

ROC

V1.0

Robust Optimization C++ Reference Mannual

by Meilin ZHANG

Mon Jun 17 2013 12:52:10

Contents

1	Hierarchical Index	3
1.1	Class Hierarchy	3
2	Class Index	5
2.1	Class List	5
3	Class Documentation	7
3.1	ROBaseConstraint Class Reference	7
3.1.1	Detailed Description	7
3.1.2	Constructor & Destructor Documentation	7
3.1.2.1	ROBaseConstraint	7
3.1.2.2	ROBaseConstraint	8
3.1.2.3	ROBaseConstraint	8
3.1.3	Member Function Documentation	8
3.1.3.1	getTag	8
3.1.3.2	operator=	8
3.1.3.3	operator=	8
3.2	ROBinVar Class Reference	9
3.2.1	Detailed Description	9
3.2.2	Constructor & Destructor Documentation	9
3.2.2.1	ROBinVar	9
3.2.2.2	ROBinVar	9
3.2.3	Member Function Documentation	9
3.2.3.1	getVar	9
3.3	ROConstraint Class Reference	10
3.3.1	Detailed Description	10
3.3.2	Constructor & Destructor Documentation	10
3.3.2.1	ROConstraint	10
3.3.2.2	ROConstraint	11
3.3.2.3	ROConstraint	11
3.3.2.4	ROConstraint	11
3.3.2.5	ROConstraint	11

3.3.2.6	ROConstraint	12
3.3.2.7	ROConstraint	12
3.3.2.8	ROConstraint	12
3.3.2.9	ROConstraint	12
3.3.2.10	ROConstraint	13
3.3.2.11	ROConstraint	13
3.3.2.12	ROConstraint	13
3.3.2.13	ROConstraint	13
3.3.3	Member Function Documentation	13
3.3.3.1	operator=	13
3.4	ROConstraintSet Class Reference	14
3.4.1	Detailed Description	14
3.4.2	Constructor & Destructor Documentation	15
3.4.2.1	ROConstraintSet	15
3.4.2.2	ROConstraintSet	15
3.4.2.3	ROConstraintSet	15
3.4.2.4	ROConstraintSet	15
3.4.2.5	ROConstraintSet	15
3.4.2.6	ROConstraintSet	15
3.4.2.7	ROConstraintSet	16
3.4.2.8	ROConstraintSet	16
3.4.2.9	ROConstraintSet	16
3.4.2.10	ROConstraintSet	16
3.4.2.11	ROConstraintSet	16
3.4.3	Member Function Documentation	16
3.4.3.1	add	17
3.4.3.2	add	17
3.4.3.3	add	17
3.4.3.4	add	17
3.4.3.5	operator=	17
3.4.3.6	set	18
3.5	ROException Class Reference	18
3.5.1	Detailed Description	18
3.6	ROExceptionDivideZero Class Reference	19
3.6.1	Detailed Description	19
3.7	ROExceptionDynamicCast Class Reference	19
3.7.1	Detailed Description	19
3.8	ROExceptionIncompatibleExpr Class Reference	20
3.8.1	Detailed Description	20
3.9	ROExceptionModelUnsolve Class Reference	21

3.9.1 Detailed Description	21
3.10 ROExceptionNullPointer Class Reference	21
3.10.1 Detailed Description	22
3.11 ROExceptionOutOfRange Class Reference	22
3.11.1 Detailed Description	22
3.12 ROExceptionTypeIncompatible Class Reference	22
3.12.1 Detailed Description	23
3.13 ROExceptionUnExpectCount Class Reference	23
3.13.1 Detailed Description	23
3.14 ROExpr Class Reference	24
3.14.1 Detailed Description	24
3.14.2 Constructor & Destructor Documentation	25
3.14.2.1 ROExpr	25
3.14.2.2 ROExpr	25
3.14.2.3 ROExpr	25
3.14.2.4 ROExpr	25
3.14.2.5 ROExpr	25
3.14.2.6 ROExpr	25
3.14.2.7 ROExpr	26
3.14.2.8 ~ROExpr	26
3.14.3 Member Function Documentation	26
3.14.3.1 expect	26
3.14.3.2 expect	26
3.14.3.3 operator*=	26
3.14.3.4 operator*=	27
3.14.3.5 operator*=	27
3.14.3.6 operator*=	27
3.14.3.7 operator+=	27
3.14.3.8 operator+=	28
3.14.3.9 operator+=	28
3.14.3.10 operator+=	28
3.14.3.11 operator+=	28
3.14.3.12 operator-=	28
3.14.3.13 operator-=	29
3.14.3.14 operator-=	29
3.14.3.15 operator-=	29
3.14.3.16 operator-=	29
3.14.3.17 operator/=	30
3.14.3.18 operator=	30
3.14.3.19 operator=	30

3.14.3.20 operator=	30
3.14.3.21 operator=	31
3.14.3.22 operator=	31
3.15 ROIntVar Class Reference	31
3.15.1 Detailed Description	31
3.15.2 Constructor & Destructor Documentation	32
3.15.2.1 ROIntVar	32
3.15.2.2 ROIntVar	32
3.15.3 Member Function Documentation	32
3.15.3.1 getVar	32
3.16 ROModel Class Reference	32
3.16.1 Detailed Description	33
3.16.2 Constructor & Destructor Documentation	33
3.16.2.1 ROModel	33
3.16.2.2 ~ROModel	33
3.16.3 Member Function Documentation	33
3.16.3.1 add	33
3.16.3.2 add	33
3.16.3.3 add	34
3.16.3.4 add	34
3.16.3.5 add	34
3.16.3.6 add	34
3.16.3.7 exportModel	34
3.16.3.8 exportModel	35
3.16.3.9 getObjValue	35
3.16.3.10 getStatus	35
3.16.3.11 getVarValue	35
3.16.3.12 solve	35
3.17 ROObjective Class Reference	36
3.17.1 Detailed Description	36
3.17.2 Constructor & Destructor Documentation	36
3.17.2.1 ROObjective	36
3.17.2.2 ROObjective	37
3.17.2.3 ROObjective	37
3.17.3 Member Function Documentation	37
3.17.3.1 getTag	37
3.17.3.2 operator=	37
3.18 ROUn Class Reference	37
3.18.1 Detailed Description	38
3.18.2 Constructor & Destructor Documentation	38

3.18.2.1	ROUn	38
3.18.2.2	ROUn	38
3.18.2.3	~ROUn	38
3.18.3	Member Function Documentation	38
3.18.3.1	getId	38
3.18.3.2	getNo	39
3.19	ROUnLinConstraint Class Reference	39
3.19.1	Detailed Description	39
3.19.2	Constructor & Destructor Documentation	39
3.19.2.1	ROUnLinConstraint	39
3.19.2.2	ROUnLinConstraint	40
3.19.2.3	ROUnLinConstraint	40
3.19.3	Member Function Documentation	40
3.19.3.1	getTag	40
3.19.3.2	operator=	40
3.19.3.3	operator=	40
3.20	ROUnSOCConstraint Class Reference	41
3.20.1	Detailed Description	41
3.20.2	Constructor & Destructor Documentation	41
3.20.2.1	ROUnSOCConstraint	41
3.20.2.2	ROUnSOCConstraint	41
3.20.2.3	ROUnSOCConstraint	41
3.20.2.4	ROUnSOCConstraint	42
3.20.2.5	ROUnSOCConstraint	42
3.20.2.6	ROUnSOCConstraint	42
3.20.3	Member Function Documentation	42
3.20.3.1	operator=	42
3.21	ROUnVarC Class Reference	42
3.21.1	Detailed Description	43
3.21.2	Constructor & Destructor Documentation	43
3.21.2.1	ROUnVarC	43
3.21.2.2	ROUnVarC	43
3.21.3	Member Function Documentation	43
3.21.3.1	expect	43
3.21.3.2	expect	44
3.21.3.3	operator*=	44
3.21.3.4	operator*=	44
3.21.3.5	operator*=	44
3.21.3.6	operator*=	45
3.21.3.7	operator+=	45

3.21.3.8	operator+=	45
3.21.3.9	operator+=	45
3.21.3.10	operator+=	46
3.21.3.11	operator+=	46
3.21.3.12	operator-=	46
3.21.3.13	operator-=	46
3.21.3.14	operator-=	46
3.21.3.15	operator-=	47
3.21.3.16	operator-=	47
3.21.3.17	operator/=	47
3.21.3.18	operator=	47
3.21.3.19	operator=	48
3.22	ROVar Class Reference	48
3.22.1	Detailed Description	48
3.22.2	Constructor & Destructor Documentation	49
3.22.2.1	ROVar	49
3.22.2.2	ROVar	49
3.22.2.3	ROVar	49
3.22.2.4	ROVar	49
3.22.2.5	~ROVar	49
3.22.3	Member Function Documentation	49
3.22.3.1	getId	50
3.22.3.2	getNo	50
3.23	ROVarDR Class Reference	50
3.23.1	Detailed Description	50
3.23.2	Constructor & Destructor Documentation	50
3.23.2.1	ROVarDR	50
3.23.3	Member Function Documentation	51
3.23.3.1	addDR	51
3.23.3.2	addDR	51
3.23.3.3	clone	51
3.23.3.4	getExpr	51
3.23.3.5	getVar	51
3.23.3.6	getVar	52
3.23.3.7	getVar	52

