Package ‘cplexAPI’

January 31, 2017

Type Package
Title R Interface to C API of IBM ILOG CPLEX
Version 1.3.3
Date 2017-01-18
Depends R (>= 2.6.0)
Imports methods
Description This is the R Interface to the C API of IBM ILOG CPLEX. It necessarily depends on IBM ILOG CPLEX (>= 12.1).
SystemRequirements IBM ILOG CPLEX (>= 12.1)
License GPL-3 | file LICENSE
LazyLoad yes
Collate generics.R cplexConst.R cplexErrorClass.R cplexPtrClass.R
cplex.R cplexAPI.R cplex_checkAPI.R cplex_longparamAPI.R zzz.R
Author C. Jonathan Fritzemeier [cre, ctb], Gabriel Gelius-Dietrich [aut]
Maintainer C. Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
NeedsCompilation yes
Repository CRAN
Date/Publication 2017-01-31 08:14:41

R topics documented:
cplexAPI-package .............................................. 6
addColsCPLEX .............................................. 7
addFpDestCPLEX ........................................... 9
addIndConstrCPLEX ....................................... 10
addMIPstartsCPLEX ........................................ 11
addQConstrCPLEX .......................................... 12
addRowsCPLEX ............................................. 13
baroptCPLEX ................................................. 14
baseWriteCPLEX ............................................ 15
basicPresolveCPLEX ....................................... 16
boundSaCPLEX .................................................. 17
checkAddColsCPLEX ............................................. 18
checkAddRowsCPLEX ............................................. 19
checkChgCoefListCPLEX ........................................ 20
checkCopyColTypeCPLEX ......................................... 21
checkCopyLpCPLEX ............................................... 22
checkCopyLpwNamesCPLEX ....................................... 23
checkCopyQPsepCPLEX .......................................... 25
checkCopyQuadCPLEX ........................................... 26
checkValsCPLEX ............................................... 27
chgBndsCPLEX .................................................. 28
chgCoefCPLEX .................................................. 29
chgCoefListCPLEX ............................................... 30
chgColNameCPLEX ............................................... 31
chgColsBndsCPLEX .............................................. 32
cchgColTypeCPLEX .............................................. 33
cchgMIPstartsCPLEX ............................................ 34
chgNameCPLEX .................................................. 35
cchgObjCPLEX ................................................... 36
cchgProbNameCPLEX ............................................ 37
cchgProbTypeCPLEX ............................................. 38
cchgQPcoeffCPLEX .............................................. 39
cchgRhsCPLEX .................................................. 40
cchgRngValCPLEX ............................................... 41
cchgRowNameCPLEX ............................................. 42
cchgSenseCPLEX ................................................ 43
cchgTerminateCPLEX .......................................... 44
cleanupCoeffCPLEX .............................................. 44
ccloneProbCPLEX ................................................. 45
clcloseEnvCPLEX ............................................... 46
cCloseFileCPLEX ............................................... 47
cCloseProbCPLEX ............................................... 48
cLpWriteCPLEX .................................................. 49
completeLpCPLEX ............................................... 50
copyBaseCPLEX .................................................. 51
copyColTypeCPLEX ............................................. 52
copyLpCPLEX .................................................... 53
copyLpwNamesCPLEX ........................................... 54
copyObjNameCPLEX ............................................ 55
copyOrderCPLEX ................................................ 56
copyPartBaseCPLEX ............................................ 57
copyQPsepCPLEX ............................................... 58
copyQuadCPLEX ............................................... 59
copyStartCPLEX ............................................... 60
cplexConstants .................................................. 61
cplexError-class .............................................. 82
cplexPtr-class .................................................. 83
delColsCPLEX ................................................... 84
<table>
<thead>
<tr>
<th>R topics documented:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>delFpDestCPLEX</td>
<td>85</td>
</tr>
<tr>
<td>delIndConstrsCPLEX</td>
<td>86</td>
</tr>
<tr>
<td>delMIPstartsCPLEX</td>
<td>87</td>
</tr>
<tr>
<td>delNamesCPLEX</td>
<td>88</td>
</tr>
<tr>
<td>delProbCPLEX</td>
<td>89</td>
</tr>
<tr>
<td>delQConstrsCPLEX</td>
<td>90</td>
</tr>
<tr>
<td>delRowsCPLEX</td>
<td>91</td>
</tr>
<tr>
<td>delSetColsCPLEX</td>
<td>92</td>
</tr>
<tr>
<td>delSetRowsCPLEX</td>
<td>93</td>
</tr>
<tr>
<td>delTerminateCPLEX</td>
<td>94</td>
</tr>
<tr>
<td>disconnectChannelCPLEX</td>
<td>95</td>
</tr>
<tr>
<td>dualoptCPLEX</td>
<td>96</td>
</tr>
<tr>
<td>dualWriteCPLEX</td>
<td>97</td>
</tr>
<tr>
<td>feasOptCPLEX</td>
<td>98</td>
</tr>
<tr>
<td>fileputCPLEX</td>
<td>99</td>
</tr>
<tr>
<td>flushChannelCPLEX</td>
<td>100</td>
</tr>
<tr>
<td>flushStdChannelsCPLEX</td>
<td>101</td>
</tr>
<tr>
<td>freePresolveCPLEX</td>
<td>102</td>
</tr>
<tr>
<td>getBaseCPLEX</td>
<td>103</td>
</tr>
<tr>
<td>getBestObjValCPLEX</td>
<td>104</td>
</tr>
<tr>
<td>getChannelsCPLEX</td>
<td>105</td>
</tr>
<tr>
<td>getChgParmCPLEX</td>
<td>106</td>
</tr>
<tr>
<td>getCoefCPLEX</td>
<td>107</td>
</tr>
<tr>
<td>getColIndexCPLEX</td>
<td>108</td>
</tr>
<tr>
<td>getColInfeasCPLEX</td>
<td>109</td>
</tr>
<tr>
<td>getColNameCPLEX</td>
<td>110</td>
</tr>
<tr>
<td>getColsCPLEX</td>
<td>111</td>
</tr>
<tr>
<td>getColTypeCPLEX</td>
<td>112</td>
</tr>
<tr>
<td>getConflictCPLEX</td>
<td>113</td>
</tr>
<tr>
<td>getConflictExtCPLEX</td>
<td>114</td>
</tr>
<tr>
<td>getCutoffCPLEX</td>
<td>115</td>
</tr>
<tr>
<td>getDblParmCPLEX</td>
<td>116</td>
</tr>
<tr>
<td>getDblQualCPLEX</td>
<td>117</td>
</tr>
<tr>
<td>getDbsCntCPLEX</td>
<td>118</td>
</tr>
<tr>
<td>getDjCPLEX</td>
<td>119</td>
</tr>
<tr>
<td>getErrorStrCPLEX</td>
<td>120</td>
</tr>
<tr>
<td>getGradCPLEX</td>
<td>121</td>
</tr>
<tr>
<td>getIndConstrCPLEX</td>
<td>122</td>
</tr>
<tr>
<td>getInfoDblParmCPLEX</td>
<td>123</td>
</tr>
<tr>
<td>getInfoIntParmCPLEX</td>
<td>124</td>
</tr>
<tr>
<td>getInfoLongParmCPLEX</td>
<td>125</td>
</tr>
<tr>
<td>getInfoStrParmCPLEX</td>
<td>126</td>
</tr>
<tr>
<td>getIntParmCPLEX</td>
<td>127</td>
</tr>
<tr>
<td>getIntQualCPLEX</td>
<td>128</td>
</tr>
<tr>
<td>getItCntCPLEX</td>
<td>129</td>
</tr>
<tr>
<td>getLogFileCPLEX</td>
<td>130</td>
</tr>
<tr>
<td>getLongParmCPLEX</td>
<td>131</td>
</tr>
<tr>
<td>getLowBndsIdsCPLEX</td>
<td>132</td>
</tr>
<tr>
<td>Topic</td>
<td>Page</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>getLowerBndsCPLEX</td>
<td>133</td>
</tr>
<tr>
<td>getMethodCPLEX</td>
<td>134</td>
</tr>
<tr>
<td>getMIPrelGapCPLEX</td>
<td>135</td>
</tr>
<tr>
<td>getMIPstartIndexCPLEX</td>
<td>136</td>
</tr>
<tr>
<td>getMIPstartNameCPLEX</td>
<td>137</td>
</tr>
<tr>
<td>getMIPstartsCPLEX</td>
<td>138</td>
</tr>
<tr>
<td>getNumColsCPLEX</td>
<td>139</td>
</tr>
<tr>
<td>getNumMIPstartsCPLEX</td>
<td>140</td>
</tr>
<tr>
<td>getNumNnzCPLEX</td>
<td>141</td>
</tr>
<tr>
<td>getNumQConstrsCPLEX</td>
<td>142</td>
</tr>
<tr>
<td>getNumQPnzCPLEX</td>
<td>143</td>
</tr>
<tr>
<td>getNumQuadCPLEX</td>
<td>144</td>
</tr>
<tr>
<td>getNumRowsCPLEX</td>
<td>145</td>
</tr>
<tr>
<td>getObjCPLEX</td>
<td>146</td>
</tr>
<tr>
<td>getObjDirCPLEX</td>
<td>147</td>
</tr>
<tr>
<td>getObjNameCPLEX</td>
<td>148</td>
</tr>
<tr>
<td>getObjOffsetCPLEX</td>
<td>149</td>
</tr>
<tr>
<td>getObjValCPLEX</td>
<td>150</td>
</tr>
<tr>
<td>getOrderCPLEX</td>
<td>151</td>
</tr>
<tr>
<td>getParmNameCPLEX</td>
<td>152</td>
</tr>
<tr>
<td>getParmNumCPLEX</td>
<td>153</td>
</tr>
<tr>
<td>getParmTypeCPLEX</td>
<td>154</td>
</tr>
<tr>
<td>getParmValCPLEX</td>
<td>155</td>
</tr>
<tr>
<td>getPhase1CntCPLEX</td>
<td>155</td>
</tr>
<tr>
<td>getPiCPLEX</td>
<td>156</td>
</tr>
<tr>
<td>getPreStatCPLEX</td>
<td>157</td>
</tr>
<tr>
<td>getProbNameCPLEX</td>
<td>158</td>
</tr>
<tr>
<td>getProbTypeCPLEX</td>
<td>159</td>
</tr>
<tr>
<td>getProbVarCPLEX</td>
<td>160</td>
</tr>
<tr>
<td>getQConstrCPLEX</td>
<td>161</td>
</tr>
<tr>
<td>getQPcoefCPLEX</td>
<td>162</td>
</tr>
<tr>
<td>getQuadCPLEX</td>
<td>163</td>
</tr>
<tr>
<td>getRedLP_CPLEX</td>
<td>164</td>
</tr>
<tr>
<td>getRhsCPLEX</td>
<td>165</td>
</tr>
<tr>
<td>getRngValCPLEX</td>
<td>166</td>
</tr>
<tr>
<td>getRowIndexCPLEX</td>
<td>167</td>
</tr>
<tr>
<td>getRowInfeasCPLEX</td>
<td>168</td>
</tr>
<tr>
<td>getRowNameCPLEX</td>
<td>169</td>
</tr>
<tr>
<td>getRowsCPLEX</td>
<td>170</td>
</tr>
<tr>
<td>getSenseCPLEX</td>
<td>171</td>
</tr>
<tr>
<td>getSiftItCntCPLEX</td>
<td>172</td>
</tr>
<tr>
<td>getSiftPase1CntCPLEX</td>
<td>173</td>
</tr>
<tr>
<td>getSlackCPLEX</td>
<td>174</td>
</tr>
<tr>
<td>getStatCPLEX</td>
<td>175</td>
</tr>
<tr>
<td>getStatStrCPLEX</td>
<td>176</td>
</tr>
<tr>
<td>getStrParmCPLEX</td>
<td>177</td>
</tr>
<tr>
<td>getSubMethodCPLEX</td>
<td>178</td>
</tr>
<tr>
<td>getSubStatCPLEX</td>
<td>179</td>
</tr>
</tbody>
</table>
R topics documented:

