Package ‘rLindo’

July 15, 2018

Type Package

Title R Interface to LINDO API

Version 12.0.1

Date 2017-03-02

Author Zhe Liu

Maintainer Zhe Liu<zliu@lindo.com>

Depends R (>= 2.14.1)

Description An interface to LINDO API. Supports Linear, Integer, Quadratic, Conic, General Nonlinear, Global, and Stochastic Programming models. To download the trial version LINDO API, please visit www.lindo.com/rlindo.

SystemRequirements LINDO API 12.0

URL www.lindo.com/rlindo

License LGPL (>= 2.1)

Collate rLindo.R zzz.R rLindoParam.R

Repository CRAN

NeedsCompilation yes

R topics documented:

`rLindo` *R interface to LINDO API.*

Description

R interface to LINDO API functions. For more information, please refer to LINDO API User Manual.

Details

In R interface all function names use the convention of 'r' + LINDO API function name. E.g, function rLScreateEnv in R corresponds to LScreateEnv in LINDO API.

References

LINDO SYSTEMS home page at www.lindo.com

Examples

```
#solve an LP

#load the package
library(rLindo)

#create LINDO enviroment object
rEnv <- rLScreateEnv()

#create LINDO model object
rModel <- rLScreateModel(rEnv)

#load LP data
nVars = 4
nCons = 4
nDir = 1
dObjConst = 0.
```

```
adC = c(1., 1., 1., 1.)
adB = c( 20., 20., 40., 10.)
acConTypes = "EGEG"
nNZ = 9
anBegCol = c( 0 , 2 , 5 , 7 , 9)
adA = c( 3.0, 4.0, 6.0, 5.0, 7.0, 8.0, 1.0, 2.0, 9.0)
anRowX = c( 0 , 2 , 1 , 2 , 3 , 2 , 3 , 0 , 1 )
pdLower = c(2, 1, -1.0E+30, -1.0E+30)
pdUpper = c(5, 1.0E+30, 10, 1.0E+30)
rLSloadLPData(rModel , nCons, nVars, nDir, dObjConst, adC, adB, acConTypes,
              nNZ, anBegCol, NULL, adA, anRowX, pdLower, pdUpper)

#solve the model
rLSoptimize(rModel,0)

#get primal solution
rLSgetPrimalSolution(rModel)

#get dual solution
rLSgetDualSolution(rModel)

#retrieve information
rLSgetDInfo(rModel,LS_DINFO_POBJ)
rLSgetIInfo(rModel,LS_IINFO_MODEL_STATUS)

#get basis
rLSgetBasis(rModel)

#delete enviroment and model objects
#free memory
rLSdeleteModel(rModel)
rLSdeleteEnv(rEnv)
```

---

rLSaddChanceConstraint

Add a new chance-constraint to the SP model.

Description

R interface function for LINDO API function LSaddChanceConstraint. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddChanceConstraint(model,iSense,nCons,paiCons,dPrLevel,dObjWeight)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
iSense	The sense of the chance-constraint. Possible values are LS_CONTYPE_LE and LS_CONTYPE_GE.
nCons	Number of rows in this chance constraint.
paiCons	An integer array containing row indices in the chanceconstraint.

dPrLevel	Probability level of this chance constraint.
dObjWeight	The constraint's weight in the probabilistic objective relative to the orignal objective function.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSaddCones *Add cones to a given model.*

Description

R interface function for LINDO API function LSaddCones. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddCones(model,nCone,pszConeTypes,paszConenames = NULL,paiConebegcol,paiConecols)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nCone	Number of cones to append to the model.
pszConeTypes	A character array containing the type of each cone to be added to the model.
paszConenames	A string array containing the names of each new cone.
paiConebegcol	An integer vector containing the index of the first variable in each new cone.
paiConecols	An integer array containing the indices of the variables in the new cones.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadConeData

rLSaddConstraints *Add constraints to a given model.*

Description

R interface function for LINDO API function LSaddConstraints. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddConstraints(model,nNumaddcons,pszConTypes,paszConNames = NULL,
                  paiArows,padAcoef,paiAcols,padB)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nNumaddcons	Number of constraints to append.
pszConTypes	A character array containing the type of each constraint to be added to the model.
paszConNames	A string array containing the names of each new constraint.
paiArows	An integer array containing the index of the first nonzero element in each new constraint.
padAcoef	A double array containing the nonzero coefficients of the new constraints.
paiAcols	An integer array containing the column indices of the nonzeros in the new constraints.
padB	A double array containing the right-hand side coefficients for each new constraint.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadLPData rLSaddVariables

rLSaddDiscreteBlocks *Add a new discrete stochastic block to the SP model.*

Description

R interface function for LINDO API function LSaddDiscreteBlocks. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddDiscreteBlocks(model,iStage,nRealzBlock,padProb,
                     pakStart,paiRows,paiCols,paiStvs,padVals,nModifyRule)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
iStage	The stage of the stochastic block.
nRealzBlock	Number of discrete events in the block.
padProb	An double array containing the event probabilities.
pakStart	An integer array containing the starting positions of events in the sparse matrix or instruction list.
paiRows	An integer array containing row indices of stochastic parameters.
paiCols	An integer array containing column indices of stochastic parameters.
paiStvs	An integer array containing stochastic parameters in the instruction list.
padVals	A double array containing stochastic values associated with the stochastic parameters listed in paiStvs or (paiArows,paiAcols).
nModifyRule	A flag indicating whether stochastic parameters update the core model by adding or replacing.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSaddDiscreteIndep *Add a new discrete independent stochastic parameter to the SP model.*

Description

R interface function for LINDO API function `LSaddDiscreteIndep`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddDiscreteIndep(model,iRow,jCol,iStv,nRealizations,
                    padProbs,padVals,nModifyRule)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iRow`	Row index of the stochastic parameter.
`jCol`	Column index of the stochastic parameter.
`iStv`	Index of stochastic parameter in the instruction list.
`nRealizations`	Number of all possible realizations for the specified stochastic parameter.
`padProbs`	A double array containing probabilities associated with the realizations of the stochastic parameter.
`padVals`	A double array containing values associated with the probabilities.
`nModifyRule`	A flag indicating whether stochastic parameters update the core model by adding or replacing.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSaddEmptySpacesAcolumns *Add empty space A, this can make inserting constraints more efficient.*

Description

R interface function for LINDO API function `LSaddEmptySpacesAcolumns`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddEmptySpacesAcolumns(model,paiColnnz)
```

Arguments

model
: A LINDO API model object, returned by `rLScreateModel`.

paiColnnz
: An integer array containing number of spaces to be inserted into each column.

Value

An R list object with components:

ErrorCode
: Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSaddEmptySpacesNLPAcolumns

Add empty space A, this can make inserting constraints more efficient.

Description

R interface function for LINDO API function `LSaddEmptySpacesNLPAcolumns`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddEmptySpacesNLPAcolumns(model,paiColnnz)
```

Arguments

model
: A LINDO API model object, returned by `rLScreateModel`.

paiColnnz
: An integer array containing number of spaces to be inserted into each column.

Value

An R list object with components:

ErrorCode
: Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSaddInstruct` *Add instruction lists into a model structure.*

Description

R interface function for LINDO API function `LSaddInstruct`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddInstruct(model,nCons,nObjs,nVars,nNumbers,panObjSense,pszConType,
               pszVarType = NULL,panInstruct,nInstruct,paiCons = NULL,
               padNumVal,padVarVal,paiObjBeg,panObjLen,paiConBeg,
               panConLen,padLB = NULL,padUB = NULL)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nCons`	Number of constraints in the model.
`nObjs`	Number of objectives in the model.
`nVars`	Number of variables in the model.
`nNumbers`	Number of real numbers in the model.
`panObjSense`	An integer array containing the indicator stating whether the objective is to be maximized or minimized. Valid values are `LS_MAX` or `LS_MIN`, respectively.
`pszConType`	A character array containing the type of each constraint. Each constraint is represented by a single byte in the array. Valid values for each constraint are 'L', 'E', 'G', or 'N' for less-than-or-equal-to, equal to, great-than-or-equal-to, or neutral, respectively.
`pszVarType`	A character array containing the type of each variable. Valid values for each variable are 'C', 'B', or 'I', for continuous, binary, or general integer, respectively.
`panInstruct`	An integer array containing the instruction list.
`nInstruct`	Number of items in the instruction list.
`paiCons`	An integer array containing the variable index.
`padNumVal`	A double array containing the value of each real number in the model.
`padVarVal`	A double array containing starting values for each variable in the given model.
`paiObjBeg`	An integer array containing the beginning positions on the instruction list for each objective row.
`panObjLen`	An integer array containing the length of instruction code (i.e., the number of individual instruction items) for each objective row.
`paiConBeg`	An integer array containing the beginning positions on the instruction list for each constraint row.
`panConLen`	An integer array containing the length of instruction code (i.e., the number of individual instruction items) for each constraint row.
`padLB`	A double array containing the lower bound of each variable.
`padUB`	A double array containing the upper bound of each variable.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadInstruct`

`rLSaddNLPAj` *Add NLP elements to the specified column for the given model.*

Description

R interface function for LINDO API function `LSaddNLPAj`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddNLPAj(model,iVar1,nRows,paiRows,padAj)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iVar1`	The index of the variable to which NLP elements will be added.
`nRows`	Number of constraints for which NLP elements will be added.
`paiRows`	An integer array containing row indices of the nonlinear elements.
`padAj`	A double array containing the initial nonzero coefficients of the NLP elements

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadNLPData`

`rLSaddNLPobj` *Add NLP elements to the objective function for the given model.*

Description

R interface function for LINDO API function `LSaddNLPobj`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddNLPobj(model,nCols,paiCols,padColj)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nCols`	Number of variables for which NLP elements will be added.
`paiCols`	An integer array containing the variable indices of the nonlinear elements.
`padColj`	A double array containing the initial nonzero coefficients of the NLP elements.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadNLPData`

`rLSaddParamDistIndep` *Add a new independent stochastic parameter with a parametric distribution to the SP model.*

Description

R interface function for LINDO API function `LSaddParamDistIndep`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddParamDistIndep(model,iRow,jCol,iStv,nDistType,nParams,padParams,iModifyRule)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iRow`	Row index of the stochastic parameter.
`jCol`	Column index of the stochastic parameter.
`iStv`	Index of stochastic parameter in the instruction list.
`nDistType`	The parametric distribution type.
`nParams`	Length of padParams.
`padParams`	A double array containing the parameters of given distribution.
`iModifyRule`	A flag indicating whether stochastic parameters update the core model by adding or replacing.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSaddQCShift` *Shift diag(Q_i) by lambda.*

Description

R interface function for LINDO API function `LSaddQCShift`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddQCShift(model,iRow,dShift)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iRow`	An integer specifying the index of the QC row.
`dShift`	A double specifying the shift parameter.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSaddQCterms — *Add quadratic elements to the given model.*

Description

R interface function for LINDO API function `LSaddQCterms`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddQCterms(model,nQCnonzeros,paiQCconndx,paiQCvarndx1,paiQCvarndx2,padQCcoef)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQCnonzeros`	Total number of nonzeros in quadratic coefficient matrices to be added.
`paiQCconndx`	An integer array containing the index of the constraint associated with each nonzero quadratic term.
`paiQCvarndx1`	An integer array containing the first variable defining each quadratic term.
`paiQCvarndx2`	An integer array containing the second variable defining each quadratic term.
`padQCcoef`	A double array containing the nonzero coefficients in the quadratic matrix.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadQCData`

rLSaddScenario — *Add a new scenario block to the SP model.*

Description

R interface function for LINDO API function `LSaddScenario`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddScenario(model,jScenario,iParentScen,iStage,dProb,nElems,
               paiRows,paiCols,paiStvs,padVals,nModifyRule)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	Index of the new scenario to be added.
`iParentScen`	Index of the parent scenario.
`iStage`	Index of the stage the new scenario branches from its parent.
`dProb`	The scenario probability.
`nElems`	Number of stochastic parameters realized at stage iStage in the new scenario.
`paiRows`	An integer array containing the row indices of stochastic parameters.
`paiCols`	An integer array containing the column indices of stochastic parameters.
`paiStvs`	A double array containing indices of stochastic parameters in instruction list.
`padVals`	A double array containing values of stochastic parameters.
`nModifyRule`	A flag indicating whether stochastic parameters update the core model by adding or replacing.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSaddSETS` *Add sets to a given model.*

Description

R interface function for LINDO API function `LSaddSETS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddSETS(model,nSETS,pszSETStype,paiCARDnum,paiSETSbegcol,paiSETScols)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nSETS`	Number of sets to append to the model.
`pszSETStype`	A character array containing the type of each set to be added to the model.
`paiCARDnum`	An integer array containing the cardinalities of the sets to be added.
`paiSETSbegcol`	An integer array containing the index of the first variable in each new set.
`paiSETScols`	An integer array containing the indices of the variables in the new sets.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadSETSData`

`rLSaddVariables` *Add variables to a given model.*

Description

R interface function for LINDO API function `LSaddVariables`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaddVariables(model,nNumaddvars,pszVarTypes,paszVarNames = NULL,
                paiAcols,panAcols = NULL,padAcoef,paiArows,padC,
                padL = NULL,padU = NULL)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nNumaddvars`	Number of variables to append to the model.
`pszVarTypes`	A character array containing the types of each variable to be added to the model.
`paszVarNames`	A string array containing the names of each new variable.
`paiAcols`	An integer array containing the index of the first nonzero element in each new column.
`panAcols`	An integer array containing the length of each column.
`padAcoef`	A double array containing the nonzero coefficients of the new columns.
`paiArows`	An integer array containing the row indices of the nonzeros in the new columns.
`padC`	A double array containing the objective coefficients for each new variable.
`padL`	A double array containing the lower bound of each new variable.
`padU`	A double array containing the upper bound of each new variable.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadLPData rLSaddConstraints

---

rLSaggregateStages *Load stage aggregation scheme for the SP model.*

Description

R interface function for LINDO API function `LSaggregateStages`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSaggregateStages(model,panScheme,nLength)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`panScheme`	An integer array containing the stage aggregation scheme.
`nLength`	Length of panScheme.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLScalinfeasMIPsolution *Calculate the feasibility of a MIP solution.*

Description

R interface function for LINDO API function `LScalinfeasMIPsolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLScalinfeasMIPsolution(model,padPrimalMipsol = NULL)
```

Arguments

model A LINDO API model object, returned by rLScreateModel.

padPrimalMipsol
The primal solution. If it is NULL, the procedure returns the infeasibility of the internal solution if any.

Value

An R list object with components:

ErrorCode Zero if successful, nonzero otherwise.

pdIntPfeas Infeasibility of int variables.

pbConsPfeas Infeasibility of constraints.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLScheckConvexity *Optimize a quadratic model with the best suitable solver.*

Description

R interface function for LINDO API function `LScheckConvexity`. For more information, please refer to LINDO API User Manual.

Usage