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

exception	
ROException	18
ROExceptionDivideZero	19
ROExceptionDynamicCast	19
ROExceptionIncompatibleExpr	20
ROExceptionModelUnsolve	21
ROExceptionNullPointer	21
ROExceptionOutOfRange	22
ROExceptionTypeIncompatible	22
ROExceptionUnExpectCount	23
ROBaseConstraint	7
ROBinVar	9
ROConstraint	10
ROConstraintSet	14
ROExpr	24
ROIntVar	31
ROModel	32
ROObjective	36
ROUn	37
ROUnLinConstraint	39
ROUnSOCCConstraint	41
ROUnVarC	42
ROVar	48
ROVarDR	50

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

ROBaseConstraint	7
ROBinVar	9
ROConstraint	10
ROConstraintSet	14
ROException	18
ROExceptionDivideZero	19
ROExceptionDynamicCast	19
ROExceptionIncompatibleExpr	20
ROExceptionModelUnsolve	21
ROExceptionNullPointer	21
ROExceptionOutOfRange	22
ROExceptionTypeIncompatible	22
ROExceptionUnExpectCount	23
ROExpr	24
ROIntVar	31
ROModel	32
ROObjective	36
ROUn	37
ROUnLinConstraint	39
ROUnSOCCConstraint	41
ROUnVarC	42
ROVar	48
ROVarDR	50

Chapter 3

Class Documentation

3.1 ROBaseConstraint Class Reference

```
#include <ROBaseConstraint.h>
```

Public Member Functions

- **ROBaseConstraint** (const **ROBaseConstraint** &p_bConst)
- **ROBaseConstraint** (const **ROUnLinConstraint** &p_unConst)
- **ROBaseConstraint** (const ROConstTag p_tag=EnumLeq)
- **ROBaseConstraint** & **operator=** (const **ROBaseConstraint** &p_expr)
- **ROBaseConstraint** & **operator=** (const **ROUnLinConstraint** &p_expr)
- ROConstTag **getTag** () const

3.1.1 Detailed Description

An instance of this class is a base constraint in a model. A base constraint consists of its equation tag and an expression (LHS).

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 19 of file ROBaseConstraint.h.

3.1.2 Constructor & Destructor Documentation

3.1.2.1 ROBaseConstraint::ROBaseConstraint (const ROBaseConstraint & p_bConst)

This is the copy constructor.

Parameters

<i>p_bConst</i>	A reference to an ROBaseConstraint (p. 7) instance.
-----------------	--

3.1.2.2 ROBaseConstraint::ROBaseConstraint (const ROUnLinConstraint & *p_unConst*)

This is the copy constructor.

Parameters

<i>p_unConst</i>	A reference to an ROUnLinConstraint (p. 39) instance.
------------------	--

3.1.2.3 ROBaseConstraint::ROBaseConstraint (const ROConstTag *p_tag* = EnumLeq)

The default constructor of **ROBaseConstraint** (p. 7).

Parameters

<i>p_tag</i>	Indicating the constraint tag, default is <= .
--------------	--

3.1.3 Member Function Documentation

3.1.3.1 ROConstTag ROBaseConstraint::getTag () const

This member function returns the equation tag ("<=" or "==") of invoking object.

Returns

ROConstTag The equation tag ("<=" or "==") of invoking object.

3.1.3.2 ROBaseConstraint& ROBaseConstraint::operator= (const ROBaseConstraint & *p_expr*)

The assignment operator "=".

Parameters

<i>p_expr</i>	A reference to an ROBaseConstraint (p. 7) instance.
---------------	--

Returns

ROBaseConstraint (p. 7)& A reference of current invoking object.

3.1.3.3 ROBaseConstraint& ROBaseConstraint::operator= (const ROUnLinConstraint & *p_expr*)

The assignment operator "=".

Parameters

<i>p_expr</i>	A reference to an ROUnLinConstraint (p. 39) instance.
---------------	--

Returns

ROBaseConstraint (p. 7)& A reference of current invoking object.

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

- include/ROBaseConstraint.h

3.2 ROBinVar Class Reference

```
#include <ROBinVar.h>
```

Public Member Functions

- **ROBinVar** (const std::string &p_name="")
- **ROBinVar** (const **ROBinVar** &p_binVar)
- const **ROVar** &**getVar** () const

3.2.1 Detailed Description

An instance of this class is an binary variable without uncertainty.

Date: 31-May-2013

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 23 of file ROBinVar.h.

3.2.2 Constructor & Destructor Documentation

3.2.2.1 ROBinVar::ROBinVar (const std::string & p_name = " ")

This constructor creates an **ROBinVar** (p. 9) instance from given name.

Parameters

<i>p_name</i>	Name of invoking object.
---------------	--------------------------

3.2.2.2 ROBinVar::ROBinVar (const **ROBinVar** & p_binVar)

The copy constructor.

Parameters

<i>p_binVar</i>	A reference to an ROBinVar (p. 9) instance.
-----------------	--

3.2.3 Member Function Documentation

3.2.3.1 const **ROVar**& ROBinVar::getVar () const

Get the variable reference.

Returns

ROVar (p. 48)& A reference to the invoking object of **ROBinVar** (p. 9).

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

- include/ROBinVar.h

3.3 ROConstraint Class Reference

```
#include <ROConstraint.h>
```

Public Member Functions

- **ROConstraint** ()
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROConstraintSet** &p_uncertaintySet)
- **ROConstraint** (const **ROConstraintSet** &p_varConstSet, const **ROConstraintSet** &p_uncertaintySet)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnLinConstraintArray** &p_unConstArray, const **ROUnSOCCConstraintArray** &p_unNormConstArray)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnLinConstraint** &p_unConst, const **ROUnSOCCConstraintArray** &p_unNormConstArray)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnLinConstraintArray** &p_unConstArray, const **ROUnSOCCConstraint** &p_unNormConst)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnLinConstraint** &p_unConst, const **ROUnSOCCConstraint** &p_unNormConst)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnLinConstraintArray** &p_unConstArray)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnLinConstraint** &p_unConst)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnSOCCConstraintArray** &p_unNormConstArray)
- **ROConstraint** (const **ROBaseConstraint** &p_varConst, const **ROUnSOCCConstraint** &p_unNormConst)
- **ROConstraint** (const **ROBaseConstraint** &p_const)
- **ROConstraint** (const **ROConstraint** &p_const)
- **ROConstraint** & **operator=** (const **ROConstraint** &p_const)

3.3.1 Detailed Description

An instance of this class is a constraint in a model. A constraint could be either a **ROBaseConstraint** (p. 7) or a **ROConstraint** (p. 10) with uncertainty set.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 25 of file ROConstraint.h.

3.3.2 Constructor & Destructor Documentation

3.3.2.1 ROConstraint::ROConstraint ()

The default constructor which creates an empty object. It must be initialized before use.

3.3.2.2 ROConstraint::ROConstraint (const ROBaseConstraint & *p_varConst*, const ROConstraintSet & *p_uncertaintySet*)

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7) and **ROConstraintSet** (p. 14).

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_uncertaintySet</i>	A reference to a ROConstraintSet (p. 14) instance which specifies the set of uncertainty constraints.

3.3.2.3 ROConstraint::ROConstraint (const ROConstraintSet & *p_varConstSet*, const ROConstraintSet & *p_uncertaintySet*)

This constructor creates a **ROConstraint** (p. 10) from **ROConstraintSet** (p. 14) and **ROConstraintSet** (p. 14).

Parameters

<i>p_varConstSet</i>	A reference to a ROConstraintSet (p. 14) instance which specifies a set of base constraints.
<i>p_uncertaintySet</i>	A reference to a ROConstraintSet (p. 14) instance which specifies the set of uncertainty constraints.

3.3.2.4 ROConstraint::ROConstraint (const ROBaseConstraint & *p_varConst*, const ROUnLinConstraintArray & *p_unConstArray*, const ROUnSOCConstraintArray & *p_unNormConstArray*)

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7), **ROUnLinConstraintArray** and **ROUnSOCConstraintArray**.

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unConstArray</i>	A reference to a ROUnLinConstraintArray instance which specifies the base set of uncertainty constraints.
<i>p_unNormConstArray</i>	A reference to a ROUnSOCConstraintArray instance which specifies the norm set of uncertainty constraints.

3.3.2.5 ROConstraint::ROConstraint (const ROBaseConstraint & *p_varConst*, const ROUnLinConstraint & *p_unConst*, const ROUnSOCConstraintArray & *p_unNormConstArray*)

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7), **ROUnLinConstraint** (p. 39) and **ROUnSOCConstraintArray**.

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unConst</i>	A reference to a ROUnLinConstraint (p. 39) instance which specifies an base uncertainty constraint.
<i>p_unNormConstArray</i>	A reference to a ROUnSOCConstraintArray instance which specifies the norm set of uncertainty constraints.

3.3.2.6 `ROConstraint::ROConstraint (const ROBaseConstraint & p_varConst, const ROUnLinConstraintArray & p_unConstArray, const ROUnSOCConstraint & p_unNormConst)`

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7), **ROUnLinConstraintArray** and **ROUnSOCConstraint** (p. 41).