- getTimeCPLEX ............................................. 180
- getUppBndsIdsCPLEX ..................................... 181
- getUpperBndsCPLEX .................................... 182
- getVersionCPLEX ........................................ 183
- hybbaroptCPLEX ........................................ 184
- hybnetoptCPLEX ......................................... 185
- initProbCPLEX ........................................... 186
- lpoptCPLEX ............................................... 187
- mipoptCPLEX ............................................. 188
- newColsCPLEX ........................................... 189
- newRowsCPLEX ........................................... 190
- objSaCPLEX .............................................. 191
- openEnvCPLEX ........................................... 192
- openFileCPLEX ........................................... 193
- openProbCPLEX .......................................... 194
- ordWriteCPLEX .......................................... 195
- preslvWriteCPLEX ........................................ 196
- presolveCPLEX .......................................... 197
- primoptCPLEX ............................................ 198
- printTerminateCPLEX ..................................... 199
- qpoptCPLEX .............................................. 199
- readCopyBaseCPLEX ....................................... 200
- readCopyMIPstartsCPLEX .................................. 201
- readCopyOrderCPLEX ...................................... 202
- readCopyParmCPLEX ...................................... 203
- readCopyProbCPLEX ...................................... 204
- readCopySolCPLEX ........................................ 205
- refineConflictCPLEX ..................................... 206
- refineConflictExtCPLEX .................................. 207
- refineMIPstartConflictCPLEX ............................. 208
- refineMIPstartConflictExtCPLEX ......................... 209
- return_codeCPLEX ....................................... 210
- rhsSaCPLEX ............................................. 211
- setDblParmCPLEX ......................................... 212
- setDefaultParmCPLEX .................................... 213
- setIntParmCPLEX ......................................... 214
- setLogFileCPLEX ......................................... 215
- setLongParmCPLEX ....................................... 216
- setObjDirCPLEX .......................................... 217
- setStrParmCPLEX ......................................... 218
- setTerminateCPLEX ....................................... 219
- siftoptCPLEX ............................................ 220
- solnInfoCPLEX ........................................... 221
- solutionCPLEX ........................................... 222
- solWriteCPLEX ........................................... 223
- status_codeCPLEX ...................................... 224
- tightenBndsCPLEX ....................................... 225
- tuneParmCPLEX .......................................... 226
cplexAPI-package

Description

A low level interface to IBM ILOG CPLEX.

Details

The package cplexAPI provides access to the callable library of IBM ILOG CPLEX from within R.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


Examples

# load package
library(cplexAPI)

# Open a CPLEX environment
env <- openEnvCplex()

# Create a problem object
prob <- initProbCplex(env)

# Assign a name to the problem object
chgProbNameCplex(env, prob, "sample")

# Prepare data structures for the problem object
# Number of columns and rows
nc <- 3
nr <- 3

# Objective function
obj <- c(5, 4, 3)
# Right hand side
rhs <- c(5, 11, 8)

# Sense of the right hand side
sense <- rep("L", 3)

# Variable lower bounds
lb <- rep(0, 3)

# Variable upper bounds
ub <- rep(CPX_INFBOUND, 3)

# Column and row names
cn <- c("x1", "x2", "x3")

rn <- c("q1", "q2", "q3")

# The constraint matrix is passed in column major order format
# Be careful here: all indices start with 0! Begin indices of rows
beg <- c(0, 3, 6)

# Number of non-zero elements per row
cnt <- rep(3, 3)

# Column indices
ind <- c(0, 1, 2, 0, 1, 2, 0, 1, 2)

# Non-zero elements
val <- c(2, 4, 3, 3, 1, 4, 1, 2, 2)

# Load problem data
copyLpwnamesCplex(env, prob, nc, nr, CPX_MAX, obj, rhs, sense,
beg, cnt, ind, val, lb, ub, NULL, cn, rn)

# Solve the problem using the simplex algorithm
lpoptCplex(env, prob)

# Retrieve solution after optimization
solutionCplex(env, prob)

# Free memory, allocated to the problem object
delProbCplex(env, prob)
closeEnvCplex(env)

---

**addColsCplex**  
Adds Columns to a Specified CPLEX Problem Object

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXaddcols. Consult the IBM ILOG CPLEX documentation for more detailed information.
addColsCPLEX

Usage

addColsCPLEX(env, lp, ncols, nnz, objf, matbeg, matind, matval, lb = NULL, ub = NULL, cnames = NULL)

Arguments

env       An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp        An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
ncols     Number of columns.
nnz       Number of nonzero constraint coefficients.
objf      Objective function coefficients.
matbeg    Array that specifies the nonzero elements of the columns being added. Consult the IBM ILOG CPLEX documentation for more detailed information.
matind    Array that specifies the nonzero elements of the columns being added. Consult the IBM ILOG CPLEX documentation for more detailed information.
matval    Array that specifies the nonzero elements of the columns being added. Consult the IBM ILOG CPLEX documentation for more detailed information.
lb        Lower bounds of the new variables.
ub        Upper bounds of the new variables.
cnames    Names of the new variables.

Details

Interface to the C function addCols which calls the CPLEX function CPXaddcols.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

checkAddColsCPLEX, addRowsCPLEX
**Description**

Low level interface function to the IBM ILOG CPLEX function `cpxaddfpdest`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```c
addFpDestCLEX(env, newch, cpfile)
```

**Arguments**

- `env`: An object of class "`cplexPtr`" as returned by `openEnvCLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `newch`: Pointer to an IBM ILOG CPLEX channel as returned by `addChannelCLEX`.
- `cpfile`: Pointer to an IBM ILOG CPLEX file as returned by `openFileCLEX`.

**Details**

Interface to the C function `addFpDest` which calls the CPLEX function `CPXaddfpdest`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

- Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
- Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

`delFpDestCLEX`
addIndConstrCPLEX  

Adds an Indicator Constraint to the Specified CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXaddindconstr. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

addIndConstrCPLEX(env, lp, indvar, complemented, nzcnt, rhs, sense, linind, linval, indname = NULL)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp   An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
indvar The binary variable that acts as the indicator for this constraint.
complemented A Boolean value that specifies whether the indicator variable is complemented.
nzcnt An integer that specifies the number of nonzero coefficients in the linear portion of the indicator constraint.
rhs The righthand side value for the linear portion of the indicator constraint.
sense The sense of the linear portion of the indicator constraint.
linind A vector that with linval defines the linear portion of the indicator constraint.
linval A vector that with linind defines the linear portion of the indicator constraint.
indname The name of the constraint to be added (optional).

Details

Interface to the C function addIndConstr which calls the CPLEX function CPXaddindconstr.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Add Multiple MIP Starts to a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXaddmipstarts. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```
addMIPstartsCplex(env, lp, mcnt, nzcnt, beg, varindices,
values, effortlevel, mipstartname = NULL)
```

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **mcnt**: Number of MIP starts to be added.
- **nzcnt**: Number of variable values to be added.
- **beg**: Array of length mcnt used with varindices and values. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **varindices**: Array of length nzcnt containing the numeric indices of the columns corresponding to the variables which are assigned starting values. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **values**: Array of length nzcnt containing the values to use for the MIP starts. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **effortlevel**: Array of length mcnt. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **mipstartname**: Names of the MIP starts.