```
rLScheckConvexity(model)
```

Arguments

model A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLScopyParam — *Copy model parameters to another model.*

Description

R interface function for LINDO API function `LScopyParam`. For more information, please refer to LINDO API User Manual.

Usage

```
rLScopyParam(smodel,tmodel,nSolverType)
```

Arguments

`smodel`	A LINDO API model object to copy the parameters from.
`tmodel`	A LINDO API model object to copy the parameters to.
`nSolverType`	An integer specifying the solver type to copy the parameters for.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLScreateModel`

rLScreateEnv — *Create a new instance of LINDO API environment object.*

Description

R interface function for LINDO API function `LScreateEnv`. For more information, please refer to LINDO API User Manual.

Usage

```
rLScreateEnv()
```

Details

There is no argument for this function. It goes to folder "LINDOAPI_HOME/license" to locate the license key file and create the LINDO API environment object.

Value

If successful, rLScreateEnv returns a LINDO API evriorment object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSdeleteEnv

---

rLScreateModel *Create a new instance of LINDO API model object.*

Description

R interface function for LINDO API function `LScreateModel`. For more information, please refer to LINDO API User Manual.

Usage

```
rLScreateModel(env)
```

Arguments

`env`	A LINDO API environment object, returned by rLScreateEnv.

Details

Before this function is called, rLScreateEnv must be called to get a valid LINDO API environment object.

Value

If successful, rLScreateModel returns a LINDO API model object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSdeleteModel

rLScreateRG — *Create a new instance of LINDO API random generator object.*

Description

R interface function for LINDO API function `LScreateRG`. For more information, please refer to LINDO API User Manual.

Usage

```
rLScreateRG(env,nMethod)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nMethod`	An integer specifying the random number generator to use. Possible values are: • LS_RANDGEN_FREE • LS_RANDGEN_SYSTEM • LS_RANDGEN_LINDO1 • LS_RANDGEN_LINDO2 • LS_RANDGEN_LIN1 • LS_RANDGEN_MULT1 • LS_RANDGEN_MERSENNE

Details

Before this function is called, `rLScreateEnv` must be called to get a valid LINDO API environment object.

Value

If successful, rLScreateRG returns a LINDO API random generator object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

rLScreateRGMT — *Create a new instance of LINDO API random generator object (multithread thread).*

Description

R interface function for LINDO API function `LScreateRGMT`. For more information, please refer to LINDO API User Manual.

Usage

```
rLScreateRGMT(env,nMethod)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nMethod`	An integer specifying the random number generator to use. Possible values are: • LS_RANDGEN_FREE • LS_RANDGEN_SYSTEM • LS_RANDGEN_LINDO1 • LS_RANDGEN_LINDO2 • LS_RANDGEN_LIN1 • LS_RANDGEN_MULT1 • LS_RANDGEN_MERSENNE

Details

Before this function is called, `rLScreateEnv` must be called to get a valid LINDO API environment object.

Value

If successful, rLScreateRG returns a LINDO API random generator object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSdeduceStages` *Deduce constraints and variables stage info.*

Description

R interface function for LINDO API function `LSdeduceStages`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeduceStages(model,nMaxStage,panRowStagesIn,panColStagesIn,panSparStage)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nMaxStage`	Maximum number of stages
`panRowStagesIn`	An integer array containing constraint stage information.
`panColStagesIn`	An integer array containing variable stage information.
`panSparStage`	An integer array containing random parameter stage information.

Value

An R list object with components:

ErrorCode
: Zero if successful, nonzero otherwise.

panRowStagseOut
: An integer array containing constraint stage information.

panColStagesOut
: An integer array containing variable stage information.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSdeleteAj — *Delete the elements at specified rows for the specified column for the given model.*

Description

R interface function for LINDO API function `LSdeleteAj`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteAj(model,iVar1,nRows,paiRows)
```

Arguments

model	A LINDO API model object, returned by `rLScreateModel`.
iVar1	The index of the variable whose elements will be deleted.
nRows	Number of constraints at which elements will be deleted.
paiRows	An integer array containing the row indices of the elements to be deleted.

Value

An R list object with components:

ErrorCode
: Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSdeleteCones — *Delete a set of cones in the given model.*

Description

R interface function for LINDO API function `LSdeleteCones`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteCones(model,nCones,paiCones)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nCones`	Number of cones to be deleted.
`paiCones`	An integer array containing the indices of the cones to be deleted.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSdeleteConstraints — *Delete a set of constraints in the given model.*

Description

R interface function for LINDO API function `LSdeleteConstraints`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteConstraints(model,nCons,paiCons)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nCons`	Number of constraints to be deleted.
`paiCons`	An integer array containing the indices of the constraints to be deleted.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSdeleteEnv` *Delete LINDO API environment object.*

Description

R interface function for LINDO API function `LSdeleteEnv`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteEnv(env)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.

Details

The memory used by the environment object is freed.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLScreateEnv`

`rLSdeleteModel` *Delete LINDO API model object.*

Description

R interface function for LINDO API function `LSdeleteModel`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteModel(model)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

Details

The memory used by the model object is freed.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLScreateModel`

`rLSdeleteNLPobj` *Delete NLP elements from the objective function for the given model.*

Description

R interface function for LINDO API function `LSdeleteNLPobj`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteNLPobj(model,nCols,paiCols)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

`nCols` Number of variables for which NLP elements will be deleted..

`paiCols` An integer array containing the indices of the variables whose NLP elements are to be deleted.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSdeleteQCterms — *Delete the quadratic terms from a set of constraints in the given model.*

Description

R interface function for LINDO API function `LSdeleteQCterms`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteQCterms(model,nCons,paiCons)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nCons`	Number of constraints in the model whose quadratic terms will be deleted.
`paiCons`	An integer array containing the indices of the constraints whose quadratic terms will be deleted.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSdeleteSemiContVars — *Delete a set of semi-continuous variables in the given model.*

Description

R interface function for LINDO API function `LSdeleteSemiContVars`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteSemiContVars(model,nSCVars,paiSCVars)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nSCVars`	Number of semi-continuous variables to be deleted.
`paiSCVars`	An integer array containing the indices of the semi-continuous variables to be deleted.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSdeleteSETS — *Delete the sets in the given model.*

Description

R interface function for LINDO API function `LSdeleteSETS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteSETS(model,nSETS,paiSETS)
```

Arguments

model	A LINDO API model object, returned by `rLScreateModel`.
nSETS	Number of sets to be deleted.
paiSETS	An integer array containing the indices of the sets to be deleted.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSdeleteString` *Delete the complete string data, including the string vector and values.*

Description

R interface function for LINDO API function `LSdeleteString`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteString(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSdeleteStringData` *Delete the string values data.*

Description

R interface function for LINDO API function `LSdeleteStringData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteStringData(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSdeleteVariables *Delete a set of variables in the given model.*

Description

R interface function for LINDO API function `LSdeleteVariables`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdeleteVariables(model,nVars,paiVars)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nVars`	Number of variables to be deleted.
`paiVars`	An integer array containing the indices of the ariables to be deleted.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSdisposeRG *Delete the specified random generator object.*

Description

R interface function for LINDO API function `LSdisposeRG`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSdisposeRG(rg)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfillRGBuffer` *Generate next batch of random numbers into random number buffer.*

Description

R interface function for LINDO API function `LSfillRGBuffer`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfillRGBuffer(rg)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfindBlockStructure` *Examine the nonzero structure of the constraint matrix and tries to identify block structures in the model.*

Description

R interface function for LINDO API function `LSfindBlockStructure`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfindBlockStructure(model,nBlock,nType)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nBlock`	Number of blocks to decompose the coefficient matrix into.
`nType`	Type of decomposition requested. The possible values are identified with the following macros: • LS_LINK_BLOCKS_NONETry total decomposition (no linking rows or columns). • LS_LINK_BLOCKS_COLSThe decomposed model will have dual angular structure (linking columns). • LS_LINK_BLOCKS_ROWSThe decomposed model will have block angular structure (linking rows).

- LS_LINK_BLOCKS_BOTHThe decomposed model will have both dual and block angular structure (linking rows and columns).
- LS_LINK_BLOCKS_FREESolver decides which type of decomposition to use.

Details

If neither linking rows nor linking columns exist, then the model is called 'totally decomposable'. Unless total decomposition is requested, the user should specify as an input the number of blocks to decompose the matrix into.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfindIIS` *Find an irreducibly inconsistent set (IIS) of constraints for an infeasible model.*

Description

R interface function for LINDO API function `LSfindIIS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfindIIS(model,nLevel)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

`nLevel` The level of analysis in finding the IIS. Bit mask values are:

- LS_NECESSARY_ROWS = 1
- LS_NECESSARY_COLS = 2
- LS_SUFFICIENT_ROWS = 4
- LS_SUFFICIENT_COLS = 8

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfindIUS` *Find an irreducibly unbounded set (IUS) of columns for an unbounded linear program.*

Description

R interface function for LINDO API function `LSfindIUS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfindIUS(model,nLevel)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

`nLevel` The level of analysis in finding the IUS. Bit mask values are:

- LS_NECESSARY_COLS = 2
- LS_SUFFICIENT_COLS = 8

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfreeGOPSolutionMemory` *Free up the arrays associated with the GOP solution of a given model.*

Description

R interface function for LINDO API function `LSfreeGOPSolutionMemory`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfreeGOPSolutionMemory(model)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

Details

After freeing the memory, you will lose all access to the information associated to GOP solutions.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfreeHashMemory` *Free up work arrays associated with a given model's variable name hashing.*

Description

R interface function for LINDO API function `LSfreeHashMemory`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfreeHashMemory(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Details

This will release memory to the system pool, but will cause any subsequent variable name lookup to pause to regenerate these tables.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSfreeMIPSolutionMemory

Free up the arrays associated with the MIP solution of a given model.

Description

R interface function for LINDO API function `LSfreeMIPSolutionMemory`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfreeMIPSolutionMemory(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Details

After freeing the memory, you will lose all access to the information associated to MIP solutions.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSfreeSolutionMemory

Free up the arrays associated with the solution of a given model.

Description

R interface function for LINDO API function `LSfreeSolutionMemory`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfreeSolutionMemory(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Details

This will release the associated memory blocks to the system, but will not cause the solver to loose any warm start capability for the model on its next run. However, you will lose all access to the model's solution information.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfreeSolverMemory` *Free up solver work arrays associated with a given model.*

Description

R interface function for LINDO API function `LSfreeSolverMemory`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfreeSolverMemory(model)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

Details

This will release the associated memory to the system, but will cause any subsequent reoptimization of the model to take more time. In other words, the solver will lose its warm start capability for the model on its next run. Note that by freeing solver memory, you will not lose access to the model's solution information.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfreeStocHashMemory` *Free up Stoch Hashtable memory used in the model for fast name lookup.*

Description

R interface function for LINDO API function `LSfreeStocHashMemory`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfreeStocHashMemory(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSfreeStocMemory` *Free up stochastic memory.*

Description

R interface function for LINDO API function `LSfreeStocMemory`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSfreeStocMemory(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetBasis` *Get information about the basis that was found after optimizing the given model.*

Description

R interface function for LINDO API function `LSgetBasis`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetBasis(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`panCstatus`	An integer array containing information about the status of the variables.
`panCstatus`	An integer array containing information about the status of the constraints.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadBasis`

`rLSgetBestBounds` *Finds the best implied variable bounds for the specified model by improving the original bounds using extensive preprocessing and probing.*

Description

R interface function for LINDO API function `LSgetBestBounds`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetBestBounds(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padBestL`	Best implied lower bounds if different from NULL.
`padBestU`	Best implied upper bounds if different from NULL.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetBlockStructure` *Retrieve the block structure information.*

Description

R interface function for LINDO API function `LSgetBlockStructure`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetBlockStructure(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Details

Following a call to LSfindBlockStructure().

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnBlock`	Number of blocks to decompose the model matrix into.
`panRblock`	An integer array containing information about the block membership of the constraints.
`panCblock`	An integer array containing information about the block membership of the variables.
`pnType`	Type of the decomposition.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSfindBlockStructure`

rLSgetBoundRanges

Retrieves the maximum allowable decrease and increase in the primal variables for which the optimal basis remains unchanged.

Description

R interface function for LINDO API function `LSgetBoundRanges`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetBoundRanges(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padDec`	Maximum allowable decrease in the lower and upper bounds.
`padInc`	Maximum allowable increase in the lower and upper bounds.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetChanceConstraint

Get the stochastic data for the specified chance constraint.

Description

R interface function for LINDO API function `LSgetChanceConstraint`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetChanceConstraint(model,iChance)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iChance`	Index of the chance constraint.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`piSense`	Sense of the chance constraint.
`pnCons`	Number of constraints in the chance-constraint.
`paiCons`	An integer array containing the indices of the constraints in the constraints in the chance-constraint pnCons or more.
`pdProb`	Probability level required.
`pdObjWeight`	Weight of the chance constraint in the probabilistic objective.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetConeDatai` *Retrieve data for cone i.*

Description

R interface function for LINDO API function `LSgetConeDatai`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConeDatai(model,iCone)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iCone`	Index of the cone you wish to receive information on.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pchConeType`	Type of the cone.
`piNnz`	Number of variables characterizing the cone.
`paiCols`	An integer array containing indices of variables characterizing the cone.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetConstraintDatai`

`rLSgetConeIndex` *Retrieve the name of a cone, given its index.*

Description

R interface function for LINDO API function `LSgetConeIndex`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConeIndex(model,pszConeName)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszConeName`	Name of the cone whose index you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`piCone`	Index of the cone.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetConstraintIndex`

`rLSgetConeNamei` *Get the name of a cone with a specified index.*

Description

R interface function for LINDO API function `LSgetConeNamei`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConeNamei(model,iCone)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iCone`	Index of the cone whose name you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pachConeName`	Name of the cone.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetConstraintNamei`

`rLSgetConstraintDatai` *Get data on a specified constraint.*

Description

R interface function for LINDO API function `LSgetConstraintDatai`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConstraintDatai(model,iCon)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iCon`	Index of the constraint you wish to receive information on.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pchConType`	Type of the constraint.
`pchIsNlp`	0 if the constraint is linear and 1 if it is nonlinear.
`pdB`	Right-hand side value of the constraint.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetConstraintIndex *Retrieve the index of a constraint, given its name.*

Description

R interface function for LINDO API function `LSgetConstraintIndex`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConstraintIndex(model,pszConName)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszConName`	Name of the constraint whose index you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`piCon`	Index of the constraint.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetVariableIndex`

rLSgetConstraintNamei *Retrieve the name of a constraint, given its index number.*

Description

R interface function for LINDO API function `LSgetConstraintNamei`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConstraintNamei(model,iCon)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iCon`	Index of the constraint whose name you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pachConName`	Name of the constraint.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetVariableNamej`

`rLSgetConstraintProperty`

Return the property of the specified constraint of the given model.

Description

R interface function for LINDO API function `LSgetConstraintProperty`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConstraintProperty(model,ndxCons)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`ndxCons`	The index of the constraint for which the property is requested.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnConptype`	The constraint property.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSsetConstraintProperty`

`rLSgetConstraintRanges` *Retrieves the maximum allowable decrease and increase in the right-hand side values of constraints for which the optimal basis remains unchanged.*

Description

R interface function for LINDO API function `LSgetConstraintRanges`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConstraintRanges(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padDec`	Maximum allowable decrease in the right-hand sides of constraints.
`padInc`	Maximum allowable increase in the right-hand sides of constraints.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetConstraintStages` *Retrieve the stage indices of constraints.*

Description

R interface function for LINDO API function `LSgetConstraintStages`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetConstraintStages(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`panStage`	An integer array containing the stage indices of constraints in the core model.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetCorrelationMatrix

Get the correlation structure between variables.

Description

R interface function for LINDO API function `LSgetCorrelationMatrix`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetCorrelationMatrix(model,iFlag,nCorrType)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

`iFlag` An integer specifying the sample (original or corr-induced). Possible values are:
- 0: Use independent sample
- 1: Use dependent (correlation induced) sample

`nCorrType` Correlation type. Possible values are:
- LS_CORR_PEARSON
- LS_CORR_SPEARMAN
- LS_CORR_KENDALL
- LS_CORR_TARGET

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnQCnnz`	Number of nonzero correlation coefficients.
`paiQCcols1`	An integer array containing the first index of variable the correlation term belongs to.
`paiQCcols2`	An integer array containing the second index of variable the correlation term belongs to.
`padQCcoef`	A double array containing the correlation terms.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetDeteqModel — *Get the deterministic equivalent (DEQ) of the SP model.*

Description

R interface function for LINDO API function `LSgetDeteqModel`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDeteqModel(model,iDeqType)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iDeqType`	An integer specifying the DEQ type (implicit or explicit). Possible values are: • LS_DETEQ_FREE • LS_DETEQ_IMPLICIT • LS_DETEQ_EXPLICIT

Value

If successful, rLSgetDeteqModel returns a LINDO API model object referring to the DEQ model; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetDInfo — *Return model or solution double information about the current state of the LINDO API solver after model optimization is completed.*

Description

R interface function for LINDO API function `LSgetDInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDInfo(model,nQuery)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	Number of stages/blocks in the dual angular model.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdResult`	A double value for requested information.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetIInfo`

---

rLSgetDiscreteBlockOutcomes

Get the outcomes for the specified block-event at specified block-realization index.

Description

R interface function for LINDO API function `LSgetDiscreteBlockOutcomes`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDiscreteBlockOutcomes(model,iEvent,iRealz)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iEvent`	Index of the discrete block event.
`iRealz`	Index of a block realization in the specified block event.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`nRealz`	Number of individual stochastic parameters consitituting the block realization iRealz.
`paiArows`	An integer array containing the row indices of stochastic parameters.
`paiAcols`	An integer array containing the column indices of stochastic parameters.
`paiStvs`	An integer array containing indices of stochastic parameters.
`padVals`	A double array containing the values associated with the stochastic parameters listed in paiStvs or (paiArows,paiAcols).

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetDiscreteBlocks — *Get the stochastic data for the discrete block event at specified index.*

Description

R interface function for LINDO API function `LSgetDiscreteBlocks`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDiscreteBlocks(model,iEvent)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iEvent`	Index of the discrete block event.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`nDistType`	Distribution type of the event.
`iStage`	Stage index of the block event.
`nRealzBlock`	Number of block realizations in the event.
`padProbs`	A double array containing the event probabilities.
`iModifyRule`	A flag indicating whether stochastic parameters update the core model by adding or replacing.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetDiscreteIndep — *Get the stochastic data for the (independent) discrete stochastic parameter at the specified event index.*

Description

R interface function for LINDO API function `LSgetDiscreteIndep`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDiscreteIndep(model,iEvent)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iEvent`	Index of the discrete independent event.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`nDistType`	Distribution type of the event.
`iStage`	Stage index of the discrete-independent event.
`iRow`	Row index of the stochastic parameter.
`jCol`	Column index of the stochastic parameter.
`iStv`	Index of stochastic parameter in the instruction list.
`nRealizations`	Number of all possible realizations for the stochastic parameter.
`padProbs`	A double array containing the probabilities associated with the realizations of the stochastic parameter.
`padVals`	A double array containing the values associated with the realizations of the stochastic parameter.
`iModifyRule`	A flag indicating whether stochastic parameters update the core model by adding or replacing.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetDistrRV` *Get the next double random variate of underlying distribution.*

Description

R interface function for LINDO API function `LSgetDistrRV`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDistrRV(rg)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdResult`	The next random value from underlying distribution.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetDoubleRV` *Get the next standard uniform random variate in the stream.*

Description

R interface function for LINDO API function `LSgetDoubleRV`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDoubleRV(rg)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdResult`	The result.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetDouParameterRange` *Retrieve the range of a parameter of type double.*

Description

R interface function for LINDO API function `LSgetDouParameterRange`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDouParameterRange(model,nParameter)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nParameter`	An integer parameter identifier.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdValMIN`	The minimum value of parameter.
`pdValMAX`	The maximum value of parameter.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetDualModel` *Construct the explicit dual from a primal problem.*

Description

R interface function for LINDO API function `LSgetDualModel`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDualModel(model,dualmodel)
```