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unConstArray</i>	A reference to a ROUnLinConstraintArray instance which specifies the base set of uncertainty constraints.
<i>p_unNormConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance which specifies a norm uncertainty constraint.

3.3.2.7 `ROConstraint::ROConstraint (const ROBaseConstraint & p_varConst, const ROUnLinConstraint & p_unConst, const ROUnSOCConstraint & p_unNormConst)`

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7), **ROUnLinConstraint** (p. 39) and **ROUnSOCConstraint** (p. 41).

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unConst</i>	A reference to a ROUnLinConstraint (p. 39) instance which specifies an base uncertainty constraint.
<i>p_unNormConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance which specifies a norm uncertainty constraint.

3.3.2.8 `ROConstraint::ROConstraint (const ROBaseConstraint & p_varConst, const ROUnLinConstraintArray & p_unConstArray)`

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7) and **ROUnLinConstraintArray**.

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unConstArray</i>	A reference to a ROUnLinConstraintArray instance which specifies the set of base uncertainty constraints.

3.3.2.9 `ROConstraint::ROConstraint (const ROBaseConstraint & p_varConst, const ROUnLinConstraint & p_unConst)`

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7) and **ROUnLinConstraint** (p. 39).

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unConst</i>	A reference to a ROUnLinConstraint (p. 39) instance which specifies an base uncertainty constraint.

3.3.2.10 ROConstraint::ROConstraint (const ROBaseConstraint & *p_varConst*, const ROUnSOCConstraintArray & *p_unNormConstArray*)

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7) and ROUnSOCConstraintArray.

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unNormConstArray</i>	A reference to a ROUnSOCConstraintArray instance which specifies the norm set of uncertainty constraints.

3.3.2.11 ROConstraint::ROConstraint (const ROBaseConstraint & *p_varConst*, const ROUnSOCConstraint & *p_unNormConst*)

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7) and **ROUnSOCConstraint** (p. 41).

Parameters

<i>p_varConst</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
<i>p_unNormConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance which specifies a norm uncertainty constraint.

3.3.2.12 ROConstraint::ROConstraint (const ROBaseConstraint & *p_const*)

This constructor creates a **ROConstraint** (p. 10) from **ROBaseConstraint** (p. 7), no uncertainty attached.

Parameters

<i>p_const</i>	A reference to a ROBaseConstraint (p. 7) instance which specifies the primary variable constraint.
----------------	---

3.3.2.13 ROConstraint::ROConstraint (const ROConstraint & *p_const*)

The copy constructor.

Parameters

<i>p_const</i>	A reference to a ROConstraint (p. 10) instance.
----------------	--

3.3.3 Member Function Documentation

3.3.3.1 ROConstraint& ROConstraint::operator= (const ROConstraint & *p_const*)

The assignment operator "="

Parameters

<i>p_const</i>	A reference to a ROConstraint (p. 10) instance.
----------------	--

Returns

ROConstraint (p. 10)& A reference of the invocign object.

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

- include/ROConstraint.h

3.4 ROConstraintSet Class Reference

```
#include <ROConstraintSet.h>
```

Public Member Functions

- **ROConstraintSet** ()
- **ROConstraintSet** (const **ROConstraintSet** &p_constSet)
- **ROConstraintSet** (const **ROBaseConstraint** &p_baseConst)
- **ROConstraintSet** (const **ROUnLinConstraintArray** &p_unConstArray, const **ROUnSOCConstraintArray** &p_unNormConstArray)
- **ROConstraintSet** (const **ROUnLinConstraint** &p_unConst, const **ROUnSOCConstraintArray** &p_unNormConstArray)
- **ROConstraintSet** (const **ROUnLinConstraintArray** &p_unConstArray, const **ROUnSOCConstraint** &p_unNormConst)
- **ROConstraintSet** (const **ROUnLinConstraint** &p_unConst, const **ROUnSOCConstraint** &p_unNormConst)
- **ROConstraintSet** (const **ROUnLinConstraintArray** &p_unConstArray)
- **ROConstraintSet** (const **ROUnLinConstraint** &p_unConst)
- **ROConstraintSet** (const **ROUnSOCConstraintArray** &p_unNormConstArray)
- **ROConstraintSet** (const **ROUnSOCConstraint** &p_unNormConst)
- **ROConstraintSet** & operator= (const **ROConstraintSet** &p_const)
- void **set** (const **ROBaseConstraint** &p_baseConst)
- void **add** (const **ROBaseConstraint** &p_baseConst)
- void **add** (const **ROUnLinConstraint** &p_unConst)
- void **add** (const **ROUnSOCConstraint** &p_socConst)
- void **add** (const **ROConstraintSet** &p_constSet)

3.4.1 Detailed Description

An instance of this class is a set of constraints in a model.

Date: 17-Oct-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 19 of file ROConstraintSet.h.

3.4.2 Constructor & Destructor Documentation

3.4.2.1 ROConstraintSet::ROConstraintSet ()

The default constructor which creates an empty object. It must be initialized before use.

3.4.2.2 ROConstraintSet::ROConstraintSet (const ROConstraintSet & *p_constSet*)

This is the copy constructor.

Parameters

<i>p_constSet</i>	A reference to a ROConstraintSet (p. 14) instance.
-------------------	---

3.4.2.3 ROConstraintSet::ROConstraintSet (const ROBaseConstraint & *p_baseConst*)

This constructor creates a **ROConstraintSet** (p. 14) from a single base constraint.

Parameters

<i>p_baseConst</i>	A reference to a ROBaseConstraint (p. 7) instance.
--------------------	---

3.4.2.4 ROConstraintSet::ROConstraintSet (const ROUnLinConstraintArray & *p_unConstArray*, const ROUnSOCConstraintArray & *p_unNormConstArray*)

This constructor creates a **ROConstraintSet** (p. 14) from ROUnLinConstraintArray and ROUnSOCConstraintArray.

Parameters

<i>p_unConstArray</i>	A reference to a ROUnLinConstraintArray instance.
<i>p_unNormConstArray</i>	A reference to a ROUnSOCConstraintArray instance.

3.4.2.5 ROConstraintSet::ROConstraintSet (const ROUnLinConstraint & *p_unConst*, const ROUnSOCConstraintArray & *p_unNormConstArray*)

This constructor creates a **ROConstraintSet** (p. 14) from **ROUnLinConstraint** (p. 39) and ROUnSOCConstraintArray.

Parameters

<i>p_unConst</i>	A reference to a ROUnLinConstraint (p. 39) instance.
<i>p_unNormConstArray</i>	A reference to a ROUnSOCConstraintArray instance.

3.4.2.6 ROConstraintSet::ROConstraintSet (const ROUnLinConstraintArray & *p_unConstArray*, const ROUnSOCConstraint & *p_unNormConst*)

This constructor creates a **ROConstraintSet** (p. 14) from ROUnLinConstraintArray and **ROUnSOCConstraint** (p. 41).

Parameters

<i>p_unConstArray</i>	A reference to a ROUnLinConstraintArray instance.
<i>p_unNormConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance.

3.4.2.7 **ROConstraintSet::ROConstraintSet** (const **ROUnLinConstraint** & *p_unConst*, const **ROUnSOCConstraint** & *p_unNormConst*)

This constructor creates a **ROConstraintSet** (p. 14) from **ROUnLinConstraint** (p. 39) and **ROUnSOCConstraint** (p. 41).

Parameters

<i>p_unConst</i>	A reference to a <i>p_unConst</i> instance.
<i>p_unNormConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance.

3.4.2.8 **ROConstraintSet::ROConstraintSet** (const **ROUnLinConstraintArray** & *p_unConstArray*)

This constructor creates a **ROConstraintSet** (p. 14) from **ROUnLinConstraintArray**.

Parameters

<i>p_unConstArray</i>	A reference to a ROUnLinConstraintArray instance.
-----------------------	--

3.4.2.9 **ROConstraintSet::ROConstraintSet** (const **ROUnLinConstraint** & *p_unConst*)

This constructor creates an **ROConstraintSet** (p. 14) from **ROUnLinConstraint** (p. 39).

Parameters

<i>p_unConst</i>	A reference to a ROUnLinConstraint (p. 39) instance.
------------------	---

3.4.2.10 **ROConstraintSet::ROConstraintSet** (const **ROUnSOCConstraintArray** & *p_unNormConstArray*)

This constructor creates an **ROConstraintSet** (p. 14) from **ROUnSOCConstraintArray**.

Parameters

<i>p_unNormConstArray</i>	A reference to a ROUnSOCConstraintArray instance.
---------------------------	--

3.4.2.11 **ROConstraintSet::ROConstraintSet** (const **ROUnSOCConstraint** & *p_unNormConst*)

This constructor creates an **ROConstraintSet** (p. 14) from **ROUnSOCConstraint** (p. 41).

Parameters

<i>p_unNormConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance.
----------------------	---

3.4.3 Member Function Documentation

3.4.3.1 void ROConstraintSet::add (const ROBaseConstraint & *p_baseConst*)

This member function add the elementary base constraint.

Parameters

<i>p_baseConst</i>	A reference to a ROBaseConstraint (p. 7) instance.
--------------------	---

Returns

NULL.