Details

Interface to the C function addMIPstarts which calls the CPLEX function CPXaddmipstarts.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
References


addQConstrCPLEX  Add Quadratic Constraint to a Specified CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXaddqconstr. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

addQConstrCPLEX(env, lp, lzn, qzn, rhs, sense, 
                 lind = NULL, lval = NULL, 
                 qrow, qcol, qval, qname = NULL)

Arguments

env      An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp       An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
lzn      Number of nonzero constraint coefficients in the linear part of the constraint.
qzn      Number of nonzero constraint coefficients in the quadratic part of the constraint.
rhs      Righthand side term.
sense    The sense of the constraint to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
lind     Linear part of the quadratic constraint to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
lval     Linear part of the constraint to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
qrow     Quadratic part of the quadratic constraint to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
qcol     Quadratic part of the quadratic constraint to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
qval     Quadratic part of the quadratic constraint to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
qname    Name of the constraint to be added.

Details

Interface to the C function addQConstr which calls the CPLEX function CPXaddqconstr.
**addRowsCPLEX**

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


---

**addRowsCPLEX**  
*Add Constraints to a Specified CPLEX Problem Object*

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXaddrows. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
addRowsCPLEX(env, lp, ncols, nrows, nnz, matbeg, matind, matval,
            rhs = NULL, sense = NULL,
            cnames = NULL, rnames = NULL)
```

**Arguments**

- **env**: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **ncols**: Number of new columns in the constraints being added to the constraint matrix.
- **nrows**: Number of rows.
- **nnz**: Number of nonzero constraint coefficients.
- **matbeg**: An array used with rmatind and rmatval to define the rows to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **matind**: An array used with rmatind and rmatval to define the rows to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **matval**: An array used with rmatind and rmatval to define the rows to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **rhs**: Righthand side term for each constraint to be added.
- **sense**: Sense of each constraint to be added.
- **cnames**: Names of the new columns.
- **rnames**: Names of the new rows.
Details

Interface to the C function addCols which calls the CPLEX function CPXaddcols.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

checkAddRowsCplex, addColsCplex, copyLpCplex, chgRngValCplex

---

baroptCplex  
Solve LP, QP or QCP Problem by Means of the Barrier Algorithm

Description

Low level interface function to the IBM ILOG CPLEX function CPXbaropt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

baroptCplex(env, lp)

Arguments

<table>
<thead>
<tr>
<th>env</th>
<th>An object of class &quot;cplexPtr&quot; as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
</tbody>
</table>

Details

Interface to the C function baropt which calls the CPLEX function CPXbaropt.

Value

Zero if successful, otherwise nonzero.
baseWriteCplex

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

solnInfoCplex, getStatCplex, solutionCplex

---

**baseWriteCplex**

*Write the Most Current Basis Associated With a CPLEX Problem Object to a File*

**Description**

Low level interface function to the IBM ILOG CPLEX function CPLEXbasewrite. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```
baseWriteCplex(env, lp, fname)
```

**Arguments**

- **env**
  
  An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

- **lp**
  
  An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

- **fname**
  
  A filename.

**Details**

Interface to the C function baseWrite which calls the CPLEX function CPLEXbasewrite.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
basicPresolveCplex

Perform Bound Strengthening and Detect Redundant Rows

Description
Low level interface function to the IBM ILOG CPLEX function CPXbasicpresolve. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
basicPresolveCplex(env, lp)

Arguments
env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details
Interface to the C function basicPresolve which calls the CPLEX function CPXbasicpresolve.

Value
If successfull, a list will be returned:
redlb strengthened lower bounds
redub strengthened upper bounds
rstat status of the row

Otherwise an object of class "cplexError".

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References
boundSaCPLEX

Access Upper and Lower Sensitivity Ranges for Lower and Upper Variable Bounds

Description

Low level interface function to the IBM ILOG CPLEX function CPXboundsa. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

boundSaCPLEX(env, lp, begin, end)

Arguments

e env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

begin Beginning of the range of ranges to be returned.

d end End of the range of ranges to be returned.

Details

Interface to the C function boundSa which calls the CPLEX function CPXboundsa.

Value

If successfull, a list will be returned:

lb lower lower bound lower range values

lb upper lower bound upper range values

ub lower upper bound lower range values

ub upper upper bound upper range values

Otherwise an object of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

checkAddColsCPLEX  Validate Arguments of the Corresponding addColsCPLEX Routine

Description

Low level interface function to the IBM ILOG CPLEX function CPXcheckaddcols. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

checkAddColsCPLEX(env, lp, ncols, nnz, objf, matbeg, matind, matval, 
 lb = NULL, ub = NULL, cnames = NULL)

Arguments

env          An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp           An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
ncols        Number of columns.
nnz          Number of nonzero constraint coefficients.
objf         Objective function coefficients.
matbeg       Array that specifies the nonzero elements of the columns being added. Consult the IBM ILOG CPLEX documentation for more detailed information.
matind       Array that specifies the nonzero elements of the columns being added. Consult the IBM ILOG CPLEX documentation for more detailed information.
matval       Array that specifies the nonzero elements of the columns being added. Consult the IBM ILOG CPLEX documentation for more detailed information.
lb           Lower bounds of the new variables.
ub            Upper bounds of the new variables.
cnames       Names of the new variables.

Details

Interface to the C function checkAddCols which calls the CPLEX function CPXcheckaddcols.

Value

Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
References


See Also

addColsCplex

---

Description

Low level interface function to the IBM ILOG CPLEX function CPXcheckaddrows. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

checkAddRowsCplex(env, lp, ncols, nrows, nnz, matbeg, matind, matval, rhs = NULL, sense = NULL, cnames = NULL, rnames = NULL)

Arguments

env
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp
An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

ncols
Number of new columns in the constraints being added to the constraint matrix.

nrows
Number of rows.

nnz
Number of nonzero constraint coefficients.

matbeg
An array used with rmatind and rmatval to define the rows to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.

matind
An array used with rmatind and rmatval to define the rows to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.

matval
An array used with rmatind and rmatval to define the rows to be added. Consult the IBM ILOG CPLEX documentation for more detailed information.

rhs
Righthand side term for each constraint to be added.

sense
Sense of each constraint to be added.

cnames
Names of the new columns.

rnames
Names of the new rows.
checkChgCoefListCPLEX

Details
Interface to the C function checkAddRows which calls the CPLEX function CPXcheckaddrows.

Value
Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
addRowsCPLEX

Description
Low level interface function to the IBM ILOG CPLEX function CPXcheckchgcoeflist. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
checkChgCoefListCPLEX(env, lp, nnz, ia, ja, ra)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>env</td>
<td>An object of class &quot;cplexPtr&quot; as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.</td>
</tr>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
<tr>
<td>nnz</td>
<td>Number of nonzero constraint coefficients.</td>
</tr>
<tr>
<td>ia</td>
<td>Row indices of the nonzero elements.</td>
</tr>
<tr>
<td>ja</td>
<td>Column indices of the nonzero elements.</td>
</tr>
<tr>
<td>ra</td>
<td>Nonzero elements.</td>
</tr>
</tbody>
</table>
Details

Interface to the C function checkChgCoefList which calls the CPLEX function CPXcheckchgcoeflist.

Value

Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)

Gabriel Gelius-Dietrich < geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier < clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

chgCoefListCPLEX

Description

Low level interface function to the IBM ILOG CPLEX function CPXcheckcopyctype. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

checkCopyColTypeCPLEX(env, lp, xtype)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>env</td>
<td>An object of class &quot;cplexPtr&quot; as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.</td>
</tr>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
<tr>
<td>xtype</td>
<td>A vector containing the type of each column in the constraint matrix.</td>
</tr>
</tbody>
</table>

Details

Interface to the C function checkCopyColType which calls the CPLEX function CPXcheckcopyctype.
checkCopyLpCplex

Value
Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
copyColTypeCplex

checkCopyLpCPLEX Validate Arguments of the Corresponding copyLpCplex Routine

Description
Low level interface function to the IBM ILOG CPLEX function CPXcheckcopylp. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
checkCopyLpCPLEX(env, lp, ncols, nrows, lpdir, objf, rhs, sense, matbeg, matcnt, matind, matval, lb, ub, rngval = NULL)

Arguments
env An object of class "cplexPtr" as returned by openEnvCPOLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCPOLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
ncols Number of columns in the constraint matrix.
nrows Number of rows in the constraint matrix.
lpdir Single integer value that specifies whether the problem is a minimization or maximization problem.
objf The objective function coefficients.
rhs The righthand side values for each constraint in the constraint matrix.
sense The sense of each constraint in the constraint matrix.
matbeg Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
checkCopyLpwNamesCPLEX

matcnt  Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
matind  Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
matval  Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
lb      Containing the lower bound on each of the variables.
ub      Containing the lower bound on each of the variables.
rngval Containing the range value of each ranged constraint.