Arguments

`model`	Primal model. A LINDO API model object, returned by `rLScreateModel`.
`dualmodel`	Dual model. A LINDO API model object, returned by `rLScreateModel`.

Details

The dual has its own memory blocks and does not share any with the primal.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetDualSolution` *Return the dual solution values for a given model.*

Description

R interface function for LINDO API function `LSgetDualSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetDualSolution(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padDual`	A double array containing the dual solution.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetPrimalSolution`

`rLSgetEnvDouParameter` *Retrieves a double precision parameter for a specified environment.*

Description

R interface function for LINDO API function `LSgetEnvDouParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetEnvDouParameter(env,nParameter)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nParameter`	An integer referring to a double precision parameter.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdValue`	The parameter's value.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetModelDouParameter`

`rLSgetEnvIntParameter` *Retrieves a integer precision parameter for a specified environment.*

Description

R interface function for LINDO API function `LSgetEnvIntParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetEnvIntParameter(env,nParameter)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nParameter`	An integer referring to an integer precision parameter.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnValue`	The parameter's value.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetModelIntParameter`

`rLSgetEnvStocParameterChar`
Get a stochastic parameter value of type characters from the given env.

Description

R interface function for LINDO API function `LSgetEnvStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetEnvStocParameterChar(env,nQuery)
```

Arguments

`env`	A LINDO API env object, returned by `rLScreateEnv`.
`nQuery`	A valid query macro.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pachResult	The result of type characters.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetEnvStocParameterDou
Get a stochastic parameter value of type double from the given env.

Description

R interface function for LINDO API function `LSgetEnvStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetEnvStocParameterDou(env,nQuery)
```

Arguments

env	A LINDO API env object, returned by rLScreateEnv.
nQuery	A valid query macro.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pdResult	The result of type double.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetEnvStocParameterInt` *Get a stochastic parameter value of type integer from the given env.*