3.4.3.2 void ROConstraintSet::add (const ROUnLinConstraint & *p_unConst*)

This member function add the base uncertain constraint.

Parameters

<i>p_unConst</i>	A reference to a ROUnLinConstraint (p. 39) instance.
------------------	---

Returns

NULL.

3.4.3.3 void ROConstraintSet::add (const ROUnSOCConstraint & *p_socConst*)

This member function add the SOC uncertain constraint.

Parameters

<i>p_socConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance.
-------------------	---

Returns

NULL.

3.4.3.4 void ROConstraintSet::add (const ROConstraintSet & *p_constSet*)

This member function add a set of constraints.

Parameters

<i>p_constSet</i>	A reference to a ROConstraintSet (p. 14) instance.
-------------------	---

Returns

NULL.

3.4.3.5 ROConstraintSet& ROConstraintSet::operator= (const ROConstraintSet & *p_const*)

The assignment operator "=".

Parameters

<i>p_const</i>	A reference to a ROConstraintSet (p. 14) instance.
----------------	---

Returns

ROConstraintSet (p. 14)& A reference of current invoking object.

3.4.3.6 void ROConstraintSet::set (const ROBaseConstraint & p_baseConst)

This member function add the primary base constraint.

Parameters

<i>p_baseConst</i>	A reference to a ROBaseConstraint (p. 7) instance.
--------------------	---

Returns

NULL.

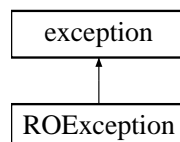
The documentation for this class was generated from the following file:

- include/ROConstraintSet.h

3.5 ROException Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROException:



3.5.1 Detailed Description

This class is the base class for exceptions in robust optimization.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 24 of file roexceptions.h.

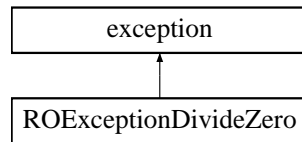
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.6 ROExceptionDivideZero Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionDivideZero:



3.6.1 Detailed Description

An instance of this class is an exception thrown when expressions are divided by zero.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 86 of file roexceptions.h.

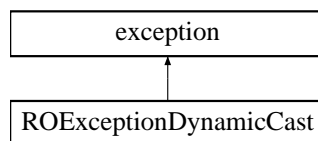
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.7 ROExceptionDynamicCast Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionDynamicCast:



Public Member Functions

- **ROExceptionDynamicCast** (const char *log=NULL)
- virtual const char * **what** () const throw ()

3.7.1 Detailed Description

An instance of this class is an exception thrown when dynamic cast impossible.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 148 of file roexceptions.h.

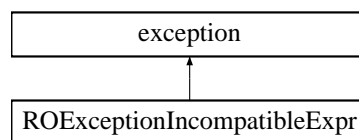
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.8 ROExceptionIncompatibleExpr Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionIncompatibleExpr:

**Public Member Functions**

- **ROExceptionIncompatibleExpr** (const char *log=NULL)
- virtual const char * **what** () const throw ()

3.8.1 Detailed Description

An instance of this class is an exception thrown when types are incompatible.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 56 of file roexceptions.h.

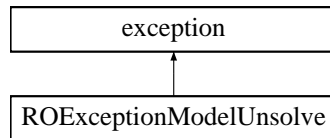
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.9 ROExceptionModelUnsolve Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionModelUnsolve:



Public Member Functions

- **ROExceptionModelUnsolve** (const char *log=NULL)
- virtual const char * **what** () const throw ()

3.9.1 Detailed Description

An instance of this class is an exception thrown when model is used before solve.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 178 of file roexceptions.h.

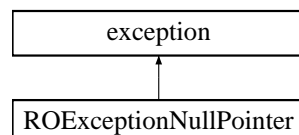
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.10 ROExceptionNullPointer Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionNullPointer:



Public Member Functions

- **ROExceptionNullPointer** (const char *log=NULL)
- virtual const char * **what** () const throw ()

3.10.1 Detailed Description

An instance of this class is an exception thrown when null pointer encountered.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 118 of file roexceptions.h.

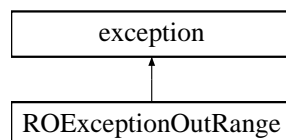
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.11 ROExceptionOutOfRange Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionOutOfRange:



3.11.1 Detailed Description

An instance of this class is an exception thrown when index out of range.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 102 of file roexceptions.h.

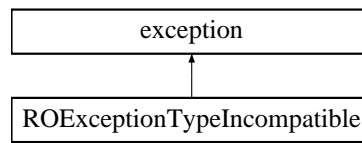
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.12 ROExceptionTypeIncompatible Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionTypeIncompatible:



3.12.1 Detailed Description

An instance of this class is an exception thrown when types are incompatible.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 40 of file roexceptions.h.

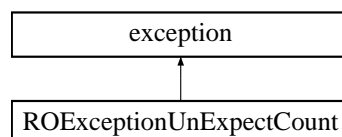
The documentation for this class was generated from the following file:

- include/roexceptions.h

3.13 ROExceptionUnExpectCount Class Reference

```
#include <roexceptions.h>
```

Inheritance diagram for ROExceptionUnExpectCount:



Public Member Functions

- **ROExceptionUnExpectCount** (const char *log=NULL)
- virtual const char * **what** () const throw ()

3.13.1 Detailed Description

An instance of this class is an exception thrown when unexpected var or unvar count met.

Date: 05-Apr-2013

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 208 of file roexceptions.h.

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

- include/roexceptions.h

3.14 ROExpr Class Reference

```
#include <ROExpr.h>
```

Public Member Functions

- **ROExpr** ()
- **ROExpr** (const **ROExpr** &p_expr)
- **ROExpr** (const **ROVar** &p_var)
- **ROExpr** (const **ROIntVar** &p_var)
- **ROExpr** (const **ROBinVar** &p_var)
- **ROExpr** (const **ROUn** &p_un)
- **ROExpr** (RONum val)
- **ROExpr** & **operator=** (RONum p_num)
- **ROExpr** & **operator=** (const **ROVar** &p_var)
- **ROExpr** & **operator=** (const **ROUn** &p_un)
- **ROExpr** & **operator=** (const **ROExpr** &p_expr)
- **ROExpr** & **operator=** (const **ROVarDR** &p_varDR)
- **ROExpr** & **operator+=** (RONum p_val)
- **ROExpr** & **operator-=** (RONum p_val)
- **ROExpr** & **operator*=** (RONum p_val)
- **ROExpr** & **operator/=** (RONum p_val)
- **ROExpr** & **operator+=** (const **ROVar** &p_var)
- **ROExpr** & **operator-=** (const **ROVar** &p_var)
- **ROExpr** & **operator+=** (const **ROUn** &p_un)
- **ROExpr** & **operator-=** (const **ROUn** &p_un)
- **ROExpr** & **operator+=** (const **ROExpr** &p_expr)
- **ROExpr** & **operator+=** (const **ROVarDR** &p_varDR)
- **ROExpr** & **operator-=** (const **ROExpr** &p_expr)
- **ROExpr** & **operator-=** (const **ROVarDR** &p_varDR)
- **ROExpr** & **operator*=** (const **ROVar** &p_var)
- **ROExpr** & **operator*=** (const **ROUn** &p_un)
- **ROExpr** & **operator*=** (const **ROExpr** &p_expr)
- **ROExpr** & **expect** (const **ROUn** &p_un)
- **ROExpr** & **expect** (const **ROExpr** &p_expr)
- **~ROExpr** ()

3.14.1 Detailed Description

An instance of this class represents an expression in a model.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 21 of file ROExpr.h.

3.14.2 Constructor & Destructor Documentation

3.14.2.1 ROExpr::ROExpr ()

This constructor creates an empty expression.

3.14.2.2 ROExpr::ROExpr (const ROExpr & *p_expr*)

The copy constructor of **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

3.14.2.3 ROExpr::ROExpr (const ROVar & *p_var*)

The copy constructor of **ROExpr** (p. 24).

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

3.14.2.4 ROExpr::ROExpr (const ROIntVar & *p_var*)

The copy constructor of **ROExpr** (p. 24).

Parameters

<i>p_var</i>	A reference to a ROIntVar (p. 31) instance.
--------------	--

3.14.2.5 ROExpr::ROExpr (const ROBinVar & *p_var*)

The copy constructor of **ROExpr** (p. 24).

Parameters

<i>p_var</i>	A reference to a ROBinVar (p. 9) instance.
--------------	---

3.14.2.6 ROExpr::ROExpr (const ROUn & *p_un*)

The copy constructor of **ROExpr** (p. 24).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

3.14.2.7 **ROExpr::ROExpr** (**RONum** *val*)

The copy constructor of **ROExpr** (p. 24).

Parameters

<i>val</i>	A RONum type number.
------------	-----------------------------

3.14.2.8 **ROExpr::~~ROExpr** ()

Destructor of **ROExpr** (p. 24).

3.14.3 Member Function Documentation

3.14.3.1 **ROExpr& ROExpr::expect** (**const ROUn** & *p_un*)

This member function takes the expected form of given **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.2 **ROExpr& ROExpr::expect** (**const ROExpr** & *p_expr*)

This member function takes the expected form of given **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.3 **ROExpr& ROExpr::operator*=**(**RONum** *p_val*)

This is the overloaded operator "*" between **ROExpr** (p. 24) and **RONum**.