Details

Interface to the C function checkCopyLp which calls the CPLEX function CPXcheckcopylp.

Value

Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

copyLpCPLEX

checkCopyLpwNamesCPLEX

Validate Arguments of the Corresponding copyLpwNamesCPLEX Routine

Description

Low level interface function to the IBM ILOG CPLEX function CPXcheckcopylpwnames. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

checkCopyLpwNamesCPLEX(env, lp, nCols, nRows, lpdir, objf, rhs, sense, matbeg, matcnt, matind, matval, lb, ub, rngval = NULL, cnames = NULL, rnames = NULL)
checkCopyLpwNamesCPLEX

Arguments

env | An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp  | An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
nCols | Number of columns in the constraint matrix.
nRows | Number of rows in the constraint matrix.
lpdir | Single integer value that specifies whether the problem is a minimization or maximization problem.
objf   | The objective function coefficients.
rhs    | The righthand side values for each constraint in the constraint matrix.
sense  | The sense of each constraint in the constraint matrix.
matbeg | Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
matcnt | Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
matind | Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
matval | Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
lb       | Containing the lower bound on each of the variables.
ub       | Containing the lower bound on each of the variables.
rngval  | Containing the range value of each ranged constraint.
cnames  | Names of the matrix columns or, equivalently, the variable names.
rnames  | Names of the matrix rows or, equivalently, the constraint names.

Details

Interface to the C function checkCopyLpwNames which calls the CPLEX function CPXcheckcopylawnames.

Value

Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

checkCopyQPsepCPLEX

See Also

copyLpWNamesCplex

---

checkCopyQPsepCPLEX Validate Arguments of the Corresponding copyQPsepCplex Routine

Description

Low level interface function to the IBM ILOG CPLEX function CPXcheckcopyqpsep. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

checkCopyQPsepCPLEX(env, lp, qsepvec)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
qsepvec A vector containing the quadratic coefficients.

Details

Interface to the C function checkCopyQPsep which calls the CPLEX function CPXcheckcopyqpsep.

Value

Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

copyQPsepCplex
checkCopyQuadCplex

Validate Arguments of the Corresponding checkCopyQuadCplex Routine

Description
Low level interface function to the IBM ILOG CPLEX function CPXcheckcopyquad. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
checkCopyQuadCplex(env, lp, qmatbeg, qmatcnt, qmatind, qmatval)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **qmatbeg**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **qmatcnt**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **qmatind**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **qmatval**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.

Details
Interface to the C function checkCopyQuad which calls the CPLEX function CPXcheckcopyquad.

Value
Nonzero if it detects an error in the data; it returns zero if it does not detect any data errors.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References
checkValsCPLEX

See Also

copyQuadCPLEX

checkValsCPLEX  Check an Array of Indices and a Corresponding Array of Values for Input Errors

Description

Low level interface function to the IBM ILOG CPLEX function CPXcheckvals. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

checkValsCPLEX(env, lp, nval, rind = NULL, cind = NULL, val = NULL)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- nval: Number of values to be examined.
- rind: Row indices.
- cind: Column indices.
- val: The values itself.

Details

Interface to the C function checkVals which calls the CPLEX function CPXcheckvals.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**chgBndsCPLEX**  
*Change the Lower or Upper Bounds on a Set of Variables of a Problem*

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXchgbds. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
cchgBndsCPLEX(env, lp, ncols, ind, lu, bd)
```

**Arguments**

- `env`  
  An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp`  
  An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `ncols`  
  Number of bounds to be changed.
- `ind`  
  Indices of bounds to be changed.
- `lu`  
  A character vector, specifying whether an entry in `bd` is a upper or a lower bound on variable `ind[j]`.
- `bd`  
  Values of the lower or upper bounds of the variables present in `ind`.

**Details**

Interface to the C function `chgBnds` which calls the CPLEX function CPXchgbds.

**Value**

`Zero` if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

`getLowerBndsCPLEX`, `getUpperBndsCPLEX`
chgCoefCplex

Change a Single Coefficient in the Constraint Matrix, Linear Objective Coefficients, Righthand Side, or Ranges of a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgcoef. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgCoefCplex(env, lp, i, j, val)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
i An integer that specifies the numeric index of the row in which the coefficient is located. The linear objective row is referenced with i = -1.
j An integer that specifies the numeric index of the column in which the coefficient is located. The RHS column is referenced with j = -1. The range value column is referenced with j = -2. If j = -2 is specified and row i is not a ranged row, an error status is returned.
val The new value for the coefficient being changed.

Details

Interface to the C function chgCoef which calls the CPLEX function CPXchgcoef.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

gNumRowsCplex, gNumColsCplex, chgObjCplex, chgRhsCplex, chgRngValCplex
chgCoefListCPLEX | Change a List of Matrix Coefficients of a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgcoeflist. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgCoefListCPLEX(env, lp, nnz, ia, ja, ra)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **nnz**: Number of nonzero constraint coefficients.
- **ia**: Row indices of the nonzero elements.
- **ja**: Column indices of the nonzero elements.
- **ra**: Nonzero elements.

Details

Interface to the C function chgcoeflist which calls the CPLEX function CPXchgcoeflist.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

chgColNameCPLEX

Change the Names of Variables in a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgcolname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgColNameCPLEX(env, lp, nnames, ind, names)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- nnames: A vector that specifies the total number of variable names to be changed.
- ind: A vector containing the numeric indices indices of the variables for which the names are to be changed.
- names: A vector containing the strings of the new variable names for the columns specified in ind.

Details

Interface to the C function chgColName which calls the CPLEX function CPXchgcolname.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

chgColsBndsCplex

Change Lower and Upper Bounds on a Set of Variables of a Problem

Description

Set lower and upper bounds on a set of variables in one step. If \( lb[i] = ub[i] \) the type of the bound is set to "B", otherwise \( lb[i] \) is set to "L" and \( ub[i] \) is set to "U".

Usage

\[
\text{chgColsBndsCplex}(\text{env}, \text{lp}, j, lb, ub)
\]

Arguments

- \text{env}: An object of class "cplexPtr" as returned by \text{openEnvCplex}. This is basically a pointer to an IBM ILOG CPLEX environment.
- \text{lp}: An object of class "cplexPtr" as returned by \text{initProbCplex}. This is basically a pointer to an IBM ILOG CPLEX problem object.
- \text{j}: An integer that specifies the numeric index of the column in which the coefficient is located.
- \text{lb}: A vector containing the lower bounds.
- \text{ub}: A vector containing the upper bounds.

Details

Interface to the C function \text{chgColsBnds} which calls the CPLEX function \text{CPXchgBds}.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

\text{chgBndsCplex}, \text{tightenBndsCplex}
chgColTypeCPLEX  

Change Types of a Set of Variables of a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgctype. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgColTypeCPLEX(env, lp, ncols, ind, xtype)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- ncols: Number of bounds to be changed.
- ind: Indices of bounds to be changed.
- xtype: A vector containing characters that represent the new types for the columns specified in indices.

Details

Interface to the C function chgColType which calls the CPLEX function CPXchgctype.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgMipstarts. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```c
chgMIPstartsCPLEX(env, lp, mcnt, mipstartindices, nzcnt,
                     beg, varindices, values, effortlevel)
```

Arguments

- `env` An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `mcnt` Number of MIP starts to be changed.
- `mipstartindices` Array of length `mcnt` containing the numeric indices of the MIP starts to be changed.
- `nzcnt` Number of entries to be changed.
- `beg` Array of length `mcnt` used with `varindices` and `values`. Consult the IBM ILOG CPLEX documentation for more detailed information.
- `varindices` Aarray of length `nzcnt` containing the numeric indices of the columns corresponding to the variables which are assigned starting values. Consult the IBM ILOG CPLEX documentation for more detailed information.
- `values` Array of length `nzcnt` containing the values to use for the MIP starts. Consult the IBM ILOG CPLEX documentation for more detailed information.
- `effortlevel` Array of length `mcnt`. Consult the IBM ILOG CPLEX documentation for more detailed information.

Details

Interface to the C function `chgMIPstarts` which calls the CPLEX function CPXchgMipstarts.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausJonathan.fritzemeier@uni-duesseldorf.de>
chgNameCPLEX

References


---

chgNameCPLEX  

Change the Name of a Constraint a Variable in a CPLEX Problem Object.

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

  chgNameCPLEX(env, lp, key, ij, name)

Arguments

  env     An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
  lp      An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
  key     A character to specify whether a row name or a column name should be changed.
  ij      An integer that specifies the numeric index of the column or row whose name is to be changed.
  name    A pointer to a character string containing the new name.

Details

Interface to the C function chgName which calls the CPLEX function CPXchgname.

Value

  Zero if successful, otherwise nonzero.