Description

R interface function for LINDO API function `LSgetEnvStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetEnvStocParameterInt(env,nQuery)
```

Arguments

`env`	A LINDO API env object, returned by `rLScreateEnv`.
`nQuery`	A valid query macro.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnResult`	The result of type integer.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetErrorMessage` *Retrieves the error message associated with the given error code.*

Description

R interface function for LINDO API function `LSgetErrorMessage`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetErrorMessage(env,errorcode)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`errorcode`	An integer referring to the error code.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pachMessage`	The error message associated with the given error code.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetErrorRowIndex` *Retrieves the index of the row where a numeric error has occurred.*

Description

R interface function for LINDO API function `LSgetErrorRowIndex`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetErrorRowIndex(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`piRow`	The row index with numeric error.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetFileError` *Provides the line number and text of the line in which an error occurred while reading or writing a file.*

Description

R interface function for LINDO API function `LSgetFileError`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetFileError(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnLinenum`	line number in the I/O file where the error has occurred.
`pachLinetxt`	the text of the line where the error has occurred.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetIInfo` *Return model or solution integer information about the current state of the LINDO API solver after model optimization is completed.*

Description

R interface function for LINDO API function `LSgetIInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetIInfo(model,nQuery)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	Number of stages/blocks in the dual angular model.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnResult	An integer value for requested information.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSgetDInfo

---

rLSgetIIS *Retrieve the irreducibly inconsistent set (IIS) of constraints for an infeasible model.*

Description

R interface function for LINDO API function `LSgetIIS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetIIS(model)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.

Details

Following a call to LSfindIIS().

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnSuf_r	Number of constraints in the sufficient set.
pnIIS_r	Number of rows in the IIS.
paiCons	An integer array containg indices of the rows in the IIS.
pnSuf_c	Number of column bounds in the sufficient set.
pnIIS_c	number of column bounds in the IIS.
paiVars	An integer array containg indices of the column bounds in the IIS.
panBnds	An integer array indicating whether the lower or the upper bound of the variable is in the IIS.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSfindIIS`

`rLSgetInitSeed` *Get the seed initiated this random generator.*

Description

R interface function for LINDO API function `LSgetInitSeed`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetInitSeed(rg)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnResult`	The seed initiated the random generator.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetInt32RV` *Get the next integer random variate in the stream.*

Description

R interface function for LINDO API function `LSgetInt32RV`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetInt32RV(rg,iLow,iHigh)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.
`iLow`	The lower bound.
`iHigh`	The upper bound.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnResult`	The result.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetIntParameterRange` *Retrieve the range of a parameter of type integer.*

Description

R interface function for LINDO API function `LSgetIntParameterRange`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetIntParameterRange(model,nParameter)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nParameter`	An integer parameter identifier.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnValMIN`	The minimum value of parameter.
`pnValMAX`	The maximum value of parameter.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetIUS` *Retrieve the irreducibly unbounded set (IUS) of columns for an unbounded linear model.*

Description

R interface function for LINDO API function `LSgetIUS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetIUS(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Details

Following a call to LSfindIUS().

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnSuf`	Number of columns in the sufficient set.
`pnIUS`	Number of columns in the IIS.
`paiVars`	An integer array containg indices of the columns in the IUS.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSfindIUS`

`rLSgetLPConstraintDatai` *Retrieve the formulation data for a specified constraint in a linear or mixed integer linear program.*

Description

R interface function for LINDO API function `LSgetLPConstraintDatai`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetLPConstraintDatai(model,iCon)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iCon`	Index of the constraint you wish to receive information on.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pchConType`	Type of the constraint.
`pdB`	Right-hand side value of the constraint.
`pnNnz`	Number of nonzero coefficients in the constraint.
`paiVar`	An integer array containing the indices of the variables with nonzero coefficients in the constraint.
`padAcoef`	A double array containing the constraint's nonzero coefficients.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetLPVariableDataj`

`rLSgetLPData` *Retrieve the formulation data for a given linear or mixed integer linear programming model.*

Description

R interface function for LINDO API function `LSgetLPData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetLPData(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnObjSense`	An integer indicating whether the objective is to be maximized or minimized.
`pdObjConst`	A double precision constant to be added to the objective value.
`padC`	A double array containing the linear program's objective coefficients.
`padB`	A double array containing the constraint right-hand side coefficients.
`pachConTypes`	A character array containing the type of each constraint.
`paiAcols`	An integer array containing the index of the first nonzero in each column.
`panAcols`	An integer array containing the length of each column.
`padAcoef`	A double array containing the nonzero coefficients of the constraint matrix.
`paiArows`	An integer array containing the row indices of the nonzeros in the constraint matrix.
`padL`	A double array containing the lower bound of each variable.
`padU`	A double array containing the upper bound of each variable.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadLPData`

`rLSgetLPVariableDataj` *Retrieve the formulation data for a specified variable.*

Description

R interface function for LINDO API function `LSgetLPVariableDataj`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetLPVariableDataj(model,iVar)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iVar`	Index of the variable whose data you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pchVartype`	Type of the variable.
`pdC`	Objective coefficient of the variable.
`pdL`	Lower bound of the variable.
`pdU`	Upper bound of the variable.
`pnAnnz`	Number of nonzero constraint coefficients in the variable's column.
`paiArows`	An integer array containing the row indices of the variable's nonzeros.
`padAcoef`	A double array that returns the variable's nonzero coefficients.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetLPData`

`rLSgetMIPBasis` *Get information about the basis that was found at the node that yielded the optimal MIP solution.*

Description

R interface function for LINDO API function `LSgetMIPBasis`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetMIPBasis(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`panCstatus`	An integer array containing information about the status of the variables.
`panCstatus`	An integer array containing information about the status of the constraints.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadBasis` `rLSgetBasis`

`rLSgetMIPDualSolution` *Get the current dual solution for a MIP model.*

Description

R interface function for LINDO API function `LSgetMIPDualSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetMIPDualSolution(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padDual`	A double array containing the dual solution to the integer model.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetMIPPrimalSolution`

`rLSgetMIPPrimalSolution` *Get the current primal solution for a MIP model.*

Description

R interface function for LINDO API function `LSgetMIPPrimalSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetMIPPrimalSolution(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padPrimal`	A double array containing the primal solution to the integer model.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetMIPReducedCosts` *Get the current reduced cost for a MIP model.*

Description

R interface function for LINDO API function `LSgetMIPReducedCosts`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetMIPReducedCosts(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padReducedCost`	A double array containing the reduced cost to the integer model.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetMIPPrimalSolution`

`rLSgetMIPSlacks` *Get the current slack values for a MIP model.*

Description

R interface function for LINDO API function `LSgetMIPSlacks`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetMIPSlacks(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padSlack`	A double array containing the slack values to the integer model.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetMIPDualSolution`

`rLSgetMIPVarStartPoint` *Retrieve the values of the initial MIP primal solution.*

Description

R interface function for LINDO API function `LSgetMIPVarStartPoint`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetMIPVarStartPoint(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
padPrimal	A double array containing the starting values for each variable in the given MIP model.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadMIPVarStartPoint

---

rLSgetMIPVarStartPointPartial

Retrieve the resident initial point for MIP/MINLP models.

Description

R interface function for LINDO API function `LSgetMIPVarStartPointPartial`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetMIPVarStartPointPartial(model)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnCols	The number of variables in the partial solution.
paiCols	An integer array containing the indicies of variables in the partial solution.
panPrimal	An integer array containing the values of the partial solution.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadMIPVarStartPointPartial

`rLSgetModelDouParameter`

Retrieves a double precision parameter for a specified model.

Description

R interface function for LINDO API function `LSgetModelDouParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetModelDouParameter(model,nParameter)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nParameter`	An integer referring to a double precision parameter.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdValue`	The parameter's value.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetModelIntParameter`

Retrieves a integer precision parameter for a specified model.

Description

R interface function for LINDO API function `LSgetModelIntParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetModelIntParameter(model,nParameter)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nParameter`	An integer referring to an integer precision parameter.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnValue	The parameter's value.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSgetModelDouParameter

---

rLSgetModelStocDouParameter

Get the current value of a double valued parameter for the given model.

Description

R interface function for LINDO API function LSgetModelStocDouParameter. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetModelStocDouParameter(model,iPar)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
iPar	A valid parameter macro.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pdValue	The current value of the parameter.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetModelStocIntParameter`
Get the current value of an integer valued parameter for the given model.

Description

R interface function for LINDO API function `LSgetModelStocIntParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetModelStocIntParameter(model,iPar)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iPar`	A valid parameter macro.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`piValue`	The current value of the parameter.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetModelStocParameterChar`
Get a stochastic parameter value of type characters from the given model.

Description

R interface function for LINDO API function `LSgetModelStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetModelStocParameterChar(model,nQuery)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid query macro.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pachResult	The result of type characters.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetModelStocParameterDou

Get a stochastic parameter value of type double from the given model.

Description

R interface function for LINDO API function `LSgetModelStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetModelStocParameterDou(model,nQuery)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nQuery	A valid query macro.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pdResult	The result of type double.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetModelStocParameterInt` *Get a stochastic parameter value of type integer from the given model.*

Description

R interface function for LINDO API function `LSgetModelStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetModelStocParameterInt(model,nQuery)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid query macro.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnResult`	The result of type integer.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetNextBestMIPSol` *Generate the next best (in terms of objective value) solution for the current mixed integer model.*

Description

R interface function for LINDO API function `LSgetNextBestMIPSol`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNextBestMIPSol(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Details

Repeated calls to `rLSgetNextBestMIPSol` will allow one to generate the so-called K-Best solutions to mixed-integer model. This is useful for revealing alternate optima.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnStatus	The status on the new, next-best solution.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetNLPConstraintDatai

Get data about the nonlinear structure of a specific row of the model.

Description

R interface function for LINDO API function `LSgetNLPConstraintDatai`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNLPConstraintDatai(model,iCon)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
iCon	Index of the constraint you wish to receive information on.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnNnz	Number of nonlinear nonzeros in constraint i.
paiNLPcols	An integer array containing the column indices of the nonlinear nonzeros in the ith row of the constraint matrix.
padNLPcoef	A double array containing the current values of the nonzero coefficients in the ith row of the coefficient (Jacobian) matrix.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSgetNLPData

rLSgetNLPData — *Get data about the nonlinear structure of a model.*

Description

R interface function for LINDO API function LSgetNLPData. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNLPData(model)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
paiNLPcols	An integer array containing the index of the first nonlinear nonzero in each column.
panNLPcols	An integer array containing the number of nonlinear elements in each column.
padNLPcoef	A double array containing the current values of the nonzero coefficients in the (Jacobian) matrix.
paiNLProws	An integer array containing the row indices of the nonlinear nonzeros in the coefficient matrix.
pnNLPobj	Number of nonlinear variables in the objective function.
paiNLPobj	An integer array containing column indices of the nonlinear terms in the objective.
padNLPobj	A double array containing the current partial derivatives of the objective corresponding to the variables in paiNLPobj.
pachNLPConTypes	A character array whose elements indicate whether a constraint has nonlinear terms or not.If pachNLPConTypes[i] > 0, then constraint i has nonlinear terms.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadNLPData

`rLSgetNLPObjectiveData` *Get the NLP data of the Objective.*

Description

R interface function for LINDO API function `LSgetNLPObjectiveData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNLPObjectiveData(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnNLPobjnnz`	Number of NLP nonzeros in objective.
`paiNLPobj`	An integer array containing the column indices for NLP elements in the objective.
`padNLPobj`	A double array containing NLP row values for NLP elements in the objective.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetNLPData`

`rLSgetNLPVariableDataj` *Get data about the nonlinear structure of a specific column of the model.*

Description

R interface function for LINDO API function `LSgetNLPVariableDataj`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNLPVariableDataj(model,iVar)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iVar`	Index of the variable you wish to receive information on.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnNnz`	Number of nonlinear nonzeros in column j.
`panNLProws`	An integer array containing the row indices of the nonlinear nonzeros in the jth column of the constraint matrix.
`padNLPcoef`	A double array containing the current values of the nonzero coefficients in the jth column of the coefficient (Jacobian) matrix with respect to the last primal solution computed during the iterations.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetNLPData`

`rLSgetNodeDualSolution` *Return the dual solution for the specified node.*

Description

R interface function for LINDO API function `LSgetNodeDualSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNodeDualSolution(model,jScenario,iStage)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.
`iStage`	An integer specifying the stage index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padY`	A double array containing the specified node's dual solution.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetNodeListByScenario` *Retrieve the indices of the nodes that belong to a given scenario.*

Description

R interface function for LINDO API function `LSgetNodeListByScenario`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNodeListByScenario(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`paiNodes`	An integer array containing he node list constituting the scenario.
`pnNodes`	Actual number of nodes on the scenario.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetNodePrimalSolution` *Return the primal solution for the specified node.*

Description

R interface function for LINDO API function `LSgetNodePrimalSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNodePrimalSolution(model,jScenario,iStage)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.
`iStage`	An integer specifying the stage index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padX`	A double array containing the specified node's primal solution.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetNodeReducedCost` *Return the reduced cost for the specified node.*

Description

R interface function for LINDO API function `LSgetNodeReducedCost`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNodeReducedCost(model,jScenario,iStage)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.
`iStage`	An integer specifying the stage index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padX`	A double array containing the specified node's reduced cost.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetNodeSlacks` *Return the slack values for the specified node.*

Description

R interface function for LINDO API function `LSgetNodeSlacks`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetNodeSlacks(model,jScenario,iStage)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.
`iStage`	An integer specifying the stage index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padY`	A double array containing the specified node's slack values.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetObjectiveRanges` *Retrieves the maximum allowable decrease and increase in objective function coefficients for which the optimal basis remains unchanged.*

Description

R interface function for LINDO API function `LSgetObjectiveRanges`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetObjectiveRanges(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padDec`	Maximum allowable decrease in the objective function coefficients.
`padInc`	Maximum allowable increase in the objective function coefficients.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetParamDistIndep` *Get the stochastic data for the (independent) parametric stochastic parameter at the specified event index.*

Description

R interface function for LINDO API function `LSgetParamDistIndep`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetParamDistIndep(model,iEvent)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iEvent`	Index of the discrete independent event.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`nDistType`	Distribution type of the event.
`iStage`	Stage index of the discrete-independent event.
`iRow`	Row index of the stochastic parameter.
`jCol`	Column index of the stochastic parameter.
`iStv`	Index of stochastic parameter in the instruction list.
`nParams`	Length of padParams.
`padParams`	A double array containing parameters defining the underlying distribution.
`iModifyRule`	A flag indicating whether stochastic parameters update the core model by adding or replacing.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetParamLongDesc` *Get the specified parameter's long description.*

Description

R interface function for LINDO API function `LSgetParamLongDesc`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetParamLongDesc(env,nParam)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nParam`	An integer parameter identifier.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`szDescription`	The parameter's long description.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetParamMacroID` *Get the integer identifier and the data type of a parameter specified by its name.*

Description

R interface function for LINDO API function `LSgetParamMacroID`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetParamMacroID(env,szParam)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`szParam`	A parameter macro name.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnParamType`	The data type of the parameter.
`pnParam`	The integer identifier of the parameter.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetParamMacroName` *Get the specified parameter's macro name.*

Description

R interface function for LINDO API function `LSgetParamMacroName`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetParamMacroName(env,nParam)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nParam`	An integer parameter identifier.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`szParam`	The macro's name.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetParamShortDesc *Get the specified parameter's short description.*

Description

R interface function for LINDO API function LSgetParamShortDesc. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetParamShortDesc(env,nParam)
```

Arguments

env	A LINDO API environment object, returned by rLScreateEnv.
nParam	An integer parameter identifier.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
szDescription	The parameter's short description.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetPrimalSolution *Return the primal solution values for a given model.*

Description

R interface function for LINDO API function LSgetPrimalSolution. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetPrimalSolution(model)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
padPrimal	A double array containing the primal solution.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSgetDualSolution

`rLSgetProbabilityByNode` *Return the probability of a given node in the stochastic tree.*

Description

R interface function for LINDO API function `LSgetProbabilityByNode`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetProbabilityByNode(model,iNode)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iNode`	An integer specifying the node index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdProb`	The probabability of the node.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetProbabilityByScenario` *Return the probability of a given scenario.*

Description

R interface function for LINDO API function `LSgetProbabilityByScenario`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetProbabilityByScenario(model,jScenario)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
jScenario	An integer specifying the scenario index.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pdProb	The probabability of the scenario.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetProfilerContext *Get profiler info for the specified context.*

Description

R interface function for LINDO API function LSgetProfilerContext. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetProfilerContext(model,mContext)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
mContext	An integer macro specifying the profiler context.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
szDescription	Profiler context description.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetProfilerInfo` *Get profiler info for the specified context.*

Description

R interface function for LINDO API function `LSgetProfilerInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetProfilerInfo(model,mContext)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`mContext`	An integer macro specifying the profiler context.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnCalls`	The number of calls/hits to the context.
`pdElapsedTime`	The elapsed time in the context.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetQCData` *Retrieve the quadratic data from a model data structure.*

Description

R interface function for LINDO API function `LSgetQCData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetQCData(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`paiQCrows`	An integer array containing the index of the constraint associated with each quadratic term with a nonzero coefficient.
`paiQCcols1`	An integer array containing the index of the first variable defining each quadratic term.
`paiQCcols2`	An integer array containing the index of the second variable defining each quadratic term.
`padQCcoef`	A double array containing the nonzero coefficients in the quadratic matrix.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadQCData`

`rLSgetQCDatai` *Retrieve the quadratic data associated with constraint i from a model data structure.*

Description

R interface function for LINDO API function `LSgetQCDatai`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetQCDatai(model,iCon)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iCon`	The index of the constraint for which the quadratic data will be retrieved.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnQCnnz`	An integer indicating the number of nonzeros in the coefficient matrix of the quadratic term.
`paiQCcols1`	An integer array containing the index of the first variable defining each quadratic term.
`paiQCcols2`	An integer array containing the index of the second variable defining each quadratic term.
`padQCcoef`	A double array containing the nonzero coefficients in the quadratic matrix.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetQCData`

`rLSgetQCEigs` *Finds a few eigenvalues and eigenvectors of a Q matrix for the specified model.*

Description

R interface function for LINDO API function `LSgetQCEigs`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetQCEigs(model,iRow,pachWhich,nEigval,nCV,dTol,nMaxIter)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iRow`	The row index of the quadratic constraint for which the eigenvalues of the associated Q matrix will be computed.
`pachWhich`	A character array specifying the type of eigenvalues to return.
`nEigval`	The Number of eigenvalues to be computed.
`nCV`	The number of columns of the matrix padEigvec.
`dTol`	Stopping tolerance which is the relative accuracy of the Ritz value.
`nMaxIter`	Maximum number of iterations.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padEigVal`	Eigenvalues.
`paEigVec`	Eigenvectors.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetQCShift *Shift diag(Q_i) by lambda.*

Description

R interface function for LINDO API function LSgetQCShift. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetQCShift(model,iRow)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
iRow	An integer specifying the index of the QC row.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pdShift	The shift parameter.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetRangeData *Get constraint ranges.*

Description

R interface function for LINDO API function LSgetRangeData. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetRangeData(model)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
padR	A double array containing the range data.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetReducedCosts — *Return the reduced cost of all variables for a given model.*

Description

R interface function for LINDO API function `LSgetReducedCosts`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetReducedCosts(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padReducedCost`	A double array containing the reduced cost.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetPrimalSolution`

---

rLSgetReducedCostsCone — *Return the reduced cost of all cone variables for a given model.*

Description

R interface function for LINDO API function `LSgetReducedCostsCone`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetReducedCostsCone(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
padReducedCost	A double array containing the reduced cost.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSgetPrimalSolution

---

rLSgetRoundMIPsolution

Round the given MIP solution to nearest int solution.

Description

R interface function for LINDO API function `LSgetRoundMIPsolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetRoundMIPsolution(model,padPrimal = NULL,iUseOpti)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
padPrimal	The primal solution. If it is NULL, the solution in `model` will be rounded.
iUseOpti	Whether to use reoptimization after integers have been rounded and fixed to there optimal value.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
padPrimalRound	The rounded solution.
pdObjRound	Objective value of the rounded solution.
pdPfeasRound	Infeasibility of the rounded solution.
pnstatus	Status of solution if reoptimization have been used.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetSampleSizes` *Retrieve the number of nodes to be sampled in all stages.*

Description

R interface function for LINDO API function `LSgetSampleSizes`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetSampleSizes(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`panSampleSize`	An integer array containing the sample size per stage.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetScenario` *Get the outcomes for the specified specified scenario.*

Description

R interface function for LINDO API function `LSgetScenario`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenario(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	Index of a scenario realization.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
iParentScen	Index of parent scenario.
iBranchStage	The branching stage.
pdProb	Event probability of scenario.
nRealz	Number of individual stochastic parameters consitituting the scenario.
paiArows	An integer array containing the row indices of stochastic parameters in the scenario.
paiAcols	An integer array containing the column indices of stochastic parameters in the scenario.
paiStvs	An integer array containing indices of stochastic parameters in the scenario.
padVals	A double array containing values associated with the stochastic parameters listed in paiStvs or (paiArows,paiAcols).
iModifyRule	A flag indicating whether stochastic parameters update the core model by adding or replacing.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetScenarioDualSolution`
Return the dual solution for the specified scenario.

Description

R interface function for LINDO API function `LSgetScenarioDualSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioDualSolution(model,jScenario)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
jScenario	An integer specifying the scenario index.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
padY	A double array containing the scenario's dual solution.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetScenarioIndex` *Get index of a scenario by its name.*

Description

R interface function for LINDO API function `LSgetScenarioIndex`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioIndex(model,pszName)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszName`	The name of the scenario to return the index for.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnIndex`	Index of the scenario.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetScenarioModel` *Get a copy of the scenario model.*

Description

R interface function for LINDO API function `LSgetScenarioModel`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioModel(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario to retrieve.