Parameters

<i>p_val</i>	An object of RONum .
--------------	-----------------------------

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.4 **ROExpr**& **ROExpr::operator*=** (**const ROVar** & *p_var*)

This is the overloaded operator "*" between **ROExpr** (p. 24) and **ROVar** (p. 48).

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.5 **ROExpr**& **ROExpr::operator*=** (**const ROUn** & *p_un*)

This is the overloaded operator "*" between **ROExpr** (p. 24) and **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.6 **ROExpr**& **ROExpr::operator*=** (**const ROExpr** & *p_expr*)

This is the overloaded operator "*" between **ROExpr** (p. 24) and **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.7 **ROExpr**& **ROExpr::operator+=** (**RONum** *p_val*)

This is the overloaded operator "+" between **ROExpr** (p. 24) and **RONum**.

Parameters

<i>p_val</i>	An Object of RONum .
--------------	-----------------------------

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.8 ROExpr& ROExpr::operator+= (const ROVar & p_var)

This is the overloaded operator "+=" between **ROExpr** (p. 24) and **ROVar** (p. 48).

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.9 ROExpr& ROExpr::operator+= (const ROUn & p_un)

This is the overloaded operator "+=" between **ROExpr** (p. 24) and **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.10 ROExpr& ROExpr::operator+= (const ROExpr & p_expr)

This is the overloaded operator "+=" between **ROExpr** (p. 24) and **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.11 ROExpr& ROExpr::operator+= (const ROVarDR & p_varDR)

This is the overloaded operator "+=" between **ROExpr** (p. 24) and **ROVarDR** (p. 50).

Parameters

<i>p_varDR</i>	A reference to a ROVarDR (p. 50) instance.
----------------	---

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.12 ROExpr& ROExpr::operator-= (RNum p_val)

This is the overloaded operator "-=" between **ROExpr** (p. 24) and **RNum**.

Parameters

<i>p_val</i>	An object of RNum.
--------------	--------------------

Returns

ROExpr (p. 24) & A reference of current invoking instance.

3.14.3.13 **ROExpr**& **ROExpr::operator-=** (const **ROVar** & *p_var*)

This is the overloaded operator "-" between **ROExpr** (p. 24) and **ROVar** (p. 48).

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

ROExpr (p. 24) & A reference of current invoking instance.

3.14.3.14 **ROExpr**& **ROExpr::operator-=** (const **ROUn** & *p_un*)

This is the overloaded operator "-" between **ROExpr** (p. 24) and **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROExpr (p. 24) & A reference of current invoking instance.

3.14.3.15 **ROExpr**& **ROExpr::operator-=** (const **ROExpr** & *p_expr*)

This is the overloaded operator "-" between **ROExpr** (p. 24) and **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROExpr (p. 24) & A reference of current invoking instance.

3.14.3.16 **ROExpr**& **ROExpr::operator-=** (const **ROVarDR** & *p_varDR*)

This is the overloaded operator "-" between **ROExpr** (p. 24) and **ROVarDR** (p. 50).

Parameters

<i>p_varDR</i>	A reference to a ROVarDR (p. 50) instance.
----------------	---

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.17 ROExpr& ROExpr::operator/= (RNum *p_val*)

This is the overloaded operator "/=" between **ROExpr** (p. 24) and RNum.

Parameters

<i>p_val</i>	An object of RNum.
--------------	--------------------

Returns

ROExpr (p. 24)& A reference of current invoking instance.

3.14.3.18 ROExpr& ROExpr::operator= (RNum *p_num*)

This is the assignment "=" operator.

Parameters

<i>p_num</i>	A RNum type number.
--------------	---------------------

Returns

ROExpr (p. 24)& A reference to current invoking instance.

3.14.3.19 ROExpr& ROExpr::operator= (const ROVar & *p_var*)

This is the assignment "=" operator.

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

ROExpr (p. 24)& A reference to current invoking instance.

3.14.3.20 ROExpr& ROExpr::operator= (const ROUn & *p_un*)

This is the assignment "=" operator.

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROExpr (p. 24)& A reference to current invoking instance.

3.14.3.21 ROExpr& ROExpr::operator= (const ROExpr & p_expr)

This is the assignment "=" operator.

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance which would be copied by invoking instance.
---------------	---

Returns

ROExpr (p. 24)& A reference to current invoking instance.

3.14.3.22 ROExpr& ROExpr::operator= (const ROVarDR & p_varDR)

This is the assignment "=" operator. Caution : might lose underlying constraints attached with **ROVarDR** (p. 50).

Parameters

<i>p_varDR</i>	A reference to a ROVarDR (p. 50) instance which would be copied by invoking instance.
----------------	--

Returns

ROExpr (p. 24)& A reference to current invoking instance.

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

- include/ROExpr.h

3.15 ROIntVar Class Reference

```
#include <ROIntVar.h>
```

Public Member Functions

- **ROIntVar** (const double p_lb=RONInfinity, const double p_ub=RONInfinity, const std::string &p_name="")
- **ROIntVar** (const **ROIntVar** &p_intVar)
- const **ROVar** & **getVar** () const

3.15.1 Detailed Description

An instance of this class is an integer variable without uncertainty.

Date: 31-May-2013

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 22 of file ROIntVar.h.

3.15.2 Constructor & Destructor Documentation

3.15.2.1 `ROIntVar::ROIntVar (const double p_lb = RONInfinity, const double p_ub = ROInfinity, const std::string & p_name = " ")`

This constructor creates an **ROIntVar** (p. 31) instance from given lower bound, upper bound, and name.

Parameters

<i>p_lb</i>	lower bound of invoking object.
<i>p_ub</i>	upper bound of invoking object.
<i>p_name</i>	name of invoking object.

3.15.2.2 `ROIntVar::ROIntVar (const ROIntVar & p_intVar)`

The copy constructor.

Parameters

<i>p_intVar</i>	A reference to a ROIntVar (p. 31) instance.
-----------------	--

3.15.3 Member Function Documentation

3.15.3.1 `const ROVar& ROIntVar::getVar () const`

Get the underlying variable reference.

Returns

ROVar (p. 48)& A reference of current invoking instance.

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

- include/ROIntVar.h

3.16 ROModel Class Reference

```
#include <ROModel.h>
```

Public Member Functions

- **ROModel** ()
- void **add** (const **ROBaseConstraint** &*p_const*)
- void **add** (ROBaseConstraintArray &*p_constArray*)
- void **add** (ROConstraintArray &*p_constArray*)
- void **add** (const **ROConstraint** &*p_const*)
- void **add** (const **ROConstraintSet** &*p_constSet*)
- void **add** (const **ROObjective** &*p_obj*)
- void **solve** ()
- RNum **getObjValue** ()
- std::string **getStatus** ()
- RNum **getVarValue** (const **ROVar** &*p_var*)
- void **exportModel** ()
- void **exportModel** (std::string &*p_dirPath*)
- ~**ROModel** ()

3.16.1 Detailed Description

An instance of this class represents a model. A model consists of constraints, constrained variables, objectives, and possibly other modeling objects, that represents a robust optimization problem.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 20 of file ROModel.h.

3.16.2 Constructor & Destructor Documentation

3.16.2.1 ROModel::ROModel ()

The default constructor which creates an empty model. It must be initialized before use.

3.16.2.2 ROModel::~~ROModel ()

Destructor of **ROExpr** (p. 24).

3.16.3 Member Function Documentation

3.16.3.1 void ROModel::add (const ROBaseConstraint & *p_const*)

This member function adds "ROBaseConstraintArray" object (array of base constraints) to the invoking model.

Parameters

<i>p_const</i>	A reference to a "ROBaseConstraint" instance.
----------------	---

Returns

NULL.

3.16.3.2 void ROModel::add (ROBaseConstraintArray & *p_constArray*)

This member function adds "ROBaseConstraintArray" object (array of base constraints) to the invoking model.

Parameters

<i>p_constArray</i>	A reference to a "ROBaseConstraint" instance.
---------------------	---

Returns

NULL.

3.16.3.3 void ROModel::add (ROConstraintArray & *p_constArray*)

This member function adds "ROConstraintArray" object (array of ro-constraints) to the invoking model.

Parameters

<i>p_constArray</i>	A reference to a "ROConstraintArray" instance.
---------------------	--

Returns

NULL.

3.16.3.4 void ROModel::add (const ROConstraint & *p_const*)

This member function adds "ROConstraint" object to the invoking model.

Parameters

<i>p_const</i>	A reference to a "ROConstraint" instance.
----------------	---

Returns

NULL.

3.16.3.5 void ROModel::add (const ROConstraintSet & *p_constSet*)

This member function adds "ROConstraintSet" object to the invoking model.

Parameters

<i>p_constSet</i>	A reference to a "ROConstraintSet" instance.
-------------------	--

Returns

NULL.

3.16.3.6 void ROModel::add (const ROObjective & *p_obj*)

This member function adds "ROObjective" object to the invoking model.

Parameters

<i>p_obj</i>	A reference to a "ROObjective" instance.
--------------	--

Returns

NULL.