Author(s)

  Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
  Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

chgObjCPELEX  

Change Linear Objective Coefficients

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgoobj. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgObjCPELEX(env, lp, ncols, ind, val)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPELEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp   An object of class "cplexPtr" as returned by initProbCPELEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
ncols Number of bounds to be changed.
ind   Indices of bounds to be changed.
val   A vector containing the new values of the objective coefficients of the variables specified in ind.

Details

Interface to the C function chgobj which calls the CPLEX function CPXchgoobj.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

chgProbNameCplex

Change the Name of the Current Problem.

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgprobname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgProbNameCplex(env, lp, probname)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

probname The new name of the problem.

Details

Interface to the C function chgProbName which calls the CPLEX function CPXchgprobname.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXchgprobtype. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
cchgProbTypeCplex(env, lp, ptype)
```

**Arguments**

- `env` An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `ptype` A single integer value specifying the problem type.

**Details**

Interface to the C function `chgProbType` which calls the CPLEX function CPXchgprobtype.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

`getProbTypeCplex`, `cplexConstants` section “Problem Types”.
Description

Low level interface function to the IBM ILOG CPLEX function CPXchgqpcoef. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgQPcoefCplex(env, lp, i, j, val)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

i The first variable number.

j The second variable number.

val The new coefficient value.

Details

Interface to the C function chgQPcoef which calls the CPLEX function CPXchgqpcoef.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

chgCoefCplex
chgRhsCplex

Change Righthand Side Coefficients

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgrhs. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgRhsCplex(env, lp, nrows, ind, val)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- nrows: Number of bounds to be changed.
- ind: Indices of bounds to be changed.
- val: A vector containing the new values of the objective coefficients of the variables specified in ind.

Details

Interface to the C function chgRhs which calls the CPLEX function CPXchgrhs.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

chgRngValCplex

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgRngVal. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgRngValCplex(env, lp, nrows, ind, val)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **nrows**: Number of bounds to be changed.
- **ind**: Indices of bounds to be changed.
- **val**: A vector containing the new values of the objective coefficients of the variables specified in ind.

Details

Interface to the C function chgRngVal which calls the CPLEX function CPXchgRngVal.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

chgRowNameCPLEX Change Names of Linear Constraints

Description
Low level interface function to the IBM ILOG CPLEX function CPXchgrowname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
chgRowNameCPLEX(env, lp, nnames, ind, names)

Arguments
env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
nnames A vector that specifies the total number of variable names to be changed.
ind A vector containing the numeric indices indices of the variables for which the names are to be changed.
names A vector containing the strings of the new variable names for the columns specified in ind.

Details
Interface to the C function chgRowName which calls the CPLEX function CPXchgrowname.

Value
Zero if successful, otherwise nonzero.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References
chgSenseCplex

Change Sense of a Set of Linear Constraints

Description

Low level interface function to the IBM ILOG CPLEX function CPXchgsense. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

chgSenseCplex(env, lp, nrows, ind, sense)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- nrows: Number of bounds to be changed.
- ind: Indices of bounds to be changed.
- sense: A vector containing characters that tell the new sense of the linear constraints specified in ind.

Details

Interface to the C function chgSense which calls the CPLEX function CPXchgsense.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

chgTerminateCplex

Change Termination Signal

Description

The function chgTerminateCplex changes termination signal.

Usage

chgTerminateCplex(env, tval = 1)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- tval: Single integer value.

Value

NULL

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

setTerminateCplex, delTerminateCplex, printTerminateCplex

cleanupCoeffCplex

change Problem Coefficients to Zero That are Smaller in Magnitude Than the Tolerance Specified in the Argument eps

Description

Low level interface function to the IBM ILOG CPLEX function CPXcleanup. Consult the IBM ILOG CPLEX documentation for more detailed information.
Usage

cleanupCoeffCplex(env, lp, eps)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

eps Single numeric value giving the tolerance.

Details

Interface to the C function cleanupCoeff which calls the CPLEX function CPXcleanup.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


---

cloneProbCplex

Copy a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXcloneprob. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

cloneProbCplex(env, lp, ptrtype = "cplex_prob")

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

ptrtype A name for the pointer object.
Details

Interface to the C function cloneProb which calls the CPLEX function CPXcloneprob.

Value

If successful a pointer to the new CPLEX problem object as returned by initProbCLEX (an object of class "cplexPtr"), otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


---

closeEnvCLEX  Free all of the Data Structures Associated With CPLEX

Description

Low level interface function to the IBM ILOG CPLEX function CPXcloseCLEX. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

closeEnvCLEX(env)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

Details

Interface to the C function closeEnv which calls the CPLEX function CPXcloseCLEX.

Value

Zero if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
closeFileCPLEX

References


See Also

openEnvCPLEX

closeFileCPLEX  Close a File

Description

Low level interface function to the IBM ILOG CPLEX function CPXfclose. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

closeFileCPLEX(cpfile)

Arguments

cpfile  A pointer to a file as returned by openFileCPLEX.

Details

Interface to the C function closeFile which calls the CPLEX function CPXfclose.

Value

Zero if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

fileputCPLEX, openFileCPLEX
closeProbCplex

**Close CPLEX Environment And Remove CPLEX Problem Object**

**Description**

The function `closeProbCplex` closes a CPLEX environment and removes a CPLEX problem object.

**Usage**

```r
closeProbCplex(prob)
```

**Arguments**

- `prob` A list containing a pointer to an IBM ILOG CPLEX environment and a Pointer to an IBM ILOG CPLEX problem object. Both elements are objects of class "cplexPtr" as returned by `openProbCplex`.

**Details**

Interface to the C functions `delProb` and `closeEnv` calling CPLEX functions `CPXcloseCplex` and `CPXfreeProb`.

**Value**

An integer vector containing the return values of `CPXcloseCplex` and `CPXfreeProb`.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

`openProbCplex`
cLpWriteCplex

Write an LP Format File Containing Identified Conflict

Description

Low level interface function to the IBM ILOG CPLEX function CPXclpwrite. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

cLpWriteCplex(env, lp, fname)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

fname Single character value giving the filename to write to.

Details

Interface to the C function cLpWriteCplex which calls the CPLEX function CPXclpwrite.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Manage Modification Steps Closely

Description

Low level interface function to the IBM ILOG CPLEX function CPXcompletelp. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

completelpCPLEX(env, lp)

Arguments

env  
An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp  
An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function completelpCPLEX which calls the CPLEX function CPXcompletelp.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**copyBaseCplex**

Copies a Basis Into a CPLEX Problem Object.

**Description**

Low level interface function to the IBM ILOG CPLEX function `cpxcopybase`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
copyBaseCplex(env, lp, cstat, rstat)
```

**Arguments**

- `env` An object of class `cplexPtr` as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class `cplexPtr` as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `cstat` A vector containing the basis status of the columns in the constraint matrix.
- `rstat` A vector containing the basis status of the slack, or surplus, or artificial variable associated with each row in the constraint matrix.

**Details**

Interface to the C function `copyBase` which calls the CPLEX function `CPXcopybase`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

copyColTypeCPLEX \hspace{1cm} Copy Variable Type Information Into a Given Problem

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXchgproobname`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
copyColTypeCPLEX(env, lp, xtype)
```

**Arguments**

- `env` An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `xctype` A vector containing the type of each column in the constraint matrix.

**Details**

Interface to the C function `copyColType` which calls the CPLEX function `CPXcopyctype`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

Copy Data Defining an LP Problem to a CPLEX Problem Object.

Description

Low level interface function to the IBM ILOG CPLEX function CPXcopylp. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

copyLpCPLEX(env, lp, nCols, nRows, lpdir, objf, rhs, sense, 
matbeg, matcnt, matind, matval, lb, ub, rngval = NULL)

Arguments

env 
An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp 
An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

nCols 
Number of columns in the constraint matrix.

nRows 
Number of rows in the constraint matrix.

lpdir 
Single integer value that specifies whether the problem is a minimization or maximization problem.

objf 
The objective function coefficients.

rhs 
The righthand side values for each constraint in the constraint matrix.

sense 
The sense of each constraint in the constraint matrix.

matbeg 
Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.

matcnt 
Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.

matind 
Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.

matval 
Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.

lb 
Containing the lower bound on each of the variables.

ub 
Containing the lower bound on each of the variables.

rngval 
Containing the range value of each ranged constraint.

Details

Interface to the C function copyLp which calls the CPLEX function CPXcopylp.
**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


---

**copyLpwNamesCplex**  
*Copy Data Defining an LP Problem to a CPLEX Problem Object.*

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPlxcopyLpwNames`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
EllcopyLpwNamesCplex(env, lp, ncols, nrows, lpdire, objf, rhs, sense,
  matbeg, matcnt, matind, matval, lb, ub,
  rngval = NULL, cnames = NULL, rnames = NULL)
```

**Arguments**

- `env`: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp`: An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `ncols`: Number of columns in the constraint matrix.
- `nrows`: Number of rows in the constraint matrix.
- `lpdir`: Single integer value that specifies whether the problem is a minimization or maximization problem.
- `objf`: The objective function coefficients.
- `rhs`: The righthand side values for each constraint in the constraint matrix.
- `sense`: The sense of each constraint in the constraint matrix.
- `matbeg`: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
- `matcnt`: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
**copyObjNameCplex**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>matind</td>
<td>Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>matval</td>
<td>Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>lb</td>
<td>Containing the lower bound on each of the variables.</td>
</tr>
<tr>
<td>ub</td>
<td>Containing the lower bound on each of the variables.</td>
</tr>
<tr>
<td>rngval</td>
<td>Containing the range value of each ranged constraint.</td>
</tr>
<tr>
<td>cnames</td>
<td>Names of the matrix columns or, equivalently, the variable names.</td>
</tr>
<tr>
<td>rnames</td>
<td>Names of the matrix rows or, equivalently, the constraint names.</td>
</tr>
</tbody>
</table>