Value

If successful, rLSgetScenarioModel returns a LINDO API model object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetScenarioName *Get scenario name by index.*

Description

R interface function for LINDO API function `LSgetScenarioName`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioName(model,nIndex)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nIndex`	Index of the scenario.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pachName`	Name of the scenario.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetScenarioObjective *Return the objective value for the specified scenario.*

Description

R interface function for LINDO API function `LSgetScenarioObjective`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioObjective(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdObj`	The obejctive value for the scenario.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetScenarioPrimalSolution`
Return the primal solution for the specified scenario.

Description

R interface function for LINDO API function `LSgetScenarioPrimalSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioPrimalSolution(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padX`	A double array containing the scenario's primal solution.
`pdObj`	Objective value for the specified scenario.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetScenarioReducedCost

Return the reduced cost for the specified scenario.

Description

R interface function for LINDO API function `LSgetScenarioReducedCost`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioReducedCost(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padX`	A double array containing the scenario's reduced cost.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetScenarioSlacks

Return the primal slacks for the specified scenario.

Description

R interface function for LINDO API function `LSgetScenarioSlacks`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetScenarioSlacks(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	An integer specifying the scenario index.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
padY	A double array containing the scenario's primal slacks.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetSemiContData *Retrieve the semi continuous data from a model data structure.*

Description

R interface function for LINDO API function `LSgetSemiContData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetSemiContData(model)
```

Arguments

model	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
piNvars	Number of semi-continuous variables.
panVarndx	An integer array containing the indices of semi-continuous variables.
padL	A double array containing the lower bounds of semi-continuous variables.
padU	A double array containing the upper bounds of semi-continuous variables.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadSemiContData`

`rLSgetSETSData` *Retrieve sets data from a model data structure.*

Description

R interface function for LINDO API function `LSgetSETSData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetSETSData(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`piNsets`	Number of sets in the model.
`piNtnz`	Total number of variables in the sets.
`pachSETtype`	A character array containing the type of sets in the model.
`piCardnum`	An integer array containing the cardinalities of sets in the model.
`piNnz`	An integer array containing the number of variables in each set in the model.
`piBegset`	An integer array containing the index of the first variable in each set.
`piVarndx`	An integer array containing the indices of the variables in the sets.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadSETSData`

`rLSgetSETSDatai` *Retrieve the data for set i from a model data structure.*

Description

R interface function for LINDO API function `LSgetSETSDatai`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetSETSDatai(model,iSet)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iSet`	The index of the set to retrieve the data for.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pachSETtype`	Set type.
`piCardnum`	Set cardinality.
`piNnz`	Number of variables in the set.
`piVarndx`	An integer array containing the indices of the variables in the set.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetSETSData`

`rLSgetSlacks` *Return the value of the slack variable for each constraint of a given model.*

Description

R interface function for LINDO API function `LSgetSlacks`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetSlacks(model)
```

Arguments

model A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode Zero if successful, nonzero otherwise.

padSlack A double array containing the values of the slack variables.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSgetDualSolution

---

rLSgetSolution *Get the solution specified by the second argument.*

Description

R interface function for LINDO API function LSgetSolution. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetSolution(model,nWhich)
```

Arguments

model A LINDO API model object, returned by rLScreateModel.

nWhich An integer parameter specifying the solution to be retrieved. Possible values are:

- LSSOL_BASIC_PRIMAL
- LSSOL_BASIC_DUAL
- LSSOL_BASIC_SLACK
- LSSOL_BASIC_REDCOST
- LSSOL_INTERIOR_PRIMAL
- LSSOL_INTERIOR_DUAL
- LSSOL_INTERIOR_SLACK
- LSSOL_INTERIOR_REDCOST

Value

An R list object with components:

ErrorCode Zero if successful, nonzero otherwise.

padResult A double array containing the specified solution.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStageAggScheme` *Get stage aggregation scheme for the SP model.*

Description

R interface function for LINDO API function `LSgetStageAggScheme`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStageAggScheme(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`panScheme`	An integer array containing the stage aggregation scheme.
`pnLength`	Length of panScheme.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStageIndex` *Get the index of a stage by its name.*

Description

R interface function for LINDO API function `LSgetStageIndex`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStageIndex(model,pszName)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszName`	The name of the stage to return the index for.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnIndex	Index of the stage.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStageName` *Get scenario name by index.*

Description

R interface function for LINDO API function `LSgetStageName`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStageName(model,nIndex)
```

Arguments

model	A LINDO API model object, returned by `rLScreateModel`.
nIndex	Index of the Stage.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pachName	Name of the Stage.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetStocCCPDInfo — *Get double information about the current state of the stochastic model.*

Description

R interface function for LINDO API function `LSgetStocCCPInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocCCPDInfo(model,nQuery,nScenarioIndex,nCPPIndex)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid information macro.
`nScenarioIndex`	The scenario index.
`nCPPIndex`	The chance constraint index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdResult`	A double value of the information.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetStocCCPIInfo — *Get integer information about the current state of the stochastic model.*

Description

R interface function for LINDO API function `LSgetStocCCPInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocCCPIInfo(model,nQuery,nScenarioIndex,nCPPIndex)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid information macro.
`nScenarioIndex`	The scenario index.
`nCPPIndex`	The chance constraint index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnResult`	An integer value of the information.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStocCCPSInfo` *Get string information about the current state of the stochastic model.*

Description

R interface function for LINDO API function `LSgetStocCCPInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocCCPSInfo(model,nQuery,nScenarioIndex,nCPPIndex)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid information macro.
`nScenarioIndex`	The scenario index.
`nCPPIndex`	The chance constraint index.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pszResult`	A string value of the information.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetStocDInfo — *Get double information about the current state of the stochastic model.*

Description

R interface function for LINDO API function `LSgetStocInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocDInfo(model,nQuery,nParam)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid information macro.
`nParam`	The parameter of the query.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdResult`	A double value of the information.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetStocIInfo — *Get integer information about the current state of the stochastic model.*

Description

R interface function for LINDO API function `LSgetStocInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocIInfo(model,nQuery,nParam)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid information macro.
`nParam`	The parameter of the query.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnResult`	An integer value of the information.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStocParData` *Retrieve the data of stochastic parameters.*

Description

R interface function for LINDO API function `LSgetStocParData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocParData(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`paiStages`	An integer array containing the stages of stochastic parameters.
`padVals`	A double array containing the values of stochastic parameters for the specified scenario.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetStocParIndex *Get the index of stochastic parameter by name.*

Description

R interface function for LINDO API function `LSgetStocParIndex`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocParIndex(model,pszName)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszName`	The name of the stochastic parameter to return the index for.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnIndex`	Index of the stochastic parameter.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetStocParName *Get name of stochastic parameter by index.*

Description

R interface function for LINDO API function `LSgetStocParName`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocParName(model,nIndex)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nIndex`	Index of the stochastic parameter.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pachName`	The name of the stochastic parameter.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStocParOutcomes` *Retrieve the outcomes of stochastic parameters for the specified scenario.*

Description

R interface function for LINDO API function `LSgetStocParOutcomes`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocParOutcomes(model,jScenario)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	Index of the scenario.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padVals`	A double array containing the values of stochastic parameters for the specified scenario.
`pdProb`	Probability of the scenario.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStocParSample` *Get a sample object associated with the specified stochastic parameter.*

Description

R interface function for LINDO API function `LSgetStocParSample`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocParSample(model,iStv,iRow,jCol)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iStv`	Index of stochastic parameter in the instruction list.
`iRow`	Row index of the stochastic parameter.
`jCol`	Column index of the stochastic parameter.

Value

If successful, rLSgetStocParSample returns a LINDO API sample object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStocRowIndices` *Retrieve the indices of stochastic rows.*

Description

R interface function for LINDO API function `LSgetStocRowIndices`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocRowIndices(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`paiSrows`	An integer array containing the indices of stochastic rows in the core model.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStocSInfo` *Get string information about the current state of the stochastic model.*

Description

R interface function for LINDO API function `LSgetStocInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStocSInfo(model,nQuery,nParam)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid information macro.
`nParam`	The parameter of the query.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pszResult`	A string value of the information.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetStringValue` *Retrieve a string value for a specified string index.*

Description

R interface function for LINDO API function `LSgetStringValue`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetStringValue(model,iString)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iString`	The index of the string whose value you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdValue`	The string value.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSgetVariableIndex` *Retrieve the index of a variable, given its name.*

Description

R interface function for LINDO API function `LSgetVariableIndex`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetVariableIndex(model,pszVarName)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszVarName`	Name of the variable whose index you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`piVar`	Index of the variable.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSgetVariableNamej`

rLSgetVariableNamej *Retrieve the name of a variable, given its index number.*

Description

R interface function for LINDO API function `LSgetVariableNamej`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetVariableNamej(model,iVar)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iVar`	Index of the variable whose name you wish to retrieve.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pachVarName`	Name of the variable.

References

LINDO SYSTEMS home page at www.lindo.com

rLSgetVariableStages *Retrieve the stage indices of variables.*

Description

R interface function for LINDO API function `LSgetVariableStages`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetVariableStages(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`panStage`	An integer array containing the stage indices of variables in the core model.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSgetVarStartPoint *Retrieve the values of the initial primal solution.*

Description

R interface function for LINDO API function `LSgetVarStartPoint`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetVarStartPoint(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`padPrimal`	A double array containing the starting values for each variable in the given model.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadVarStartPoint`

---

rLSgetVarStartPointPartial *Retrieve the resident partial initial point for NLP models.*

Description

R interface function for LINDO API function `LSgetVarStartPointPartial`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetVarStartPointPartial(model)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnCols	The number of variables in the partial solution.
paiCols	An integer array containing the indicies of variables in the partial solution.
padPrimal	A double array containing the values of the partial solution.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadVarStartPointPartial

---

rLSgetVarType — *Retrieve the variable types and their respective counts in a given model.*

Description

R interface function for LINDO API function `LSgetVarType`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSgetVarType(model)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pachVarTypes	A character array containing the type of each variable.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadVarType

rLSloadBasis — *Provide a starting basis for the simplex method.*

Description

R interface function for LINDO API function `LSloadBasis`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadBasis(model,panCstatus,panRstatus)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`panCstatus`	An integer array containing the status of each column in the given model. Set each variable's element to 0, -1, -2, or -3 for Basic, Nonbasic at lower bound, Nonbasic at upper bound, or Free and nonbasic at zero value, respectively.
`panRstatus`	An integer array containing the status of each row in the given model. Set each row's element to 0 or -1 if row's associated slack variable is basic or row's associated slack variable is nonbasic at zero, respectively

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadBlockStructure — *Provide a block structure for the constraint matrix by specifying block memberships of each variable and constraint.*

Description

R interface function for LINDO API function `LSloadBlockStructure`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadBlockStructure(model,nBlock,panRblock,panCblock,nType)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nBlock`	The name of the file from which to read the starting values.
`panRblock`	An integer array in which information about the block membership of the constraints is placed.
`panCblock`	An integer array in which information about the block membership of the variables is placed.
`nType`	The type of decomposition loaded. The possible values are identified with the following macros: • LS_LINK_BLOCKS_COLSThe decomposed model has dual angular structure (linking columns). • LS_LINK_BLOCKS_ROWSThe decomposed model has block angular structure (linking rows). • LS_LINK_BLOCKS_BOTHThe decomposed model has both dual and block angular structure (linking rows and columns).

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadVarStartPoint`

`rLSloadConeData` *Load quadratic cone data into a model structure.*

Description

R interface function for LINDO API function `LSloadConeData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadConeData(model,nCone,pszConeTypes,paiConebegcone,paiConecols)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nCone`	Number of cones to add.
`pszConeTypes`	A character array containing the type of each cone being added. Valid values for each cone are 'Q' and 'R'.
`paiConebegcone`	An integer array containing the index of the first variable that appears in the definition of each cone.
`paiConecols`	An integer array containing the indices of variables representing each cone.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadConstraintStages

Load stage structure of the constraints in the model.