3.16.3.7 void ROModel::exportModel ()

This member function would print the LP formulation of invoking model in default path.

Returns

NULL.

3.16.3.8 void ROModel::exportModel (std::string & p_dirPath)

This member function would print the LP formulation of invoking model in specified directory.

Parameters

<i>std::string</i>	Specified directory path.
--------------------	---------------------------

Returns

NULL.

3.16.3.9 RNum ROModel::getObjValue ()

This member function returns the objective value of invoking model if it finds a solution.

Returns

Objective value.

3.16.3.10 std::string ROModel::getStatus ()

This member function returns the status of the model after call **solve()** (p. 35).

See Also

solve() (p. 35).

Returns

status after model solving.

3.16.3.11 RNum ROModel::getVarValue (const ROVar & p_var)

This member function returns the solution of a specified variable.

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

RNum The solved solution of specified variable.

3.16.3.12 void ROModel::solve ()

This member function solves the current model.

Returns

NULL.

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

- include/ROModel.h

3.17 ROObjective Class Reference

```
#include <ROObjective.h>
```

Public Member Functions

- **ROObjective** (const **ROUnVarC** &p_unVarC, const **ROConstraintSet** &, const ROObjectiveTag p_tag=EnumMin)
- **ROObjective** (const **ROUnVarC** &p_unVarC, const ROObjectiveTag p_tag=EnumMin)
- **ROObjective** (const **ROObjective** &p_obj)
- **ROObjective & operator=** (const **ROObjective** &p_obj)
- ROObjectiveTag **getTag** () const

3.17.1 Detailed Description

An instance of this class is an objective in a model. An objective consists of its obj tag and an expression.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 23 of file ROObjective.h.

3.17.2 Constructor & Destructor Documentation

3.17.2.1 **ROObjective::ROObjective** (const **ROUnVarC** & *p_unVarC*, const **ROConstraintSet** &, const ROObjectiveTag *p_tag* = EnumMin)

This constructor creates an **ROObjective** (p. 36) object from given **ROUnVarC** (p. 42), uncertainty set and ROObjectiveTag.

Parameters

<i>p_unVarC</i>	A reference to a ROUnVarC (p. 42) instance.
<i>p_constSet</i>	A reference to a ROConstraintSet (p. 14) instance which is the uncertainty set.
<i>p_tag</i>	The ROObjective (p. 36) tag. (Minimize or Maximize).

3.17.2.2 ROObjective::ROObjective (const ROUnVarC & p_unVarC, const ROObjectiveTag p_tag = EnumMin)

This constructor creates an **ROObjective** (p. 36) object from given **ROUnVarC** (p. 42) and ROObjectiveTag.

Parameters

<i>p_unVarC</i>	A reference to a ROUnVarC (p. 42) instance.
<i>p_tag</i>	The ROObjective (p. 36) tag. (Minimize or Maximize).

3.17.2.3 ROObjective::ROObjective (const ROObjective & p_obj)

This is the copy constructor of **ROObjective** (p. 36).

Parameters

<i>p_obj</i>	A reference to a ROObjective (p. 36) instance.
--------------	---

3.17.3 Member Function Documentation

3.17.3.1 ROObjectiveTag ROObjective::getTag () const

This member function returns the objective tag of invoking objective. (Minimize or Maximize)

Returns

ROObjectiveTag The **ROObjective** (p. 36) tag of invoking objective instance.

3.17.3.2 ROObjective& ROObjective::operator= (const ROObjective & p_obj)

This is the assignment operator "=".

Parameters

<i>p_obj</i>	A reference to a ROObjective (p. 36) instance.
--------------	---

Returns

ROObjective (p. 36)& The reference of current invoking object.

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

- include/ROObjective.h

3.18 ROUn Class Reference

```
#include <ROUn.h>
```

Public Member Functions

- **ROUn** ()
- **ROUn** (const **ROUn** &p_un)
- int **getId** () const
- **~ROUn** ()

Static Public Member Functions

- static int **getNo** ()

Friends

- class **ROModel**

3.18.1 Detailed Description

An instance of this class is an uncertain variable.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 24 of file ROUn.h.

3.18.2 Constructor & Destructor Documentation

3.18.2.1 ROUn::ROUn ()

The default constructor of **ROUn** (p. 37).

3.18.2.2 ROUn::ROUn (const ROUn & p_un)

The copy constructor of **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

3.18.2.3 ROUn::~~ROUn ()

The destructor of **ROUn** (p. 37).

3.18.3 Member Function Documentation

3.18.3.1 int ROUn::getId () const

This member function returns the unique ID of invoking uncertainty variable.

Returns

int Unique ID of invoking object.

3.18.3.2 static int ROUn::getNo () [static]

Return the number of uncertain variables created.

Returns

int Number of uncertain variables generated.

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

- include/ROUn.h

3.19 ROUnLinConstraint Class Reference

```
#include <ROUnLinConstraint.h>
```

Public Member Functions

- **ROUnLinConstraint** (const **ROUnLinConstraint** &p_unConst)
- **ROUnLinConstraint** (const **ROBaseConstraint** &p_baseConst)
- **ROUnLinConstraint** (const ROConstTag p_tag=EnumLeq)
- **ROUnLinConstraint** & **operator=** (const **ROUnLinConstraint** &p_expr)
- **ROUnLinConstraint** & **operator=** (const **ROBaseConstraint** &p_expr)
- ROConstTag **getTag** () const

3.19.1 Detailed Description

An instance of this class is an uncertain linear constraint in a model. A base constraint consists of its equation tag and an expression (LHS).

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 26 of file ROUnLinConstraint.h.

3.19.2 Constructor & Destructor Documentation

3.19.2.1 ROUnLinConstraint::ROUnLinConstraint (const ROUnLinConstraint & p_unConst)

This is the copy constructor.

Parameters

<i>p_unConst</i>	A reference to a ROUnLinConstraint (p. 39) instance.
------------------	---

3.19.2.2 ROUnLinConstraint::ROUnLinConstraint (const ROBaseConstraint & *p_baseConst*)

This is the copy constructor.

Parameters

<i>p_baseConst</i>	A reference to a ROBaseConstraint (p. 7) instance.
--------------------	---

3.19.2.3 ROUnLinConstraint::ROUnLinConstraint (const ROConstTag *p_tag* = EnumLeq)

The default constructor of **ROUnLinConstraint** (p. 39).

Parameters

<i>p_tag</i>	An object of ROConstTag, default is "<=".
--------------	---

3.19.3 Member Function Documentation

3.19.3.1 ROConstTag ROUnLinConstraint::getTag () const

This member function returns the equation tag ("<=" or "==") of invoking object.

Returns

ROConstTag The equation tag ("<=" or "==") of invoking object.

3.19.3.2 ROUnLinConstraint& ROUnLinConstraint::operator= (const ROUnLinConstraint & *p_expr*)

The assignment operator "=".

Parameters

<i>p_expr</i>	A reference to a ROUnLinConstraint (p. 39) instance.
---------------	---

Returns

ROUnLinConstraint (p. 39)& A reference of current invoking object.

3.19.3.3 ROUnLinConstraint& ROUnLinConstraint::operator= (const ROBaseConstraint & *p_expr*)

The assignment operator "=".

Parameters

<i>p_expr</i>	A reference to a ROBaseConstraint (p. 7) instance.
---------------	---

Returns

ROUnLinConstraint (p. 39)& A reference of current invoking object.

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

- include/ROUnLinConstraint.h

3.20 ROUnSOCConstraint Class Reference

```
#include <ROUnSOCConstraint.h>
```

Public Member Functions

- **ROUnSOCConstraint** ()
- **ROUnSOCConstraint** (const **ROUnSOCConstraint** &p_unSOC)
- **ROUnSOCConstraint** (ROExprArray &p_exprArray, const **ROExpr** &p_expr)
- **ROUnSOCConstraint** (ROUnArray &p_unArray, const **ROExpr** &p_expr)
- **ROUnSOCConstraint** (const **ROExpr** &p_expr, ROExprArray &p_exprArray)
- **ROUnSOCConstraint** (const **ROExpr** &p_exprL, const **ROExpr** &p_exprR)
- **ROUnSOCConstraint** & operator= (const **ROUnSOCConstraint** &p_normConst)

3.20.1 Detailed Description

An instance of this class is a norm uncertainty constraint in a **ROConstraint** (p. 10).

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 22 of file ROUnSOCConstraint.h.

3.20.2 Constructor & Destructor Documentation

3.20.2.1 ROUnSOCConstraint::ROUnSOCConstraint ()

This constructor creates an empty norm uncertainty constraint. It must be initialized before use.

3.20.2.2 ROUnSOCConstraint::ROUnSOCConstraint (const ROUnSOCConstraint & p_unSOC)

This copy constructor, which creates from another **ROUnSOCConstraint** (p. 41) object.

Parameters

<i>p_unSOC</i>	A reference to a ROUnSOCConstraint (p. 41) instance.
----------------	---

3.20.2.3 ROUnSOCConstraint::ROUnSOCConstraint (ROExprArray & p_exprArray, const ROExpr & p_expr)

This constructor creates a SOC uncertainty constraint from an object of **ROExpr** (p. 24) and an object of ROExprArray. the default equation tag is "<="

Parameters

<i>p_exprArray</i>	A reference to a ROExprArray instance.
<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.

