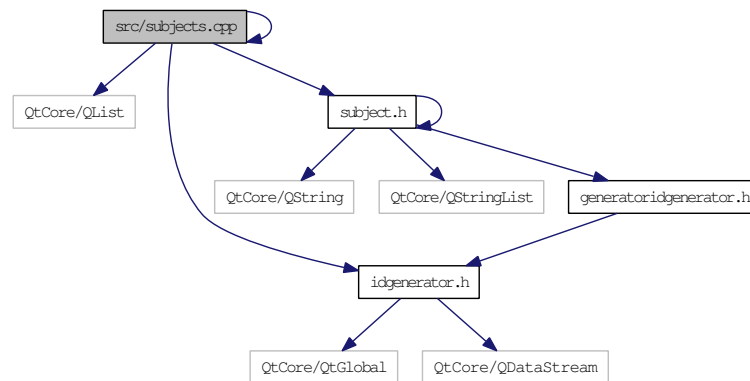
Wykres zależności załączania dla subject.cpp:



9.45 Dokumentacja pliku src/subjects.cpp

```
#include "../include/subjects.h"  
#include <QtCore/QList>  
#include "idgenerator.h"  
#include "subject.h"
```

Wykres zależności załączania dla subjects.cpp:



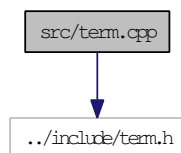
Ten wykres pokazuje, które pliki bezpośrednio lub pośrednio załączają ten plik:



9.46 Dokumentacja pliku src/term.cpp

```
#include "../include/term.h"
```

Wykres zależności załączania dla term.cpp:



Funkcje

- `QDataStream & operator<< (QDataStream &stream, const Term &term)`
Zapisuje dany termin do danego strumienia.
- `QDataStream & operator>> (QDataStream &stream, Term &term)`
Odczytuje dany termin z danego strumienia.

9.46.1 Dokumentacja funkcji

9.46.1.1 QDataStream& operator<< (QDataStream & stream, const Term & term)

Zapisuje dany termin do danego strumienia.

Parametry

stream Strumień do którego będzie zapisany termin.

term Termin który zostanie zapisany.

Zwraca

Ten sam strumień.

Definicja w linii 3 pliku term.cpp.

Odwołuje się do `Term::eventID` i `Term::markID`.

```
4 {  
5     const QDate &date=term;  
6     return stream<<date<<term.eventID<<term.markID;  
7 }
```

9.46.1.2 QDataStream& operator>> (QDataStream & stream, Term & term)

Odczytuje dany termin z danego strumienia.

Parametry

stream Strumień z którego będzie odczytany termin.

term Termin który zostanie zainicjalizowany odczytanymi danymi.

Zwraca

Ten sam strumień.

Definicja w linii 9 pliku term.cpp.

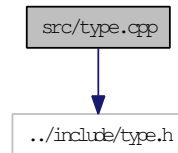
Odwołuje się do Term::eventID i Term::markID.

```
10 {  
11     QDate &date=term;  
12     return stream>>date>>term.eventID>>term.markID;  
13 }
```

9.47 Dokumentacja pliku src/type.cpp

```
#include "../include/type.h"
```

Wykres zależności załączania dla type.cpp:



Skorowidz

- AbstractID
 - [obsolete::AbstractID, 23](#)
- abstractid.cpp
 - [operator<<, 138](#)
 - [operator>>, 138](#)
- AbstractIDGenerator
 - [obsolete::AbstractIDGenerator, 26](#)
- AbstractIncrementalID
 - [obsolete::AbstractIncrementalID, 29](#)
- AbstractReservedID
 - [obsolete::AbstractReservedID, 32](#)
- addEvent
 - [Events, 45](#)
- ARCHITECTURE_ID
 - [CMakeCCompilerId.c, 97](#)
 - [CMakeCXXCompilerId.cpp, 99](#)
- build/CMakeFiles/CompilerIdC/CMakeCCompilerId.c
 - [97](#)
- build/CMakeFiles/CompilerIdCXX/CMakeCXXCompilerId.cpp
 - [99](#)
- CMakeCCompilerId.c
 - [ARCHITECTURE_ID, 97](#)
 - [COMPILER_ID, 97](#)
 - [info_arch, 98](#)
 - [info_compiler, 98](#)
 - [info_platform, 98](#)
 - [main, 98](#)
 - [PLATFORM_ID, 97](#)
- CMakeCXXCompilerId.cpp
 - [ARCHITECTURE_ID, 99](#)
 - [COMPILER_ID, 99](#)
 - [info_arch, 100](#)
 - [info_compiler, 100](#)
 - [info_platform, 100](#)
 - [main, 99](#)
 - [PLATFORM_ID, 99](#)
- COMPILER_ID
 - [CMakeCCompilerId.c, 97](#)
 - [CMakeCXXCompilerId.cpp, 99](#)
- create
 - [Events, 45](#)
 - [Subjects, 82](#)
- createID
 - [obsolete::AbstractIDGenerator, 26](#)
 - [obsolete::AbstractReservedID, 32](#)
 - [obsolete::EventIDGenerator, 42](#)
 - [obsolete::ID, 57](#)
 - [obsolete::MarkIDGenerator, 69](#)
 - [obsolete::SubjectID, 77](#)
 - [obsolete::SubjectIDGenerator, 80](#)
 - [obsolete::TypeID, 92](#)
 - [obsolete::TypeIDGenerator, 95](#)
- createVoidID
 - [obsolete::AbstractIDGenerator, 26](#)
 - [obsolete::EventIDGenerator, 42](#)
 - [obsolete::MarkIDGenerator, 69](#)
 - [obsolete::SubjectIDGenerator, 80](#)
 - [obsolete::TypeIDGenerator, 95](#)
- Dokumentacja katalogu build/, [11](#)
- Dokumentacja katalogu build/CMakeFiles/, [12](#)
- Dokumentacja katalogu build/CMakeFiles/CompilerIdC/, [13](#)
- Dokumentacja katalogu build/CMakeFiles/CompilerIdCXX/, [14](#)
- Dokumentacja katalogu include/, [15](#)
- Dokumentacja katalogu include/obsolete/, [17](#)
- Dokumentacja katalogu src/, [18](#)
- Dokumentacja katalogu src/obsolete/, [16](#)
- Event, [33](#)
 - [Event, 34](#)
 - [operator<<, 35](#)
 - [operator>>, 35](#)
 - [subjectID, 35](#)
 - [terms, 35](#)
 - [typeID, 36](#)
- event.cpp
 - [operator<<, 130](#)
 - [operator>>, 130](#)
- event.h
 - [operator<<, 101](#)
 - [operator>>, 102](#)
- EventID
 - [obsolete::EventID, 39](#)
- eventID
 - [Term, 87](#)
- eventIDGenerator