Description

R interface function for LINDO API function `LSloadConstraintStages`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadConstraintStages(model,panStage)
```

Arguments

model	A LINDO API model object, returned by `rLScreateModel`.
panStage	An integer array containing information about the stage membership of the constraints.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadCorrelationMatrix

Load a correlation matrix to be used by the sampling scheme in stochastic programming.

Description

R interface function for LINDO API function LSloadCorrelationMatrix. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadCorrelationMatrix(model,nDim,nCorrType,nQCnnz,paiQCcols1,
                         paiQCcols2,padQCcoef)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nDim	Number of stochastic parameters involved in the correlation structure.
nCorrType	Correlation type. Possible values are: • LS_CORR_PEARSON • LS_CORR_SPEARMAN • LS_CORR_KENDALL
nQCnnz	Number of nonzero correlation coefficients.
paiQCcols1	An integer array containing the first index of variable the correlation term belongs to.
paiQCcols2	An integer array containing the second index of variable the correlation term belongs to.
padQCcoef	A double array containing the correlation terms.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadGASolution — *Loads the GA solution at specified index in the final population to the main solution structures for access with solution query routines.*

Description

R interface function for LINDO API function `LSloadGASolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadGASolution(model,nIndex)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nIndex`	Index of the individual in the final population.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadInstruct — *Load instruction lists into a model structure.*

Description

R interface function for LINDO API function `LSloadInstruct`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadInstruct(model,nCons,nObjs,nVars,nNumbers,panObjSense,pszConType,
                pszVarType  = NULL,panInstruct,nInstruct,paiVars = NULL,
                padNumVal,padVarVal,paiObjBeg,panObjLen,paiConBeg,
                panConLen,padLB = NULL,padUB = NULL)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nCons	Number of constraints in the model.
nObjs	Number of objectives in the model.
nVars	Number of variables in the model.
nNumbers	Number of real numbers in the model.
panObjSense	An integer array containing the indicator stating whether the objective is to be maximized or minimized. Valid values are LS_MAX or LS_MIN, respectively.
pszConType	A character array containing the type of each constraint. Each constraint is represented by a single byte in the array. Valid values for each constraint are 'L', 'E', 'G', or 'N' for less-than-or-equal-to, equal to, great-than-or-equal-to, or neutral, respectively.
pszVarType	A character array containing the type of each variable. Valid values for each variable are 'C', 'B', or 'I', for continuous, binary, or general integer, respectively.
panInstruct	An integer array containing the instruction list.
nInstruct	Number of items in the instruction list.
paiVars	An integer array containing the variable index.
padNumVal	A double array containing the value of each real number in the model.
padVarVal	A double array containing starting values for each variable in the given model.
paiObjBeg	An integer array containing the beginning positions on the instruction list for each objective row.
panObjLen	An integer array containing the length of instruction code (i.e., the number of individual instruction items) for each objective row.
paiConBeg	An integer array containing the beginning positions on the instruction list for each constraint row.
panConLen	An integer array containing the length of instruction code (i.e., the number of individual instruction items) for each constraint row.
padLB	A double array containing the lower bound of each variable.
padUB	A double array containing the upper bound of each variable.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadLPData *Load the given LP data into a model structure.*

Description

R interface function for LINDO API function LSloadLPData. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadLPData(model,nCons,nVars,nObjSense,dObjConst,padC,padB,pszConTypes,nAnnz,
              paiAcols,panAcols = NULL,padAcoef,paiArows,padL  = NULL,padU = NULL)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nCons	Number of constraints in the model.
nVars	Number of variables in the model.
nObjSense	An indicator stating whether the objective is to be maximized or minimized.
dObjConst	A constant value to be added to the objective value.
padC	A double array containing the objective coefficients.
padB	A double array containing the constraint right hand side coefficients.
pszConTypes	A character vector containing the type of each constraint. Each constraint is represented by a single byte in the array. Valid values for each constraint are 'L', 'E', 'G', or 'N' for less than or equal to, equal to, greater than or equal to, or neutral, respectively.
nAnnz	The number of nonzeros in the constraint matrix.
paiAcols	An integer array containing the index of the first nonzero in each column.
panAcols	An integer array containing the length of each column.
padAcoef	A double array containing the nonzero coefficients of the constraint matrix.
paiArows	An integer array containing the row indices of the nonzeros in the constraint matrix.
padL	A double array vector containing the lower bound of each variable.
padU	A double array vector containing the upper bound of each variable.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadMIPVarStartPoint *Provide an initial starting point for* LSsolveMIP.

Description

R interface function for LINDO API function LSloadMIPVarStartPoint. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadMIPVarStartPoint(model,padPrimal)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
padPrimal	A double array containing starting values for each variable in the given model.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadMIPVarStartPointPartial *Load a partial MIP initial point for MIP/MINLP models.*

Description

R interface function for LINDO API function LSloadMIPVarStartPointPartial. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadMIPVarStartPointPartial(model,nCols,paiCols,paiPrimal)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nCols	Number of variables in the partial solution.
paiCols	An integer array containing the indicies of variables in the partial solution.
paiPrimal	An integer array containing starting values for each variable in the given model.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadMIPVarStartPoint`

`rLSloadMultiStartSolution`

Load the multistart solution at specified index to the main solution structures for access with solution query routines.

Description

R interface function for LINDO API function `LSloadMultiStartSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadMultiStartSolution(model,nIndex)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

`nIndex` Index of the multistart solution.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSloadNameData` *Load the given name data (e.g., row and column names), into a model structure.*

Description

R interface function for LINDO API function `LSloadNameData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadNameData(model,pszTitle = NULL,pszObjName = NULL,pszRhsName = NULL,
                pszRngName = NULL,pszBndname = NULL,paszConNames = NULL,
                paszVarNames = NULL,paszConeNames = NULL)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszTitle`	The title of the problem.
`pszObjName`	The name of the objective.
`pszRhsName`	The name of the right-hand side vector.
`pszRngName`	The name of the range vector.
`pszBndname`	The name of the bounds vector.
`paszConNames`	The constraint names.
`paszVarNames`	The variable names.
`paszConeNames`	The cone names.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadNLPData *Load a nonlinear program's data into the model data structure.*

Description

R interface function for LINDO API function `LSloadNLPData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadNLPData(model,paiNLPcols,panNLPcols,padNLPcoef = NULL,paiNLProws,
               nNLPobj,paiNLPobj,padNLPobj = NULL)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`paiNLPcols`	An integer array containing the index of the first nonlinear nonzero in each column.
`panNLPcols`	An integer array containing the number of nonlinear elements in each column.
`padNLPcoef`	A double array containing initial values of the nonzero coefficients in the (Jacobian) matrix.
`paiNLProws`	An integer array containing the row indices of the nonlinear elements.
`nNLPobj`	The number of nonlinear variables in the objective.
`paiNLPobj`	An integer array containing the column indices of nonlinear variables in the objective function.
`padNLPobj`	A double array containing the initial nonzero coefficients in the objective.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadPOSDData *Load the given POSD data into the LSmodel data structure.*

Description

R interface function for LINDO API function `LSloadPOSDData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadPOSDData(model,nPOSD,paiPOSDdim,paiPOSDbeg,
                paiPOSDrowndx,paiPOSDcolndx,paiPOSDvarndx)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nPOSD`	The number of PSD diagonal blocks to load.
`paiPOSDdim`	A vector containing the dimensions of PSD diagonal blocks.
`paiPOSDbeg`	A vector containing beginning position of each PSD matrix in paiPOSDrowndx, paiPOSDcolndx and paiPOSDvarndx vectors.
`paiPOSDrowndx`	A vector specifying the row indices of variables within PSD matrix blocks.
`paiPOSDcolndx`	A vector specifying the column indices of variables within PSD matrix blocks.
`paiPOSDvarndx`	A vector specifying the original indices of variables within PSD matrix blocks.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSloadQCData` *Load quadratic program data into a model structure.*

Description

R interface function for LINDO API function `LSloadQCData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadQCData(model,nQCnnz,paiQCrows,paiQCcols1,paiQCcols2,padQCcoef)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQCnnz`	The total number of nonzeros in quadratic coefficient matrices.
`paiQCrows`	An integer array containing the index of the constraint associated with each nonzero quadratic term.
`paiQCcols1`	An integer array containing the index of the first variable defining each quadratic term.
`paiQCcols2`	An integer array containing the index of the second variable defining each quadratic term.
`padQCcoef`	A double array containing the nonzero coefficients in the quadratic matrix.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSloadSampleSizes` *Load sample sizes per stage for the stochastic model.*

Description

R interface function for LINDO API function `LSloadSampleSizes`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadSampleSizes(model,panSampleSize)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`panSampleSize`	An integer array containing the stage sample sizes.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSloadSemiContData` *Load semi-continuous data into a model structure.*

Description

R interface function for LINDO API function `LSloadSemiContData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadSemiContData(model,nSCVars,paiVars,padL,padU)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nSCVars`	The number of semi-continuous variables.
`paiVars`	A integer array containing the indices of semicontinuous variables.
`padL`	An double array containing the lower bound associated with each semi-continuous variable.
`padU`	An double array containing the upper bound associated with each semi-continuous variable.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadSETSData *Load special sets data into a model structure.*

Description

R interface function for LINDO API function `LSloadSETSData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadSETSData(model,nSETS,pszSETStype,paiCARDnum,paiSETSbegcol,paiSETScols)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nSETS`	Number of sets to load.
`pszSETStype`	A character array containing the type of each set. Valid values for each cone are `LS_MIP_SET_CARD`, `LS_MIP_SET_SOS1`, `LS_MIP_SET_SOS2`, `LS_MIP_SET_SOS3`.
`paiCARDnum`	An integer array containing set cardinalities.
`paiSETSbegcol`	An integer array containing the index of the first variable in each set.
`paiSETScols`	An integer array containing the indices of variables in each set.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSloadStageData` *Load stage structure for the model.*

Description

R interface function for LINDO API function `LSloadStageData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadStageData(model,numStages,panRstage,panCstage)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`numStages`	Number of stages in the model.
`panRstage`	An integer array containing information about the stage membership of the constraints.
`panCstage`	An integer array containing information about the stage membership of the variables.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSloadStocParData` *Load stage structure of the stochastic parameters (SPARs) in the model.*

Description

R interface function for LINDO API function `LSloadStocParData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadStocParData(model,panSparStage,padSparValue)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`panSparStage`	An integer array containing the stages of SPARs.
`padSparValue`	A double array containing default values of SPARs.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadStocParNames *Load name data for stochastic parameters into the specified model structure.*

Description

R interface function for LINDO API function `LSloadStocParNames`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadStocParNames(model,nSvars,paszSVarNames)
```

Arguments

model	A LINDO API model object, returned by `rLScreateModel`.
nSvars	Number of stochastic parameters.
paszSVarNames	A string array containing stochastic parameter names.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadString *Load a single string into a model structure.*

Description

R interface function for LINDO API function `LSloadString`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadString(model,pszString)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszString`	A null terminated string.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSloadStringData *Load a vector of strings into a model structure and get sort order.*

Description

R interface function for LINDO API function `LSloadStringData`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadStringData(model,nStrings,paszStringData)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nStrings`	Number of strings to load.
`paszStringData`	A string array containing the strings.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadVariableStages *Load stage structure of the variables in the model.*

Description

R interface function for LINDO API function `LSloadVariableStages`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadVariableStages(model,panStage)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`panStage`	An integer array containing information about the stage membership of the variables.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadVarPriorities *Provide priorities for each variable for use by mixed-integer and global solvers.*

Description

R interface function for LINDO API function `LSloadVarPriorities`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadVarPriorities(model,panCprior)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`panCprior`	An integer array containing the priority of each column in the given model. A valid priority value is any nonnegative integer value. Variables with higher priority values are given higher branching priority.

Value

An R list object with components:

ErrorCode Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadVarStartPoint — *Provide an initial starting point for nonlinear and mixed-integer solvers.*

Description

R interface function for LINDO API function `LSloadVarStartPoint`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadVarStartPoint(model,padPrimal)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
padPrimal	A double array containing starting values for each variable in the given model.

Value

An R list object with components:

ErrorCode Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSloadVarStartPointPartial — *Load a partial initial point for NLP models.*

Description

R interface function for LINDO API function `LSloadVarStartPointPartial`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadVarStartPointPartial(model,nCols,paiCols,padPrimal)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nCols	Number of variables in the partial solution.
paiCols	An integer array containing the indicies of variables in the partial solution.
padPrimal	A double array containing starting values for each variable in the given model.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSloadVarStartPoint

---

rLSloadVarType — *Load variable type data into a model structure.*

Description

R interface function for LINDO API function `LSloadVarType`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSloadVarType(model,spszVarTypes)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
spszVarTypes	A character array containing the type of each variable. Valid values for each variable are 'C', 'B', or 'I' for continuous, binary, or general integer, respectively.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSmodifyAj *Modify the coefficients for a given column at specified constraints.*

Description

R interface function for LINDO API function `LSmodifyAj`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyAj(model,iVar1,nRows,paiRows,padAj)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iVar1`	Index of the variable to modify the constraint coefficients.
`nRows`	Number of constraints to modify.
`paiRows`	An integer array containing the indices of the constraints to modify.
`padAj`	A double array containing the values of the new coefficients.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSmodifyCone *Modify the data for the specified cone.*

Description

R interface function for LINDO API function `LSmodifyCone`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyCone(model,cConeType,iConeNum,iConeNnz,paiConeCols)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`cConeType`	New type of the cone.
`iConeNum`	Index of the cone to modify.
`iConeNnz`	Number of variables characterizing the cone.
`paiConeCols`	An integer array containing the indices of the variables characterizing the cone.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSmodifyConstraintType

Modify the type or direction of a set of constraints.

Description

R interface function for LINDO API function `LSmodifyConstraintType`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyConstraintType(model,nCons,paiCons,pszConTypes)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nCons	Number of constraint to modify.
paiCons	An integer array containing the indices of the constraints to modify.
pszConTypes	A character array in which each element is either:'L', 'E', 'G' or 'N' indicating each constraint's type.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSmodifyLowerBounds` *Modify selected lower bounds in a given model.*

Description

R interface function for LINDO API function `LSmodifyLowerBounds`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyLowerBounds(model,nVars,paiVars,padL)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nVars`	Number of bounds in the model to modify.
`paiVars`	An integer array containing the indices of the variables for which to modify the lower bounds.
`padL`	A double array containing the new values of the lower bounds on the variables.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSmodifyObjective` *Modify selected objective coefficients of a given model.*

Description

R interface function for LINDO API function `LSmodifyObjective`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyObjective(model,nVars,paiVars,padC)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nVars`	Number of objective coefficients to modify.
`paiVars`	An integer array containing the indices of the objective coefficients to modify.
`padC`	A double array containing the new values for the modified objective coefficients.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSmodifyRHS` *Modify selected constraint right-hand sides of a given model.*

Description

R interface function for LINDO API function `LSmodifyRHS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyRHS(model,nCons,paiCons,padB)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nCons`	Number of constraint right-hand sides to modify.
`paiCons`	An integer array containing the indices of the constraints whose right-hand sides are to be modified.
`padB`	A double array containing the new right-hand side values for the modified right-hand sides.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSmodifySemiContVars — *Modify data of a set of semi-continuous variables in the given model.*

Description

R interface function for LINDO API function `LSmodifySemiContVars`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifySemiContVars(model,nSCVars,paiSCVars,padL,padU)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nSCVars`	Number of semi-continuous variables to modify.
`paiSCVars`	An integer array containing the indices of the variables whose data are to be modified.
`padL`	A double array containing the new lower bound values for the semi-continuous variables.
`padU`	A double array containing the new upper bound values for the semi-continuous variables.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSmodifySET — *Modify the set for the specified cone.*

Description

R interface function for LINDO API function `LSmodifySET`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifySET(model,cSETtype,iSETnum,iSETnnz,paiSETcols)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
cSETtype	New type of the set.
iSETnum	Index of the set to modify.
iSETnnz	Number of variables in the set.
paiSETcols	An integer array containing the indices of the variables in the set.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSmodifyUpperBounds *Modify selected upper bounds in a given model.*

Description

R interface function for LINDO API function `LSmodifyUpperBounds`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyUpperBounds(model,nVars,paiVars,padU)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nVars	Number of bounds in the model to modify.
paiVars	An integer array containing the indices of the variables for which to modify the upper bounds.
padU	A double array containing the new values of the upper bounds on the variables.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSmodifyVariableType *Modify the types of the variables of the given model.*

Description

R interface function for LINDO API function `LSmodifyVariableType`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSmodifyVariableType(model,nVars,paiVars,pszVarTypes)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nVars`	Number of variables to modify.
`paiVars`	An integer array containing the indices of the variables to modify.
`pszVarTypes`	A character array in which each element is either:'C', 'B', or 'I'indicating each variable's type.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSoptimize *Optimize a continuous model by a given method.*