3.20.2.4 ROUnSOCConstraint::ROUnSOCConstraint (ROUnArray & p_unArray, const ROExpr & p_expr)

This constructor creates a SOC uncertainty constraint from an object of **ROExpr** (p. 24) and an object of ROExprArray. the default equation tag is "<="

Parameters

<i>p_unArray</i>	A reference to a ROUnArray instance.
<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.

3.20.2.5 ROUnSOCConstraint::ROUnSOCConstraint (const ROExpr & p_expr, ROExprArray & p_exprArray)

This constructor creates a SOC uncertainty constraint from an object of **ROExpr** (p. 24) and an object of ROExprArray. the default equation tag is ">="

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
<i>p_exprArray</i>	A reference to a ROExprArray instance.

3.20.2.6 ROUnSOCConstraint::ROUnSOCConstraint (const ROExpr & p_exprL, const ROExpr & p_exprR)

This constructor creates a SOC uncertainty constraint from two instances of **ROExpr** (p. 24). the default equation tag is "<="

Parameters

<i>p_exprL</i>	A reference to a ROExpr (p. 24) instance.
<i>p_exprR</i>	A reference to a ROExpr (p. 24) instance.

3.20.3 Member Function Documentation

3.20.3.1 ROUnSOCConstraint& ROUnSOCConstraint::operator= (const ROUnSOCConstraint & p_normConst)

This is the assignment operator "=".

Parameters

<i>p_normConst</i>	A reference to a ROUnSOCConstraint (p. 41) instance.
--------------------	---

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

- include/ROUnSOCConstraint.h

3.21 ROUnVarC Class Reference

```
#include <ROUnVarC.h>
```

Public Member Functions

- **ROUnVarC** ()
- **ROUnVarC** (const **ROUnVarC** &p_unC)
- **ROUnVarC** & **operator=** (const **ROUnVarC** &p_unC)

- **ROUnVarC** & **operator=** (const **ROExpr** &p_expr)
- **ROUnVarC** & **operator+=** (RNum p_val)
- **ROUnVarC** & **operator-=** (RNum p_val)
- **ROUnVarC** & **operator*=** (RNum p_val)
- **ROUnVarC** & **operator/=** (RNum p_val)
- **ROUnVarC** & **operator+=** (const **ROVar** &p_var)
- **ROUnVarC** & **operator-=** (const **ROVar** &p_var)
- **ROUnVarC** & **operator+=** (const **ROUn** &p_un)
- **ROUnVarC** & **operator-=** (const **ROUn** &p_un)
- **ROUnVarC** & **operator+=** (const **ROExpr** &p_expr)
- **ROUnVarC** & **operator+=** (const **ROUnVarC** &p_varC)
- **ROUnVarC** & **operator-=** (const **ROExpr** &p_expr)
- **ROUnVarC** & **operator-=** (const **ROUnVarC** &p_unC)
- **ROUnVarC** & **operator*=** (const **ROVar** &p_var)
- **ROUnVarC** & **operator*=** (const **ROUn** &p_un)
- **ROUnVarC** & **operator*=** (const **ROExpr** &p_expr)
- **ROUnVarC** & **expect** (const **ROUn** &p_un)
- **ROUnVarC** & **expect** (const **ROExpr** &p_expr)

3.21.1 Detailed Description

An instance of this class is a composite expression.

Date: 1-Nov-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 23 of file ROUnVarC.h.

3.21.2 Constructor & Destructor Documentation

3.21.2.1 **ROUnVarC::ROUnVarC** ()

This constructor creates an empty expression of **ROUnVarC** (p. 42).

3.21.2.2 **ROUnVarC::ROUnVarC** (const **ROUnVarC** & *p_unC*)

The copy constructor of **ROUnVarC** (p. 42).

Parameters

<i>p_unC</i>	A reference to a ROUnVarC (p. 42) instance.
--------------	--

3.21.3 Member Function Documentation

3.21.3.1 **ROUnVarC& ROUnVarC::expect** (const **ROUn** & *p_un*)

This member function takes the expected form of given **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.2 **ROUnVarC** & **ROUnVarC::expect** (*const ROExpr* & *p_expr*)

This member function takes the expected form of given **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.3 **ROUnVarC** & **ROUnVarC::operator*=(** *RONum p_val* **)**

This is the overloaded operator "*" between **ROUnVarC** (p. 42) and **RONum**.

Parameters

<i>p_val</i>	An object of RONum .
--------------	-----------------------------

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.4 **ROUnVarC** & **ROUnVarC::operator*=(** *const ROVar* & *p_var* **)**

This is the overloaded operator "*" between **ROUnVarC** (p. 42) and **ROVar** (p. 48).

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.5 **ROUnVarC** & **ROUnVarC::operator*=(** *const ROUn* & *p_un* **)**

This is the overloaded operator "*" between **ROUnVarC** (p. 42) and **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROUnVarC (p. 42)& A reference of current invoking instance.

3.21.3.6 **ROUnVarC**& **ROUnVarC::operator*=(const ROExpr & p_expr)**

This is the overloaded operator "*" between **ROUnVarC** (p. 42) and **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROUnVarC (p. 42)& A reference of current invoking instance.

3.21.3.7 **ROUnVarC**& **ROUnVarC::operator+=(RNum p_val)**

This is the overloaded operator "+" between **ROUnVarC** (p. 42) and **RNum**.

Parameters

<i>p_val</i>	An Object of RNum .
--------------	----------------------------

Returns

ROUnVarC (p. 42)& A reference of current invoking instance.

3.21.3.8 **ROUnVarC**& **ROUnVarC::operator+=(const ROVar & p_var)**

This is the overloaded operator "+" between **ROUnVarC** (p. 42) and **ROVar** (p. 48).

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

ROUnVarC (p. 42)& A reference of current invoking instance.

3.21.3.9 **ROUnVarC**& **ROUnVarC::operator+=(const ROUn & p_un)**

This is the overloaded operator "+" between **ROUnVarC** (p. 42) and **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROUnVarC (p. 42)& A reference of current invoking instance.

3.21.3.10 **ROUnVarC**& **ROUnVarC**::operator+=(const **ROExpr** & *p_expr*)

This is the overloaded operator "+" between **ROUnVarC** (p. 42) and **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.11 **ROUnVarC**& **ROUnVarC**::operator+=(const **ROUnVarC** & *p_varC*)

This is the overloaded operator "+" between **ROUnVarC** (p. 42) and **ROUnVarC** (p. 42).

Parameters

<i>p_varC</i>	A reference to a ROUnVarC (p. 42) instance.
---------------	--

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.12 **ROUnVarC**& **ROUnVarC**::operator-= (**RONum** *p_val*)

This is the overloaded operator "-" between **ROUnVarC** (p. 42) and **RONum**.

Parameters

<i>p_val</i>	An object of RONum .
--------------	-----------------------------

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.13 **ROUnVarC**& **ROUnVarC**::operator-= (const **ROVar** & *p_var*)

This is the overloaded operator "-" between **ROUnVarC** (p. 42) and **ROVar** (p. 48).

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.14 **ROUnVarC**& **ROUnVarC**::operator-= (const **ROUn** & *p_un*)

This is the overloaded operator "-" between **ROUnVarC** (p. 42) and **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.15 **ROUnVarC& ROUnVarC::operator-= (const ROExpr & p_expr)**

This is the overloaded operator "-=" between **ROUnVarC** (p. 42) and **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.16 **ROUnVarC& ROUnVarC::operator-= (const ROUnVarC & p_unC)**

This is the overloaded operator "-=" between **ROUnVarC** (p. 42) and **ROUnVarC** (p. 42).

Parameters

<i>p_unC</i>	A reference to a ROUnVarC (p. 42) instance.
--------------	--

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.17 **ROUnVarC& ROUnVarC::operator/= (RNum p_val)**

This is the overloaded operator "/=" between **ROUnVarC** (p. 42) and **RNum**.

Parameters

<i>p_val</i>	An object of RNum .
--------------	----------------------------

Returns

ROUnVarC (p. 42) & A reference of current invoking instance.

3.21.3.18 **ROUnVarC& ROUnVarC::operator= (const ROUnVarC & p_unC)**

This is the assignment "=" operator.

Parameters

<i>p_unC</i>	A reference to a ROUnVarC (p. 42) instance.
--------------	--

Returns

ROUnVarC (p. 42) & A reference to current invoking instance.

3.21.3.19 ROUnVarC&ROUnVarC::operator= (const ROExpr & p_expr)

This is the assignment "=" operator.

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROUnVarC (p. 42) & A reference to current invoking instance.

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

- include/ROUnVarC.h

3.22 ROVar Class Reference

```
#include <ROVar.h>
```

Public Member Functions

- **ROVar** (const double p_lb=RONInfinity, const double p_ub=RONInfinity, const std::string &p_name="", const ROVarType p_type=EnumROVar)
- **ROVar** (const **ROVar** &p_var)
- **ROVar** (const **ROIntVar** &p_intVar)
- **ROVar** (const **ROBinVar** &p_binVar)
- int **getId** () const
- ~**ROVar** ()

Static Public Member Functions

- static int **getNo** ()

Friends

- class **ROModel**

3.22.1 Detailed Description

An instance of this class is an variable without uncertainty.