- Events, 46
- Events, 43
 - addEvent, 45
 - create, 45
 - eventIDGenerator, 46
 - Events, 45
 - events, 46
 - existOne, 47
 - initialized, 47
 - operator<<, 46
 - operator>>, 46
- events
 - Events, 46
- events.cpp
 - operator<<, 132
 - operator>>, 133
- events.h
 - operator<<, 104
 - operator>>, 104
- exist
 - Subject, 74
 - Type, 88
- existOne
 - Events, 47
 - Subjects, 83
- freeID
 - IDGenerator, 63
- freeIDGenerator
 - GeneratorIDGenerator, 50
- GeneratorIDGenerator, 48
 - freeIDGenerator, 50
 - GeneratorIDGenerator, 49
 - IDGenerator, 62
 - nextIDGenerator, 49
 - voidIDGenerator, 49
- ID
 - IDGenerator::ID, 52
 - obsolete::AbstractID, 23
 - obsolete::AbstractIDGenerator, 26
 - obsolete::AbstractIncrementalID, 29
 - obsolete::AbstractReservedID, 32
 - obsolete::EventID, 39
 - obsolete::EventIDGenerator, 42
 - obsolete::ID, 57, 58
 - obsolete::MarkID, 66
 - obsolete::MarkIDGenerator, 69
 - obsolete::SubjectID, 77
 - obsolete::SubjectIDGenerator, 80
 - obsolete::TypeID, 92
 - obsolete::TypeIDGenerator, 95
- id
 - IDGenerator::ID, 54
- IDGenerator, 59
 - freeID, 63
 - GeneratorIDGenerator, 62
 - IDGenerator, 61
 - IDGenerator::ID, 53
 - initialize, 61
 - nextID, 61
 - operator<<, 62
 - operator>>, 62
 - operator=, 61
 - type, 63
 - voidID, 62
- idgenerator.cpp
 - operator<<, 135, 136
 - operator>>, 136
- idgenerator.h
 - operator<<, 107
 - operator>>, 107, 108
- IDGenerator::ID, 51
 - ID, 52
 - id, 54
 - IDGenerator, 53
 - operator<<, 53
 - operator>>, 53
 - operator=, 53
 - type, 54
- include/event.h, 101
- include/events.h, 103
- include/generatoridgenerator.h, 105
- include/idgenerator.h, 106
- include/obsolete/abstractid.h, 109
- include/obsolete/abstractidgenerator.h, 110
- include/obsolete/abstractincrementalid.h, 111
- include/obsolete/abstractreservedid.h, 112
- include/obsolete/eventid.h, 113
- include/obsolete/eventidgenerator.h, 114
- include/obsolete/id.h, 115
- include/obsolete/markid.h, 116
- include/obsolete/markidgenerator.h, 117
- include/obsolete/state.h, 118
- include/obsolete/subjectid.h, 119
- include/obsolete/subjectidgenerator.h, 120
- include/obsolete/typeid.h, 121
- include/obsolete/typeidgenerator.h, 122
- include/subject.h, 123
- include/subjects.h, 124
- include/term.h, 125
- include/type.h, 127
- info_arch
 - CMakeCCompilerId.c, 98
 - CMakeCXXCompilerId.cpp, 100
- info_compiler
 - CMakeCCompilerId.c, 98

- CMakeCXXCompilerId.cpp, 100
- info_platform
 - CMakeCCompilerId.c, 98
 - CMakeCXXCompilerId.cpp, 100
- initialize
 - IDGenerator, 61
- initialized
 - Events, 47
 - Subjects, 83
- main
 - CMakeCCompilerId.c, 98
 - CMakeCXXCompilerId.cpp, 99
 - main.cpp, 128
- main.cpp, 128
 - main, 128
- MarkID
 - obsolete::MarkID, 66
- markID
 - Term, 87
- name
 - Subject, 74
 - Type, 88
- nextID
 - IDGenerator, 61
- nextIDGenerator
 - GeneratorIDGenerator, 49
- obsolete, 19
 - operator<<, 20
 - operator>>, 20
- obsolete::AbstractID, 21
 - AbstractID, 23
 - ID, 23
 - operator<<, 23
 - operator>>, 23
- obsolete::AbstractIDGenerator, 24
 - AbstractIDGenerator, 26
 - createID, 26
 - createVoidID, 26
 - ID, 26
- obsolete::AbstractIncrementalID, 27
 - AbstractIncrementalID, 29
 - ID, 29
- obsolete::AbstractReservedID, 30
 - AbstractReservedID, 32
 - createID, 32
 - ID, 32
- obsolete::EventID, 37
 - EventID, 39
 - ID, 39
- obsolete::EventIDGenerator, 40
 - createID, 42
 - createVoidID, 42
 - ID, 42
- obsolete::ID, 55
 - createID, 57
 - ID, 57, 58
- obsolete::MarkID, 64
 - ID, 66
 - MarkID, 66
- obsolete::MarkIDGenerator, 67
 - createID, 69
 - createVoidID, 69
 - ID, 69
- obsolete::State, 71
 - State, 72
 - subjectIDGenerator, 72
 - subjectMapping, 72
 - subjects, 72
- obsolete::SubjectID, 75
 - createID, 77
 - ID, 77
- obsolete::SubjectIDGenerator, 78
 - createID, 80
 - createVoidID, 80
 - ID, 80
- obsolete::TypeID, 90
 - createID, 92
 - ID, 92
- obsolete::TypeIDGenerator, 93
 - createID, 95
 - createVoidID, 95
 - ID, 95
- operator<<
 - abstractid.cpp, 138
 - Event, 35
 - event.cpp, 130
 - event.h, 101
 - Events, 46
 - events.cpp, 132
 - events.h, 104
 - IDGenerator, 62
 - idgenerator.cpp, 135, 136
 - idgenerator.h, 107
 - IDGenerator::ID, 53
 - obsolete, 20
 - obsolete::AbstractID, 23
 - Term, 86
 - term.cpp, 154
 - term.h, 125
- operator>>
 - abstractid.cpp, 138
 - Event, 35
 - event.cpp, 130
 - event.h, 102
 - Events, 46