Description

R interface function for LINDO API function `LSoptimize`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSoptimize(model,nMethod)
```

Arguments

- `model` A LINDO API model object, returned by `rLScreateModel`.
- `nMethod` A parameter indicating the solver to be used in optimizing the problem. Current options for this parameter are:
  - LS_METHOD_FREE
  - LS_METHOD_PSIMPLEX
  - LS_METHOD_DSIMPLEX
  - LS_METHOD_BARRIER
  - LS_METHOD_NLP

Value

An R list object with components:

- `ErrorCode` Zero if successful, nonzero otherwise.
- `pnStatus` The status of the optimization.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSoptimizeQP *Optimize a quadratic model with the best suitable solver.*

Description

R interface function for LINDO API function `LSoptimizeQP`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSoptimizeQP(model)
```

Arguments

- `model` A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

- `ErrorCode` Zero if successful, nonzero otherwise.
- `pnStatus` The status of the optimization.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSparam` *LINDO API Parameters.*

Description

For more information, please refer to LINDO API User Manual.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadBasis` *Read an initial basis from the given file in the specified format.*

Description

R interface function for LINDO API function `LSreadBasis`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadBasis(model,pszFname,nFormat)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the basis file.
`nFormat`	An integer parameter indicating the format of the file to be read.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadEnvParameter` *Reads environment parameters from a parameter file.*

Description

R interface function for LINDO API function `LSreadEnvParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadEnvParameter(env,pszFname)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`pszFname`	The path and name of the file from which parameters will be read.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadLINDOFile` *Read the model in LINDO format from the given file and stores the problem data in the given model structure.*

Description

R interface function for LINDO API function `LSreadLINDOFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadLINDOFile(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPS file to which the model should be written.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSreadMPSFile`

`rLSreadLINDOStream` *Read from character stream in "LINDO LTX Format".*

Description

R interface function for LINDO API function `LSreadLINDOStream`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadLINDOStream(model,pszStream,nStreamLen)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszStream`	A stream of chars constituting the full model.
`nStreamLen`	Length of the stream.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadLPFile` *Read from file stream in "CPLEX LP Format".*

Description

R interface function for LINDO API function `LSreadLPFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadLPFile(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the LP file.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadLPStream` *Read from character stream in "CPLEX LP Format".*

Description

R interface function for LINDO API function `LSreadLPStream`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadLPStream(model,pszStream,nStreamLen)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszStream`	A stream of chars constituting the full model.
`nStreamLen`	Length of the stream.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSreadModelParameter *Reads model parameters from a parameter file.*

Description

R interface function for LINDO API function `LSreadModelParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadModelParameter(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the file from which parameters will be read.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSreadMPIFile *Read a model in MPI format from the given file and stores the problem data in the given problem structure.*

Description

R interface function for LINDO API function `LSreadMPIFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadMPIFile(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPI file.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSreadMPSFile

---

rLSreadMPSFile — *Read a model in MPS format from the given file and stores the problem data in the given problem structure.*

Description

R interface function for LINDO API function `LSreadMPSFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadMPSFile(model,pszFname,nFormat)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPS file.
`nFormat`	An integer indicating whether the MPS file is formatted or not.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

rLSwriteMPSFile

`rLSreadSDPAFile` *Read SDP model from an SDPA formatted file.*

Description

R interface function for LINDO API function `LSreadSDPAFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadSDPAFile(model, pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The name of the SDPA file.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadSMPIFile` *Read the CORE,TIME,STOCH files for SP models in SMPI format.*

Description

R interface function for LINDO API function `LSreadSMPIFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadSMPIFile(model,pszCorefile,pszTimefile,pszStocfile)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszCorefile`	Path and name of the CORE file in MPS format.
`pszTimefile`	Path and name of the TIME file.
`pszStocfile`	Path and name of the STOCH file.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadSMPSFile` *Read the CORE,TIME,STOCH files for SP models in SMPS format.*

Description

R interface function for LINDO API function `LSreadSMPSFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadSMPSFile(model,pszCorefile,pszTimefile,pszStocfile,nMPStype)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszCorefile`	Path and name of the CORE file in MPS format.
`pszTimefile`	Path and name of the TIME file.
`pszStocfile`	Path and name of the STOCH file.
`nMPStype`	An integer parameter indicating whether the MPS file is formatted or not. Possible values are: • LS_FORMATTED_MPS • LS_UNFORMATTED_MPS • LS_FORMATTED_MPS_COMP

Details

This subroutine is the top level input routine. It first reads a core-file in the MPS format. It then calls further subroutines to read time and stoch files

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSreadVarPriorities` *Read branching priorities of variables from a disk file.*

Description

R interface function for LINDO API function `LSreadVarPriorities`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadVarPriorities(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The name of the file from which to read the priorities.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadVarPriorities`

`rLSreadVarStartPoint` *Provide initial values for variables from a file.*

Description

R interface function for LINDO API function `LSreadVarStartPoint`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSreadVarStartPoint(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The name of the file from which to read the starting values.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSloadVarStartPoint`

`rLSresetQCShift` *Reset to zero any shift previously made to diag(Q_i).*

Description

R interface function for LINDO API function `LSresetQCShift`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSresetQCShift(model,iRow)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iRow`	An integer specifying the index of the QC row.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampCreate` *Create an instance of a sample of specified distribution.*

Description

R interface function for LINDO API function `LSsampCreate`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampCreate(env,nDistType)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nDistType`	The distribution type.

Details

Before this function is called, `rLScreateEnv` must be called to get a valid LINDO API environment object.

Value

If successful, rLSsampCreate returns a LINDO API sample object; Otherwise, it returns NULL.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampDelete` *Delete the specified sample object.*

Description

R interface function for LINDO API function `LSsampDelete`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampDelete(sample)
```

Arguments

`sample`	A LINDO API sample object, returned by `rLSsampCreate`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampEvalDistr` *Evaluate the specified function associated with given distribution at specified point.*

Description

R interface function for LINDO API function `LSsampEvalDistr`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampEvalDistr(sample,nFuncType,dXval)
```

Arguments

`sample`	A LINDO API sample object, returned by `rLSsampCreate`.
`nFuncType`	An integer specifying the function type to evaluate. Possible values are: • LS_PDF probability density function. • LS_CDF cummulative density function. • LS_CDFINV inverse of cummulative density function. • LS_PDFDIFF derivative of the probability density function.
`dXval`	A double precision value to evaluate the specified function.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdResult`	A double value to return the result.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampEvalUserDistr` *Evaluate the specified multivariate function associated with given distribution at specified point.*

Description

R interface function for LINDO API function `LSsampEvalUserDistr`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampEvalUserDistr(sample,nFuncType,padXval,nDim)
```

Arguments

`sample` A LINDO API sample object, returned by `rLSsampCreate`.

`nFuncType` An integer specifying the function type to evaluate. Possible values are:

- LS_PDF probability density function.
- LS_CDF cummulative density function.
- LS_CDFINV inverse of cummulative density function.
- LS_PDFDIFF derivative of the probability density function.
- LS_USER a user-defined function.

`padXval` A double array containing the values required to evaluate the specified function.

`nDim` An integer specifying the number of values required in the computation of the sample point.

Value

`ErrorCode` Zero if successful, nonzero otherwise.

`pdResult` A double value to return the result.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampGenerate` *Generate a sample of a given size with specified method from the underlying distribution.*

Description

R interface function for LINDO API function `LSsampGenerate`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampGenerate(sample,nMethod,nSize)
```

Arguments

`sample` A LINDO API sample object, returned by `rLSsampCreate`.

`nMethod` An integer specifying the sampling method. Possible values are:

- LS_MONTECARLO
- LS_LATINSQUARE
- LS_ANTITHETIC

`nSize` An integer specifying the sample size.

Value

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsampGetCIPoints — *Get a copy of the correlation induced sample points.*

Description

R interface function for LINDO API function `LSsampGetCIPoints`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampGetCIPoints(sample)
```

Arguments

`sample`	A LINDO API sample object, returned by `rLSsampCreate`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnSampSize`	The sample size.
`padXval`	A double array containing the sample.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsampGetDInfo — *Get double information about the sample.*

Description

R interface function for LINDO API function `LSsampGetInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampGetDInfo(sample,nQuery)
```

Arguments

`sample` — A LINDO API sample object, returned by `rLSsampCreate`.

`nQuery` — An integer specifying the information requested from the sample. Possible values are:

- LS_IINFO_DIST_TYPE
- LS_IINFO_SAMP_SIZE
- LS_DINFO_SAMP_MEAN
- LS_DINFO_SAMP_STD
- LS_DINFO_SAMP_SKEWNESS
- LS_DINFO_SAMP_KURTOSIS

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdResult`	The result.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampGetDiscretePdfTable` *Get the PDF table from a discrete distribution sample.*

Description

R interface function for LINDO API function `LSsampGetDiscretePdfTable`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampGetDiscretePdfTable(sample)
```

Arguments

`sample`	A LINDO API sample object, returned by `rLSsampCreate`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnLen`	The table length.
`padProb`	A double array containing the probabilities of outcomes.
`padVals`	A double array containing the values of outcomes.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampGetDistrParam` *Get the specified parameter of a given distribution.*

Description

R interface function for LINDO API function `LSsampGetDistrParam`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampGetDistrParam(sample,nIndex)
```

Arguments

`sample`	A LINDO API sample object, returned by `rLSsampCreate`.
`nIndex`	An integer specifying the index of the parameter.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.
`pdValue`	A double precision value specifying the parameter value.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsampGetIInfo` *Get integer information about the sample.*

Description

R interface function for LINDO API function `LSsampGetInfo`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampGetIInfo(sample,nQuery)
```

Arguments

`sample` A LINDO API sample object, returned by `rLSsampCreate`.

`nQuery` An integer specifying the information requested from the sample. Possible values are:

- LS_IINFO_DIST_TYPE
- LS_IINFO_SAMP_SIZE
- LS_DINFO_SAMP_MEAN
- LS_DINFO_SAMP_STD
- LS_DINFO_SAMP_SKEWNESS
- LS_DINFO_SAMP_KURTOSIS

Value

ErrorCode	Zero if successful, nonzero otherwise.
pnResult	The result.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsampGetPoints *Get a copy of the generated sample points.*

Description

R interface function for LINDO API function `LSsampGetPoints`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampGetPoints(sample)
```

Arguments

sample	A LINDO API sample object, returned by `rLSsampCreate`.

Value

ErrorCode	Zero if successful, nonzero otherwise.
pnSampSize	The sample size.
padXval	A double array containing the sample.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsampLoadDiscretePdfTable
Load a PDF table for a user defined discrete distribution.

Description

R interface function for LINDO API function `LSsampLoadDiscretePdfTable`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampLoadDiscretePdfTable(sample,nLen,padProb,padVals)
```

Arguments

sample	A LINDO API sample object, returned by rLSsampCreate.
nLen	An integer specifying the table length.
padProb	A double array containing the probabilities of outcomes.
padVals	A double array containing the values of outcomes.

Value

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsampLoadPoints *Load a sample of given size to the specified sample.*

Description

R interface function for LINDO API function `LSsampLoadPoints`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampLoadPoints(sample,nSampSize,padXval)
```

Arguments

sample	A LINDO API sample object, returned by rLSsampCreate.
nSampSize	The sample size.
padXval	A double array containing the sample.

Value

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsampSetDistrParam — *Set the specified parameter of the given distribution.*

Description

R interface function for LINDO API function `LSsampSetDistrParam`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampSetDistrParam(sample,nIndex,dValue)
```

Arguments

`sample`	A LINDO API sample object, returned by `rLSsampCreate`.
`nIndex`	An integer specifying the index of the parameter.
`dValue`	A double precision value specifying the parameter value.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsampSetRG — *Set a random number generator object to the specified distribution.*

Description

R interface function for LINDO API function `LSsampSetRG`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsampSetRG(sample,RG)
```

Arguments

`sample`	A LINDO API sample object, returned by `rLSsampCreate`.
`RG`	A LINDO API random generator object, returned by `rLScreateRG`.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetConstraintProperty` *Set the property of the specified constraint of the given model.*

Description

R interface function for LINDO API function `LSsetConstraintProperty`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetConstraintProperty(model,ndxCons,nConptype)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`ndxCons`	The index of the constraint for which the property is requested.
`nConptype`	An integer macro to specify the constraint property. Possible values are: • `LS_PROPERTY_UNKNOWN` • `LS_PROPERTY_LINEAR` • `LS_PROPERTY_CONVEX` • `LS_PROPERTY_CONCAVE` • `LS_PROPERTY_QUASI_CONVEX` • `LS_PROPERTY_QUASI_CONCAVE` • `LS_PROPERTY_MAX`

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetDistrParamRG` *Set distribution parameters.*

Description

R interface function for LINDO API function `LSsetDistrParamRG`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetDistrParamRG(rg,iParam,dParam)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.
`iParam`	A parameter index.
`dParam`	A parameter value.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetDistrRG` *Set a distribution function for the random generator.*

Description

R interface function for LINDO API function `LSsetDistrRG`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetDistrRG(rg,nDistType)
```

Arguments

`rg`	A LINDO API random generator object, returned by `rLScreateRG`.
`nDistType`	An integer specifying the distribution type.

Value

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSsampCreate`

`rLSsetEnvDouParameter` *Set a double precision parameter for a specified environment.*

Description

R interface function for LINDO API function `LSsetEnvDouParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetEnvDouParameter(env,nParameter,dValue)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nParameter`	An integer referring to a double precision parameter.
`dValue`	A double precision value.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSsetModelDouParameter`

`rLSsetEnvIntParameter` *Set an integer parameter for a specified environment.*

Description

R interface function for LINDO API function `LSsetEnvIntParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetEnvIntParameter(env,nParameter,nValue)
```

Arguments

`env`	A LINDO API environment object, returned by `rLScreateEnv`.
`nParameter`	An integer referring to an integer precision parameter.
`nValue`	An integer precision value.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSsetModelIntParameter`

`rLSsetEnvStocParameterChar` *Set a stochastic parameter value of type string.*

Description

R interface function for LINDO API function `LSsetEnvStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetEnvStocParameterChar(env,nQuery,pacResult)
```

Arguments

`env`	A LINDO API env object, returned by `rLScreateEnv`.
`nQuery`	A valid query macro.
`pacResult`	A character string.

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetEnvStocParameterDou`
Set a stochastic parameter value of type double.

Description

R interface function for LINDO API function `LSsetEnvStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetEnvStocParameterDou(env,nQuery,pdResult)
```

Arguments

`env`	A LINDO API env object, returned by `rLScreateEnv`.
`nQuery`	A valid query macro.
`pdResult`	A double number.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetEnvStocParameterInt`
Set a stochastic parameter value of type integer.