**Details**

Interface to the C function `copyLpWNames` which calls the CPLEX function `CPXcopyLpWNames`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


---

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXcopyObjName`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
copyObjNameCplex(env, lp, oname)
```
Arguments

- **env**: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **oname**: A pointer to a character string containing the objective name.

Details

Interface to the C function `copyobjname` which calls the CPLEX function `CPXcopyobjname`.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


---

**copyOrderCplex** *Copy Priority Order to CPLEX Problem Object*

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXcopyorder`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

`copyOrderCplex(env, lp, cnt, indices, priority = NULL, direction = NULL)`

**Arguments**

- **env**: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **cnt**: Number of entries.
- **indices**: Indices of the columns corresponding to the integer variables that are assigned priorities.
copyPartBaseCPLEX

| priority | Priorities assigned to the integer variables. |
| direction | Branching direction assigned to the integer variables. |

**Details**

Interface to the C function `copyOrder` which calls the CPLEX function `CPXcopyOrder`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


---

**copyPartBaseCPLEX**  
Copies a partial basis into an LP problem object.

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXcopyPartialBase`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
copyPartBaseCPLEX(env, lp, ncind, cind, cstat, nrind, rind, rstat)
```

**Arguments**

- **env**
  An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.

- **lp**
  An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.

- **ncind**
  An integer that specifies the number of variable or column status values specified.

- **cind**
  A vector that contains the indices of the variables for which status values are being specified.

- **cstat**
  A vector where the ith entry contains the status for variable `cind[i]`.

- **nrind**
  An integer that specifies the number of slack, surplus, or artificial status values specified.
copyQPsepCplex

\[ \text{rind} \quad \text{A vector \texttt{rcnt} that contains the indices of the slack, surplus, or artificial variables for which status values are being specified.} \]

\[ \text{rstat} \quad \text{A vector of where the i-th entry contains the status for slack, surplus, or artificial \texttt{rind}[i].} \]

Details

Interface to the C function \texttt{copyPartBase} which calls the CPLEX function \texttt{CPXcopypartialbase}.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


---

\[ \text{copyQPsepCPLEX} \quad \text{Copy the Quadratic Objective Matrix \( Q \) for a Separable QP Problem} \]

Description

Low level interface function to the IBM ILOG CPLEX function \texttt{CPXcopyqpsep}. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

\[ \text{copyQPsepCPLEX}(\texttt{env}, \texttt{lp}, \texttt{qsepvec}) \]

Arguments

\[ \text{env} \quad \text{An object of class } \texttt{cplexPtr} \text{ as returned by } \texttt{openEnvCPLEX}. \text{This is basically a pointer to an IBM ILOG CPLEX environment.} \]

\[ \text{lp} \quad \text{An object of class } \texttt{cplexPtr} \text{ as returned by } \texttt{initProbCPLEX}. \text{This is basically a pointer to an IBM ILOG CPLEX problem object.} \]

\[ \text{qsepvec} \quad \text{A vector containing the quadratic coefficients.} \]

Details

Interface to the C function \texttt{copyQPsep} which calls the CPLEX function \texttt{CPXcopyqpsep}. 
Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


copyQuadCplex

## Copy a Quadratic Objective Matrix $Q$ When $Q$ is not Diagonal.

### Description

Low level interface function to the IBM ILOG CPLEX function CPXcopyquad. Consult the IBM ILOG CPLEX documentation for more detailed information.

### Usage

```r
copyQuadCplex(env, lp, qmatbeg, qmatcnt, qmatind, qmatval)
```

### Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **qmatbeg**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **qmatcnt**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **qmatind**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **qmatval**: Array that defines the constraint matrix. Consult the IBM ILOG CPLEX documentation for more detailed information.

### Details

Interface to the C function copyQuad which calls the CPLEX function CPXcopyquad.
**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliedie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