- events.cpp, [133](#)
- events.h, [104](#)
- IDGenerator, [62](#)
- idgenerator.cpp, [136](#)
- idgenerator.h, [107](#), [108](#)
- IDGenerator::ID, [53](#)
- obsolete, [20](#)
- obsolete::AbstractID, [23](#)
- Term, [87](#)
- term.cpp, [154](#)
- term.h, [126](#)
- operator=
 - IDGenerator, [61](#)
 - IDGenerator::ID, [53](#)
- PLATFORM_ID
 - CMakeCCompilerId.c, [97](#)
 - CMakeCXXCompilerId.cpp, [99](#)
- QDate, [70](#)
- revoke
 - Term, [87](#)
- src/event.cpp, [130](#)
- src/events.cpp, [132](#)
- src/generatoridgenerator.cpp, [134](#)
- src/idgenerator.cpp, [135](#)
- src/obsolete/abstractid.cpp, [138](#)
- src/obsolete/abstractidgenerator.cpp, [139](#)
- src/obsolete/abstractincrementalid.cpp, [140](#)
- src/obsolete/abstractreservedid.cpp, [141](#)
- src/obsolete/eventid.cpp, [142](#)
- src/obsolete/eventidgenerator.cpp, [143](#)
- src/obsolete/id.cpp, [144](#)
- src/obsolete/markid.cpp, [145](#)
- src/obsolete/markidgenerator.cpp, [146](#)
- src/obsolete/state.cpp, [147](#)
- src/obsolete/subjectid.cpp, [148](#)
- src/obsolete/subjectidgenerator.cpp, [149](#)
- src/obsolete/typeid.cpp, [150](#)
- src/obsolete/typeidgenerator.cpp, [151](#)
- src/subject.cpp, [152](#)
- src/subjects.cpp, [153](#)
- src/term.cpp, [154](#)
- src/type.cpp, [156](#)
- State
 - obsolete::State, [72](#)
- Subject, [73](#)
 - exist, [74](#)
 - name, [74](#)
 - typeIDGenerator, [74](#)
 - types, [74](#)
 - typesMapping, [74](#)
- subjectID
 - Event, [35](#)
- subjectIDGenerator
 - obsolete::State, [72](#)
 - Subjects, [83](#)
- subjectMapping
 - obsolete::State, [72](#)
 - Subjects, [83](#)
- Subjects, [81](#)
 - create, [82](#)
 - existOne, [83](#)
 - initialized, [83](#)
 - subjectIDGenerator, [83](#)
 - subjectMapping, [83](#)
 - subjects, [83](#)
- subjects
 - obsolete::State, [72](#)
 - Subjects, [83](#)
- Term, [84](#)
 - eventID, [87](#)
 - markID, [87](#)
 - operator<<, [86](#)
 - operator>>, [87](#)
 - revoke, [87](#)
 - Term, [86](#)
- term.cpp
 - operator<<, [154](#)
 - operator>>, [154](#)
- term.h
 - operator<<, [125](#)
 - operator>>, [126](#)
- terms
 - Event, [35](#)
- Type, [88](#)
 - exist, [88](#)
 - name, [88](#)
- type
 - IDGenerator, [63](#)
 - IDGenerator::ID, [54](#)
- typeID
 - Event, [36](#)
- typeIDGenerator
 - Subject, [74](#)
- types
 - Subject, [74](#)
- typesMapping
 - Subject, [74](#)
- voidID
 - IDGenerator, [62](#)
- voidIDGenerator
 - GeneratorIDGenerator, [49](#)