Description

R interface function for LINDO API function `LSsetEnvStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetEnvStocParameterInt(env,nQuery,pnResult)
```

Arguments

`env`	A LINDO API env object, returned by `rLScreateEnv`.
`nQuery`	A valid query macro.
`pnResult`	An integer number.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetModelDouParameter` *Set a double precision parameter for a specified model.*

Description

R interface function for LINDO API function `LSsetModelDouParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetModelDouParameter(model,nParameter,dValue)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nParameter`	An integer referring to a double precision parameter.
`dValue`	A double precision value.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsetModelIntParameter *Set an integer parameter for a specified model.*

Description

R interface function for LINDO API function `LSsetModelIntParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetModelIntParameter(model,nParameter,nValue)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nParameter`	An integer referring to an integer precision parameter.
`nValue`	An integer precision value.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSsetModelDouParameter`

---

rLSsetModelStocDouParameter *Set a double valued parameter for the given model.*

Description

R interface function for LINDO API function `LSsetModelStocDouParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetModelStocDouParameter(model,iPar,dVal)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
iPar	A valid parameter macro.
dVal	A double variable of the appropriate type.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsetModelStocIntParameter

Set an integer valued parameter for the given model.

Description

R interface function for LINDO API function LSsetModelStocIntParameter. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetModelStocIntParameter(model,iPar,iVal)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
iPar	A valid parameter macro.
iVal	An integer variable of the appropriate type.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetModelStocParameterChar` *Set a stochastic parameter value of type double.*

Description

R interface function for LINDO API function `LSsetModelStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetModelStocParameterChar(model,nQuery,pacResult)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid query macro.
`pacResult`	A character string.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetModelStocParameterDou` *Set a stochastic parameter value of type double.*

Description

R interface function for LINDO API function `LSsetModelStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetModelStocParameterDou(model,nQuery,pdResult)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nQuery`	A valid query macro.
`pdResult`	A double number.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsetModelStocParameterInt
Set a stochastic parameter value of type integer.

Description

R interface function for LINDO API function `LSsetModelStocParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetModelStocParameterInt(model,nQuery,pnResult)
```

Arguments

model	A LINDO API model object, returned by `rLScreateModel`.
nQuery	A valid query macro.
pnResult	An integer number.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsetNumStages — *Set number of stages in the model.*

Description

R interface function for LINDO API function `LSsetNumStages`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetNumStages(model,numStages)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`numStages`	Number of stages in the model.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsetPrintLogNull — *Disable the printing log function.*

Description

This function is R interface specific.

Usage

```
rLSsetPrintLogNull(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetProbAllocSizes` *Increase the allocated sizes.*

Description

R interface function for LINDO API function `LSsetProbAllocSizes`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetProbAllocSizes(model,n_vars_alloc,n_cons_alloc,n_QC_alloc,n_Annz_alloc,
                     n_Qnnz_alloc,n_NLPnnz_alloc)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`n_vars_alloc`	New size for variables.
`n_cons_alloc`	New size for constraints.
`n_QC_alloc`	New size for QC terms.
`n_Annz_alloc`	New size for non-zeros.
`n_Qnnz_alloc`	New size for quadratic non-zeros.
`n_NLPnnz_alloc`	New size for NLP non-zeros.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetProbNameAllocSizes` *Increase the allocated sizes for how much space is needed for storing names.*

Description

R interface function for LINDO API function `LSsetProbNameAllocSizes`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetProbNameAllocSizes(model,n_varname_alloc,n_rowname_alloc)
```

Arguments

model — A LINDO API model object, returned by rLScreateModel.

n_varname_alloc — New size for variable names.

n_rowname_alloc — New size for constraint names.

Value

An R list object with components:

ErrorCode — Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsetRGSeed — *Set an initialization seed for the random number generator.*

Description

R interface function for LINDO API function LSsetRGSeed. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetRGSeed(rg,nSeed)
```

Arguments

rg — A LINDO API random generator object, returned by rLScreateRG.

nSeed — An integer specifying the seed to set.

Value

ErrorCode — Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsetStocParRG` *Set an RG object to the specified stochastic parameter.*

Description

R interface function for LINDO API function `LSsetStocParRG`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsetStocParRG(model,iStv,iRow,jCol,pRG)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`iStv`	Index of stochastic parameter in the instruction list.
`iRow`	Row index of the stochastic parameter.
`jCol`	Column index of the stochastic parameter.
`pRG`	A LINDO API RG object, returned by `rLScreateRG`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsolveFileLP` *Optimizes a large LP from an MPS file.*

Description

R interface function for LINDO API function `LSsolveFileLP`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsolveFileLP(model,szFileNameMPS,szFileNameSol,nNoOfColsEvaluatedPerSet,
               nNoOfColsSelectedPerSet,nTimeLimitSec)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`szFileNameMPS`	The path and name of the input MPS file.
`szFileNameSol`	The path and name of the output solution file.
`nNoOfColsEvaluatedPerSet`	Number of columns evaluated together in one set.
`nNoOfColsSelectedPerSet`	Number of columns selected from one set.
`nTimeLimitSec`	Time limit for the program in seconds.

Details

This routine is appropriate only for LP models with many more columns, e.g., millions, than rows. It is appropriate for LP's that might otherwise not easily fit into available memory.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnSolStatusParam`	The status of the optimization.
`pnNoOfConsMps`	Number of constraints in the problem.
`pnNoOfColsMps`	Number of variables (columns) in the problem.
`pnErrorLine`	Line number at which a structural error was found.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSsolveGOP` *Optimize a global optimization problem.*

Description

R interface function for LINDO API function `LSsolveGOP`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsolveGOP(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnStatus	The status of the optimization.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsolveHS *Solve the given model heuristically using the specified search method.*

Description

R interface function for LINDO API function LSsolveHS. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsolveHS(model,nSearchMethod)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nSearchMethod	An integer macro specifying the heuristic search method.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnStatus	An integer reference for the status.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsolveMIP *Optimize a mixed integer programming model using branch-and-cut.*

Description

R interface function for LINDO API function `LSsolveMIP`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsolveMIP(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnStatus`	The status of the optimization.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsolveMipBnp *Solve the MIP model with the branch-and-price method.*

Description

R interface function for LINDO API function `LSsolveMipBnp`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsolveMipBnp(model,nBlock,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`nBlock`	Number of block in the problem.
`pszFname`	An input file specifying the block structure (optional).

Details

This routine is appropriate for problems with good block structures.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnStatus	The status of the optimization.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSsolveSBD *Optimize a given LP or MILP model with Benders' decomposition.*

Description

R interface function for LINDO API function `LSsolveSBD`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsolveSBD(model,nStages,panRowStage,panColStage)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
nStages	Number of stages/blocks in the dual angular model.
panRowStage	The stage indices of constraints. Stage-0 indicates linking row or column.
panColStage	The stage indices of variables. Stage-0 indicates linking row or column.

Details

The model should have dual angular block structure to be solved with this routine. The dual angular structure is specified explicitly with the argument list.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.
pnStatus	The status of the optimization.

References

LINDO SYSTEMS home page at www.lindo.com

rLSsolveSP *Solve the SP models.*

Description

R interface function for LINDO API function `LSsolveSP`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSsolveSP(model)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.
`pnStatus`	An integer reference for the status.

References

LINDO SYSTEMS home page at www.lindo.com

rLSwriteBasis *Writes the resident basis to the given file in the specified format.*

Description

R interface function for LINDO API function `LSwriteBasis`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteBasis(model,pszFname,nFormat)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the basis file.
`nFormat`	An integer parameter indicating the format of the file to be written.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSwriteDeteqLINDOFile` *Write the deterministic equivalent (DEQ) of the SP models in LINDO format.*

Description

R interface function for LINDO API function `LSwriteDeteqLINDOFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteDeteqLINDOFile(model,pszFilename,iType)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFilename`	Path and name of the MPS file.
`iType`	Type of the the deterministic equivalent. Possible values are: • LS_DETEQ_IMPLICIT • LS_DETEQ_EXPLICIT (default)

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteDeteqMPSFile`

rLSwriteDeteqMPSFile *Write the deterministic equivalent for the SP model in MPS format.*

Description

R interface function for LINDO API function `LSwriteDeteqMPSFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteDeteqMPSFile(model,pszFilename,nFormat,iType)
```

Arguments

`model` A LINDO API model object, returned by `rLScreateModel`.

`pszFilename` Path and name of the MPS file.

`nFormat` An integer parameter indicating whether the MPS file is formatted or not. Possible values are:

- LS_FORMATTED_MPS
- LS_UNFORMATTED_MPS
- LS_FORMATTED_MPS_COMP

`iType` An integer specifying the type of the deterministic equivalent.Possible values are:

- LS_DETEQ_IMPLICIT
- LS_DETEQ_EXPLICIT (default)

Value

An R list object with components:

`ErrorCode` Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSwriteDualMPSFile *Writes the given problem to a specified file in MPS format.*

Description

R interface function for LINDO API function `LSwriteDualMPSFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteDualMPSFile(model,pszFname,nFormat,nObjSense)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPS file to which the dual model should be written.
`nFormat`	An integer indicating whether the MPS file is formatted or not.
`nObjSense`	An integer specifying if the dual problem will be posed as a maximization or minimization problem.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteMPSFile`

`rLSwriteIIS` *Writes the IIS of an infeasible LP to a file in LINDO file format.*

Description

R interface function for LINDO API function `LSwriteIIS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteIIS(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the file to which the IIS should be written.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteSolution`

rLSwriteIUS *Writes the IUS of an unbounded LP to a file in LINDO file format.*

Description

R interface function for LINDO API function `LSwriteIUS`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteIUS(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the file to which the IUS should be written.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteIIS`

rLSwriteLINDOFile *Writes the given problem to a file in LINDO format.*

Description

R interface function for LINDO API function `LSwriteLINDOFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteLINDOFile(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPS file to which the model should be written.

Details

Model must be linear.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteMPSFile`

`rLSwriteLINGOFile` *Writes the given problem to a file in LINGO format.*

Description

R interface function for LINDO API function `LSwriteLINGOFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteLINGOFile(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPS file to which the model should be written.

Details

Model must be linear.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteMPSFile`

rLSwriteModelParameter

Writes model parameters to a parameter file.

Description

R interface function for LINDO API function `LSwriteModelParameter`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteModelParameter(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the file from which parameters will be read.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

rLSwriteMPIFile

Writes the given problem to a specified file in MPI format.

Description

R interface function for LINDO API function `LSwriteMPIFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteMPIFile(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPI file to which the model should be written.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteMPSFile`

`rLSwriteMPSFile` *Writes the given problem to a specified file in MPS format.*

Description

R interface function for LINDO API function `LSwriteMPSFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteMPSFile(model,pszFname,nFormat)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPS file to which the model should be written.
`nFormat`	An integer indicating whether the MPS file is formatted or not.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSreadMPSFile`

`rLSwriteNodeSolutionFile` *Write the node solution to a file.*

Description

R interface function for LINDO API function `LSwriteNodeSolutionFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteNodeSolutionFile(model,jScenario,iStage,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	The scenario number the node belongs to.
`iStage`	The stage the node belongs to.
`pszFname`	Path and name of the file.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSwriteScenarioLINDOFile` *Write scenario model in LINDO format.*

Description

R interface function for LINDO API function `LSwriteScenarioLINDOFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteScenarioLINDOFile(model,jScenario,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	The scenario to write in LINDO format.
`pszFname`	Path and name of the LINDO file.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSwriteScenarioMPIFile

Write scenario model in MPI format.

Description

R interface function for LINDO API function `LSwriteScenarioMPIFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteScenarioMPIFile(model,jScenario,pszFname)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
jScenario	The scenario to write in MPI format.
pszFname	Path and name of the MPI file.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSwriteScenarioMPSFile

Write scenario model in MPS format.

Description

R interface function for LINDO API function `LSwriteScenarioMPSFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteScenarioMPSFile(model,jScenario,pszFname,nFormat)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`jScenario`	The scenario to write in MPS format.
`pszFname`	Path and name of the MPS file.
`nFormat`	An integer parameter indicating whether the MPS file is formatted or not. Possible values are: • LS_FORMATTED_MPS • LS_UNFORMATTED_MPS • LS_FORMATTED_MPS_COMP

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSwriteScenarioSolutionFile

Write the scenario solution to a file.

Description

R interface function for LINDO API function `LSwriteScenarioSolutionFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteScenarioSolutionFile(model,jScenario,pszFname)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
jScenario	The scenario to write the solution for.
pszFname	Path and name of the file.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

---

rLSwriteSMPIFile *Write the CORE,TIME,STOCH files for SP models in SMPI format.*

Description

R interface function for LINDO API function LSwriteSMPIFile. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteSMPIFile(model,pszCorefile,pszTimefile,pszStocfile)
```

Arguments

model	A LINDO API model object, returned by rLScreateModel.
pszCorefile	Path and name of the CORE file in MPS format.
pszTimefile	Path and name of the TIME file.
pszStocfile	Path and name of the STOCH file.

Value

An R list object with components:

ErrorCode	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSwriteSMPSFile` *Write the CORE,TIME,STOCH files for SP models in SMPS format.*

Description

R interface function for LINDO API function `LSwriteSMPSFile`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteSMPSFile(model,pszCorefile,pszTimefile,pszStocfile,nMPStype)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszCorefile`	Path and name of the CORE file in MPS format.
`pszTimefile`	Path and name of the TIME file.
`pszStocfile`	Path and name of the STOCH file.
`nMPStype`	An integer parameter indicating whether the MPS file is formatted or not. Possible values are: • LS_FORMATTED_MPS • LS_UNFORMATTED_MPS • LS_FORMATTED_MPS_COMP

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSwriteSolution` *Writes the LP solution to a file.*

Description

R interface function for LINDO API function `LSwriteSolution`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteSolution(model,pszFname)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the file to which the solution should be written.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

`rLSwriteSolutionOfType`

Write the solution of a given problem to a file in a specific format.

Description

R interface function for LINDO API function `LSwriteSolutionOfType`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteSolutionOfType(model,pszFname,nFormat)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the file to which the solution should be written.
`nFormat`	An integer specifying the file format.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteSolution`

`rLSwriteWithSetsAndSC` *Writes the given problem to a specified file in MPS format with the sets and SC vars inserted.*

Description

R interface function for LINDO API function `LSwriteWithSetsAndSC`. For more information, please refer to LINDO API User Manual.

Usage

```
rLSwriteWithSetsAndSC(model,pszFname,nFormat)
```

Arguments

`model`	A LINDO API model object, returned by `rLScreateModel`.
`pszFname`	The path and name of the MPS file to which the model should be written.
`nFormat`	An integer indicating whether the MPS file is formatted or not.

Value

An R list object with components:

`ErrorCode`	Zero if successful, nonzero otherwise.

References

LINDO SYSTEMS home page at www.lindo.com

See Also

`rLSwriteMPSFile`

Index