---

**copyStartCPLEX**  
*Provides Starting Information for Use in a Subsequent Call to a Simplex Optimization Routine.*

---

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXcopystart. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
copyStartCPLEX(env, lp,  
cstat = NULL, rstat = NULL,  
cprim = NULL, rprim = NULL,  
cdual = NULL, rdual = NULL)
```

**Arguments**

- `env`  
  An object of class "cpplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.

- `lp`  
  An object of class "cpplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.

- `cstat`  
  A vector containing the basis status of the columns in the constraint matrix.

- `rstat`  
  A vector containing the basis status of the slack, surplus, or artificial variable associated with each row in the constraint matrix.

- `cprim`  
  A vector containing the initial primal values of the column variables.

- `rprim`  
  A vector containing the initial primal values of the slack (row) variables.

- `cdual`  
  A vector containing the initial values of the reduced costs for the column variables.

- `rdual`  
  A vector containing the initial values of the dual variables for the rows.
Details

Interface to the C function `copyStart` which calls the CPLEX function `CPXcopyStart`.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


cplexConstants

Constants, Parameters, Return and Status Codes of IBM ILOG CPLEX

Description

This is a list containing constants used by IBM ILOG CPLEX. Consult the IBM ILOG CPLEX manual for more information, in particular for the use of control parameters.

General Parameters

- `CPX_INFBOUND`: 1.0E+20
- `CPX_STR_PARAM_MAX`: 512

Types of parameters

- `CPX_PARAMTYPE_NONE`: 0
- `CPX_PARAMTYPE_INT`: 1
- `CPX_PARAMTYPE_DOUBLE`: 2
- `CPX_PARAMTYPE_STRING`: 3
- `CPX_PARAMTYPE_LONG`: 4
### Values returned for stat by solution

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_STAT_OPTIMAL</td>
<td>1</td>
</tr>
<tr>
<td>CPX_STAT_UNBOUNDED</td>
<td>2</td>
</tr>
<tr>
<td>CPX_STAT_INFEASIBLE</td>
<td>3</td>
</tr>
<tr>
<td>CPX_STAT_INFORUNBD</td>
<td>4</td>
</tr>
<tr>
<td>CPX_STAT_OPTIMAL_INFEAS</td>
<td>5</td>
</tr>
<tr>
<td>CPX_STAT_NUM_BEST</td>
<td>6</td>
</tr>
<tr>
<td>CPX_STAT_ABORT_IT_LIM</td>
<td>10</td>
</tr>
<tr>
<td>CPX_STAT_ABORT_TIME_LIM</td>
<td>11</td>
</tr>
<tr>
<td>CPX_STAT_ABORT_OBJ_LIM</td>
<td>12</td>
</tr>
<tr>
<td>CPX_STAT_ABORT_USER</td>
<td>13</td>
</tr>
<tr>
<td>CPX_STAT_FEASIBLE_RELAXED_SUM</td>
<td>14</td>
</tr>
<tr>
<td>CPX_STAT_OPTIMAL_RELAXED_SUM</td>
<td>15</td>
</tr>
<tr>
<td>CPX_STAT_FEASIBLE_RELAXED_INF</td>
<td>16</td>
</tr>
<tr>
<td>CPX_STAT_OPTIMAL_RELAXED_INF</td>
<td>17</td>
</tr>
<tr>
<td>CPX_STAT_FEASIBLE_RELAXED_QUAD</td>
<td>18</td>
</tr>
<tr>
<td>CPX_STAT_OPTIMAL_RELAXED_QUAD</td>
<td>19</td>
</tr>
<tr>
<td>CPX_STAT_FEASIBLE</td>
<td>23</td>
</tr>
<tr>
<td>CPX_STAT_ABORT_DETTIME_LIM</td>
<td>25</td>
</tr>
</tbody>
</table>

### Solution type return values from CPXsolninfo

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_NO_SOLN</td>
<td>0</td>
</tr>
<tr>
<td>CPX_BASIC_SOLN</td>
<td>1</td>
</tr>
<tr>
<td>CPX_NONBASIC_SOLN</td>
<td>2</td>
</tr>
<tr>
<td>CPX_PRIMAL_SOLN</td>
<td>3</td>
</tr>
</tbody>
</table>

### Values of presolve stats for columns and rows

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_PRECOL_LOW</td>
<td>-1</td>
<td>fixed to original lb</td>
</tr>
<tr>
<td>CPX_PRECOL_UP</td>
<td>-2</td>
<td>fixed to original ub</td>
</tr>
<tr>
<td>CPX_PRECOL_FIX</td>
<td>-3</td>
<td>fixed to some other value</td>
</tr>
<tr>
<td>CPX_PRECOL_AGG</td>
<td>-4</td>
<td>aggregated $y = a \times x + b$</td>
</tr>
<tr>
<td>CPX_PRECOL_OTHER</td>
<td>-5</td>
<td>cannot be expressed by a linear combination of active variables in the presolved model crushing will fail if it has to touch such a variable</td>
</tr>
<tr>
<td>CPX_PREROW_RED</td>
<td>-1</td>
<td>redundant row removed in presolved model</td>
</tr>
<tr>
<td>CPX_PREROW_AGG</td>
<td>-2</td>
<td>used to aggregate a variable</td>
</tr>
<tr>
<td>CPX_PREROW_OTHER</td>
<td>-3</td>
<td>other, for example merge two inequalities into a single equation</td>
</tr>
</tbody>
</table>
Generic constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_ON</td>
<td>1</td>
</tr>
<tr>
<td>CPX_OFF</td>
<td>0</td>
</tr>
<tr>
<td>CPX_MAX</td>
<td>-1</td>
</tr>
<tr>
<td>CPX_MIN</td>
<td>1</td>
</tr>
</tbody>
</table>

Primal simplex pricing algorithm

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_PPRIIND_PARTIAL</td>
<td>-1</td>
</tr>
<tr>
<td>CPX_PPRIIND_AUTO</td>
<td>0</td>
</tr>
<tr>
<td>CPX_PPRIIND_DEVEX</td>
<td>1</td>
</tr>
<tr>
<td>CPX_PPRIIND_STEEP</td>
<td>2</td>
</tr>
<tr>
<td>CPX_PPRIIND_STEEPQSTART</td>
<td>3</td>
</tr>
<tr>
<td>CPX_PPRIIND_FULL</td>
<td>4</td>
</tr>
</tbody>
</table>

Dual simplex pricing algorithm

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_DPRIIND_AUTO</td>
<td>0</td>
</tr>
<tr>
<td>CPX_DPRIIND_FULL</td>
<td>1</td>
</tr>
<tr>
<td>CPX_DPRIIND_STEEP</td>
<td>2</td>
</tr>
<tr>
<td>CPX_DPRIIND_FULL_STEEP</td>
<td>3</td>
</tr>
<tr>
<td>CPX_DPRIIND_STEEPQSTART</td>
<td>4</td>
</tr>
<tr>
<td>CPX_DPRIIND_DEVEX</td>
<td>5</td>
</tr>
</tbody>
</table>

PARALLELMODE values

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_PARALLEL_DETERMINISTIC</td>
<td>1</td>
</tr>
<tr>
<td>CPX_PARALLEL_AUTO</td>
<td>0</td>
</tr>
<tr>
<td>CPX_PARALLEL_OPPORTUNISTIC</td>
<td>-1</td>
</tr>
</tbody>
</table>

Values for CPX_PARAM_WRITELEVEL

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_WRITELEVEL_AUTO</td>
<td>0</td>
</tr>
</tbody>
</table>
cplexConstants

Values for CPX_PARAM_SOLUTIONTARGET

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_SOLUTIONTARGET_AUTO</td>
<td>0</td>
</tr>
<tr>
<td>CPX_SOLUTIONTARGET_OPTIMALCONVEX</td>
<td>1</td>
</tr>
<tr>
<td>CPX_SOLUTIONTARGET_FIRSTORDER</td>
<td>2</td>
</tr>
<tr>
<td>CPX_SOLUTIONTARGET_OPTIMALGLOBAL</td>
<td>3</td>
</tr>
</tbody>
</table>

LP/QP solution algorithms

Used as possible values for CPX_PARAM_LPMETHOD, CPX_PARAM_QPMETHOD, CPX_PARAM_BARCROSSALG, CPXgetmethod...

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_ALG_NONE</td>
<td>-1</td>
</tr>
<tr>
<td>CPX_ALG_AUTOMATIC</td>
<td>0</td>
</tr>
<tr>
<td>CPX_ALG_PRIMAL</td>
<td>1</td>
</tr>
<tr>
<td>CPX_ALG_DUAL</td>
<td>2</td>
</tr>
<tr>
<td>CPX_ALG_NET</td>
<td>3</td>
</tr>
<tr>
<td>CPX_ALG_BARRIER</td>
<td>4</td>
</tr>
<tr>
<td>CPX_ALG_SIFTING</td>
<td>5</td>
</tr>
<tr>
<td>CPX_ALG_CONCURRENT</td>
<td>6</td>
</tr>
<tr>
<td>CPX_ALG_BAROPT</td>
<td>7</td>
</tr>
<tr>
<td>CPX_ALG_PIVOTIN</td>
<td>8</td>
</tr>
<tr>
<td>CPX_ALG_PIVOTOUT</td>
<td>9</td>
</tr>
<tr>
<td>CPX_ALG_PIVOT</td>
<td>10</td>
</tr>
<tr>
<td>CPX_ALG_FEASOPT</td>
<td>11</td>
</tr>
<tr>
<td>CPX_ALG_MIP</td>
<td>12</td>
</tr>
<tr>
<td>CPX_ALG_ROBUST</td>
<td>13</td>
</tr>
</tbody>
</table>

Basis status values

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_AT_LOWER</td>
<td>0</td>
</tr>
<tr>
<td>CPX_BASIC</td>
<td>1</td>
</tr>
<tr>
<td>CPX_AT_UPPER</td>
<td>2</td>
</tr>
<tr>
<td>CPX_FREE_SUPER</td>
<td>3</td>
</tr>
</tbody>
</table>
Variable types for ctype array

CPX_CONTINUOUS   "C"
CPX_BINARY       "B"
CPX_INTEGER      "I"
CPX_SEMICONT     "S"
CPX_SEMIINT      "N"

PREREDUCE settings

CPX_PREREDUCE_PRIMALANDDUAL 3
CPX_PREREDUCE_DUALONLY      2
CPX_PREREDUCE_PRIMALONLY    1
CPX_PREREDUCE_NOPRIMALORDUAL 0

Conflict statuses

CPX_STAT_CONFLICT_FEASIBLE            30