Date: 30-Sep-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 23 of file ROVar.h.

3.22.2 Constructor & Destructor Documentation

3.22.2.1 ROVar::ROVar (const double *p_lb* = RONInfinity, const double *p_ub* = ROInfinity, const std::string & *p_name* = " ", const ROVarType *p_type* = EnumROVar)

This constructor creates an **ROVar** (p. 48) instance from given lower bound, upper bound, and name.

Parameters

<i>p_lb</i>	lower bound of invoking object.
<i>p_ub</i>	upper bound of invoking object.
<i>p_name</i>	name of invoking object.
<i>p_type</i>	type of variable, e.g. int, bool, double

3.22.2.2 ROVar::ROVar (const ROVar & *p_var*)

The copy constructor.

Parameters

<i>p_var</i>	A reference to a ROVar (p. 48) instance.
--------------	---

3.22.2.3 ROVar::ROVar (const ROIntVar & *p_intVar*)

The copy constructor. Date: 2013-06-01

Parameters

<i>p_intVar</i>	A reference to a ROIntVar (p. 31) instance.
-----------------	--

3.22.2.4 ROVar::ROVar (const ROBinVar & *p_binVar*)

The copy constructor. Date: 2013-06-01

Parameters

<i>p_binVar</i>	A reference to a ROBinVar (p. 9) instance.
-----------------	---

3.22.2.5 ROVar::~ROVar ()

The destructor of **ROVar** (p. 48).

3.22.3 Member Function Documentation

3.22.3.1 int ROVar::getId () const

This member function returns the unique ID of invoking variable.

Returns

int Unique ID of invoking object.

3.22.3.2 static int ROVar::getNo () [static]

Return the number of variables created.

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

- include/ROVar.h

3.23 ROVarDR Class Reference

```
#include <ROVarDR.h>
```

Public Member Functions

- **ROVarDR** ()
- **ROExpr** & **getExpr** () const
- **ROVar** & **getVar** (**ROExpr** &p_expr) const
- **ROVar** & **getVar** (**ROUn** &p_un) const
- **ROVar** & **getVar** () const
- void **addDR** (**ROExpr** &p_expr)
- void **addDR** (**ROUn** &p_un)
- void **clone** (**ROVarDR** &p_varDR)

3.23.1 Detailed Description

An instance of this class is a decision rule variable.

Date: 1-Nov-2012

Author

Zhang Meilin, National University of Singapore

Version

1.0

Definition at line 24 of file ROVarDR.h.

3.23.2 Constructor & Destructor Documentation

3.23.2.1 ROVarDR::ROVarDR ()

This constructor creates a decision rule variable without uncertainty.

3.23.3 Member Function Documentation

3.23.3.1 void ROVarDR::addDR (ROExpr & p_expr)

This member function adds an uncertain "ROExpr" to the invoking object's decision rule.

Parameters

<i>p_expr</i>	A reference to ROExpr (p. 24).
---------------	---------------------------------------

Returns

NULL.

3.23.3.2 void ROVarDR::addDR (ROUn & p_un)

This member function adds an uncertain "ROUn" to the invoking object's decision rule.

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

NULL.

3.23.3.3 void ROVarDR::clone (ROVarDR & p_varDR)

This member function copy the decision rule from another **ROVarDR** (p. 50) variable.

Parameters

<i>p_varDR</i>	A reference to a ROVarDR (p. 50) instance.
----------------	---

Returns

NULL.

3.23.3.4 ROExpr& ROVarDR::getExpr () const

This member function returns the **ROExpr** (p. 24) of invoking object.

Returns

ROExpr (p. 24) & A reference to the underlying **ROExpr** (p. 24) instance.

3.23.3.5 ROVar& ROVarDR::getVar (ROExpr & p_expr) const

This member function returns the reference to the variable related to given uncertainty **ROExpr** (p. 24).

Parameters

<i>p_expr</i>	A reference to a ROExpr (p. 24) instance.
---------------	--

Returns

ROVar (p. 48)& A reference to **ROVar** (p. 48).

3.23.3.6 **ROVar**& **ROVarDR**::getVar (**ROUn** & *p_un*) const

This member function returns the reference to the variable related to given uncertainty **ROUn** (p. 37).

Parameters

<i>p_un</i>	A reference to a ROUn (p. 37) instance.
-------------	--

Returns

ROVar (p. 48)& A reference to **ROVar** (p. 48).

3.23.3.7 **ROVar**& **ROVarDR**::getVar () const

This member function returns the reference to the variable not related to uncertainty.

Returns

ROVar (p. 48)& A reference to **ROVar** (p. 48).

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

- include/ROVarDR.h

Index

- ~ROExpr
 - ROExpr, 26
- ~ROModel
 - ROModel, 33
- ~ROUn
 - ROUn, 38
- ~ROVar
 - ROVar, 49
- add
 - ROConstraintSet, 16, 17
 - ROModel, 33, 34
- addDR
 - ROVarDR, 51
- clone
 - ROVarDR, 51
- expect
 - ROExpr, 26
 - ROUnVarC, 43, 44
- exportModel
 - ROModel, 34, 35
- getExpr
 - ROVarDR, 51
- getId
 - ROUn, 38
 - ROVar, 49
- getNo
 - ROUn, 38
 - ROVar, 50
- getObjValue
 - ROModel, 35
- getStatus
 - ROModel, 35
- getTag
 - ROBaseConstraint, 8
 - ROObjective, 37
 - ROUnLinConstraint, 40
- getVar
 - ROBinVar, 9
 - ROIntVar, 32
 - ROVarDR, 51, 52
- getVarValue
 - ROModel, 35
- operator*=
 - ROExpr, 26, 27
 - ROUnVarC, 44, 45
- operator+=
 - ROExpr, 27, 28
 - ROUnVarC, 45, 46
- operator=
 - ROExpr, 28, 29
 - ROUnVarC, 46, 47
- operator/=
 - ROExpr, 30
 - ROUnVarC, 47
- operator=
 - ROBaseConstraint, 8
 - ROConstraint, 13
 - ROConstraintSet, 17
 - ROExpr, 30, 31
 - ROObjective, 37
 - ROUnLinConstraint, 40
 - ROUnSOCCConstraint, 42
 - ROUnVarC, 47, 48
- ROBaseConstraint, 7
 - getTag, 8
 - operator=, 8
 - ROBaseConstraint, 7, 8
 - ROBaseConstraint, 7, 8
- ROBinVar, 9
 - getVar, 9
 - ROBinVar, 9
 - ROBinVar, 9
- ROConstraint, 10
 - operator=, 13
 - ROConstraint, 10–13
 - ROConstraint, 10–13
- ROConstraintSet, 14
 - add, 16, 17
 - operator=, 17
 - ROConstraintSet, 15, 16
 - ROConstraintSet, 15, 16
 - set, 18
- ROException, 18
- ROExceptionDivideZero, 19
- ROExceptionDynamicCast, 19
- ROExceptionIncompatibleExpr, 20
- ROExceptionModelUnsolved, 21
- ROExceptionNullPointer, 21
- ROExceptionOutOfRange, 22
- ROExceptionTypeIncompatible, 22
- ROExceptionUnexpectedCount, 23
- ROExpr, 24
 - ~ROExpr, 26
 - expect, 26
 - operator*=, 26, 27

- operator+=, 27, 28
- operator-=, 28, 29
- operator/=: 30
- operator=, 30, 31
- ROExpr, 25, 26
- ROExpr, 25, 26
- ROIntVar, 31
 - getVar, 32
 - ROIntVar, 32
 - ROIntVar, 32
- ROModel, 32
 - ~ROModel, 33
 - add, 33, 34
 - exportModel, 34, 35
 - getObjValue, 35
 - getStatus, 35
 - getVarValue, 35
 - ROModel, 33
 - ROModel, 33
 - solve, 35
- ROObjective, 36
 - getTag, 37
 - operator=, 37
 - ROObjective, 36, 37
 - ROObjective, 36, 37
- ROUn, 37
 - ~ROUn, 38
 - getId, 38
 - getNo, 38
 - ROUn, 38
 - ROUn, 38
- ROUnLinConstraint, 39
 - getTag, 40
 - operator=, 40
 - ROUnLinConstraint, 39, 40
 - ROUnLinConstraint, 39, 40
- ROUnSOCCConstraint, 41
 - operator=, 42
 - ROUnSOCCConstraint, 41, 42
 - ROUnSOCCConstraint, 41, 42
- ROUnVarC, 42
 - expect, 43, 44
 - operator*=, 44, 45
 - operator+=, 45, 46
 - operator-=, 46, 47
 - operator/=: 47
 - operator=, 47, 48
 - ROUnVarC, 43
 - ROUnVarC, 43
- ROVar, 48
 - ~ROVar, 49
 - getId, 49
 - getNo, 50
 - ROVar, 49
 - ROVar, 49
- ROVarDR, 50
 - addDR, 51
 - clone, 51
 - getExpr, 51
 - getVar, 51, 52
 - ROVarDR, 50
 - ROVarDR, 50
- set
 - ROConstraintSet, 18
- solve
 - ROModel, 35