CPX_STAT_CONFLICT_MINIMAL             31
CPX_STAT_CONFLICT_ABORT_CONTRADICTION 32
CPX_STAT_CONFLICT_ABORT_TIME_LIM      33
CPX_STAT_CONFLICT_ABORT_IT_LIM        34
CPX_STAT_CONFLICT_ABORT_NODE_LIM      35
CPX_STAT_CONFLICT_ABORT_OBJ_LIM       36
CPX_STAT_CONFLICT_ABORT_MEM_LIM       37
CPX_STAT_CONFLICT_ABORT_USER          38
CPX_STAT_CONFLICT_ABORT_DETTIME_LIM   39

Conflict status values

CPX_CONFLICT_EXCLUDED        -1
CPX_CONFLICT_POSSIBLE_MEMBER 0
CPX_CONFLICT_POSSIBLE LB    1
CPX_CONFLICT_POSSIBLE UB    2
CPX_CONFLICT_MEMBER          3
CPX_CONFLICT_LB  4
CPX_CONFLICT UB  5

Problem Types

Types 4, 9, and 12 are internal, the others are for users.

CPXPROB_LP        0
CPXPROB_MILP      1
CPXPROB_FIXEDMILP 3
CPXPROB_NODELP    4
CPXPROB_QP        5
CPXPROB_MIQP      7
CPXPROB_FIXEDMIQP 8
CPXPROB_NODEQP    9
CPXPROB_QCP      10
CPXPROB_MIQCP    11
CPXPROB_NODEQCP  12

CPLEX Parameter numbers

CPX_PARAM_ADVIND  1001
CPX_PARAM_AGGFILL 1002
CPX_PARAM_AGGIND  1003
CPX_PARAM_BASINTERVAL 1004
CPX_PARAM_CFILEMUL 1005
CPX_PARAM_CLOCKTYPE 1006
CPX_PARAM_CRAIND  1007
CPX_PARAM_DEPIND  1008
CPX_PARAM_DPRIIND  1009
CPX_PARAM_PRICELIM 1010
CPX_PARAM_EPRMK  1013
CPX_PARAM_EPOPT  1014
CPX_PARAM_EPPER  1015
CPX_PARAM_EPRHS  1016
CPX_PARAM_FASTMIP 1017
CPX_PARAM_SIMDISPLAY 1019
CPX_PARAM_ITLIM  1020
CPX_PARAM ROWREADLIM 1021
CPX_PARAM_NETFIND 1022
CPX_PARAM_COLREADLIM 1023
CPX_PARAM_NZREADLIM 1024
CPX_PARAM_OBJLLLIM 1025
CPX_PARAM_OBJULIM  1026
<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_PARAM_PERIND</td>
<td>1027</td>
</tr>
<tr>
<td>CPX_PARAM_PERLIM</td>
<td>1028</td>
</tr>
<tr>
<td>CPX_PARAM_PPRIND</td>
<td>1029</td>
</tr>
<tr>
<td>CPX_PARAM_PREIND</td>
<td>1030</td>
</tr>
<tr>
<td>CPX_PARAM_REINV</td>
<td>1031</td>
</tr>
<tr>
<td>CPX_PARAM_REVERSEIND</td>
<td>1032</td>
</tr>
<tr>
<td>CPX_PARAM_RFILEMUL</td>
<td>1033</td>
</tr>
<tr>
<td>CPX_PARAM_SCAIND</td>
<td>1034</td>
</tr>
<tr>
<td>CPX_PARAM_SCRIND</td>
<td>1035</td>
</tr>
<tr>
<td>CPX_PARAM_SINGLIM</td>
<td>1037</td>
</tr>
<tr>
<td>CPX_PARAM_SINGTOL</td>
<td>1038</td>
</tr>
<tr>
<td>CPX_PARAM_TILIM</td>
<td>1039</td>
</tr>
<tr>
<td>CPX_PARAM_XXIND</td>
<td>1041</td>
</tr>
<tr>
<td>CPX_PARAM_PREDUAL</td>
<td>1044</td>
</tr>
<tr>
<td>CPX_PARAM_EPOPT_H</td>
<td>1049</td>
</tr>
<tr>
<td>CPX_PARAM_EPRHS_H</td>
<td>1050</td>
</tr>
<tr>
<td>CPX_PARAM_PREPASS</td>
<td>1052</td>
</tr>
<tr>
<td>CPX_PARAM_DATACHECK</td>
<td>1056</td>
</tr>
<tr>
<td>CPX_PARAM_REDUCE</td>
<td>1057</td>
</tr>
<tr>
<td>CPX_PARAM_PRELINEAR</td>
<td>1058</td>
</tr>
<tr>
<td>CPX_PARAM_LPMETHOD</td>
<td>1062</td>
</tr>
<tr>
<td>CPX_PARAM_QPMETHOD</td>
<td>1063</td>
</tr>
<tr>
<td>CPX_PARAM_WORKDIR</td>
<td>1064</td>
</tr>
<tr>
<td>CPX_PARAM_WORKMEM</td>
<td>1065</td>
</tr>
<tr>
<td>CPX_PARAM_THREADS</td>
<td>1067</td>
</tr>
<tr>
<td>CPX_PARAM_CONFLICTDISPLAY</td>
<td>1074</td>
</tr>
<tr>
<td>CPX_PARAM_SIFTDISPLAY</td>
<td>1076</td>
</tr>
<tr>
<td>CPX_PARAM_SIFTALG</td>
<td>1077</td>
</tr>
<tr>
<td>CPX_PARAM_SIFTILIM</td>
<td>1078</td>
</tr>
<tr>
<td>CPX_PARAM_MPSLONGNUM</td>
<td>1081</td>
</tr>
<tr>
<td>CPX_PARAM_MEMORYEMPHASIS</td>
<td>1082</td>
</tr>
<tr>
<td>CPX_PARAM_NUMERICALEMPHASIS</td>
<td>1083</td>
</tr>
<tr>
<td>CPX_PARAM_FEAOPTMODE</td>
<td>1084</td>
</tr>
<tr>
<td>CPX_PARAM_PARALLELMODE</td>
<td>1109</td>
</tr>
<tr>
<td>CPX_PARAM_TUNINGMEASURE</td>
<td>1110</td>
</tr>
<tr>
<td>CPX_PARAM_TUNINGREPEAT</td>
<td>1111</td>
</tr>
<tr>
<td>CPX_PARAM_TUNINGTILIM</td>
<td>1112</td>
</tr>
<tr>
<td>CPX_PARAM_TUNINGDISPLAY</td>
<td>1113</td>
</tr>
<tr>
<td>CPX_PARAM_WRITELEVEL</td>
<td>1114</td>
</tr>
<tr>
<td>CPX_PARAM_RANDOMSEED</td>
<td>1124</td>
</tr>
<tr>
<td>CPX_PARAM_DETILIM</td>
<td>1127</td>
</tr>
<tr>
<td>CPX_PARAM_FILEENCODING</td>
<td>1129</td>
</tr>
<tr>
<td>CPX_PARAM_APIENCODING</td>
<td>1130</td>
</tr>
<tr>
<td>CPX_PARAM_SOLUTIONTARGET</td>
<td>1131</td>
</tr>
<tr>
<td>CPX_PARAM_CLONELOG</td>
<td>1132</td>
</tr>
<tr>
<td>CPX_PARAM_TUNINGDETILIM</td>
<td>1139</td>
</tr>
<tr>
<td>CPX_PARAM_ALL_MIN</td>
<td>1000</td>
</tr>
<tr>
<td>CPX_PARAM_ALL_MAX</td>
<td>6000</td>
</tr>
</tbody>
</table>
Values for `CPX_PARAM_TUNINGMEASURE`

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_TUNE_AVERAGE</td>
<td>1</td>
</tr>
<tr>
<td>CPX_TUNE_MINMAX</td>
<td>2</td>
</tr>
</tbody>
</table>

Values for incomplete tuning

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_TUNE_ABORT</td>
<td>1</td>
</tr>
<tr>
<td>CPX_TUNE_TILIM</td>
<td>2</td>
</tr>
<tr>
<td>CPX_TUNE_DETILIM</td>
<td>3</td>
</tr>
</tbody>
</table>

Quality query identifiers

<table>
<thead>
<tr>
<th>Identifier</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_MAX_PRIMAL_INFEAS</td>
<td>1</td>
</tr>
<tr>
<td>CPX_MAX_SCALED_PRIMAL_INFEAS</td>
<td>2</td>
</tr>
<tr>
<td>CPX_SUM_PRIMAL_INFEAS</td>
<td>3</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_PRIMAL_INFEAS</td>
<td>4</td>
</tr>
<tr>
<td>CPX_MAX_DUAL_INFEAS</td>
<td>5</td>
</tr>
<tr>
<td>CPX_MAX_SCALED_DUAL_INFEAS</td>
<td>6</td>
</tr>
<tr>
<td>CPX_SUM_DUAL_INFEAS</td>
<td>7</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_DUAL_INFEAS</td>
<td>8</td>
</tr>
<tr>
<td>CPX_MAX_INT_INFEAS</td>
<td>9</td>
</tr>
<tr>
<td>CPX_SUM_INT_INFEAS</td>
<td>10</td>
</tr>
<tr>
<td>CPX_MAX_PRIMAL_RESIDUAL</td>
<td>11</td>
</tr>
<tr>
<td>CPX_MAX_SCALED_PRIMAL_RESIDUAL</td>
<td>12</td>
</tr>
<tr>
<td>CPX_SUM_PRIMAL_RESIDUAL</td>
<td>13</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_PRIMAL_RESIDUAL</td>
<td>14</td>
</tr>
<tr>
<td>CPX_MAX_DUAL_RESIDUAL</td>
<td>15</td>
</tr>
<tr>
<td>CPX_MAX_SCALED_DUAL_RESIDUAL</td>
<td>16</td>
</tr>
<tr>
<td>CPX_SUM_DUAL_RESIDUAL</td>
<td>17</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_DUAL_RESIDUAL</td>
<td>18</td>
</tr>
<tr>
<td>CPX_MAX_COMP_SLACK</td>
<td>19</td>
</tr>
<tr>
<td>CPX_SUM_COMP_SLACK</td>
<td>21</td>
</tr>
<tr>
<td>CPX_MAX_X</td>
<td>23</td>
</tr>
<tr>
<td>CPX_MAX_SCALED_X</td>
<td>24</td>
</tr>
<tr>
<td>CPX_MAX_PI</td>
<td>25</td>
</tr>
</tbody>
</table>
## cplexConstants

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_MAX_SCALED_PI</td>
<td>26</td>
</tr>
<tr>
<td>CPX_MAX_SLACK</td>
<td>27</td>
</tr>
<tr>
<td>CPX_MAX_SCALED_SLACK</td>
<td>28</td>
</tr>
<tr>
<td>CPX_MAX_RED_COST</td>
<td>29</td>
</tr>
<tr>
<td>CPX_MAX_SCALED_RED_COST</td>
<td>30</td>
</tr>
<tr>
<td>CPX_SUM_X</td>
<td>31</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_X</td>
<td>32</td>
</tr>
<tr>
<td>CPX_SUM_PI</td>
<td>33</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_PI</td>
<td>34</td>
</tr>
<tr>
<td>CPX_SUM_SLACK</td>
<td>35</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_SLACK</td>
<td>36</td>
</tr>
<tr>
<td>CPX_SUM_RED_COST</td>
<td>37</td>
</tr>
<tr>
<td>CPX_SUM_SCALED_RED_COST</td>
<td>38</td>
</tr>
<tr>
<td>CPX_KAPPA</td>
<td>39</td>
</tr>
<tr>
<td>CPX_OBJ_GAP</td>
<td>40</td>
</tr>
<tr>
<td>CPX_DUAL_OBJ</td>
<td>41</td>
</tr>
<tr>
<td>CPX_PRIMAL_OBJ</td>
<td>42</td>
</tr>
<tr>
<td>CPX_MAX_QCPRIMAL_RESIDUAL</td>
<td>43</td>
</tr>
<tr>
<td>CPX_SUM_QCPRIMAL_RESIDUAL</td>
<td>44</td>
</tr>
<tr>
<td>CPX_MAX_QCSLACK_INFEAS</td>
<td>45</td>
</tr>
<tr>
<td>CPX_SUM_QCSLACK_INFEAS</td>
<td>46</td>
</tr>
<tr>
<td>CPX_MAX_QCSLACK</td>
<td>47</td>
</tr>
<tr>
<td>CPX_SUM_QCSLACK</td>
<td>48</td>
</tr>
<tr>
<td>CPX_MAX_INDSLACK_INFEAS</td>
<td>49</td>
</tr>
<tr>
<td>CPX_SUM_INDSLACK_INFEAS</td>
<td>50</td>
</tr>
<tr>
<td>CPX_EXACT_KAPPA</td>
<td>51</td>
</tr>
<tr>
<td>CPX_KAPPA_STABLE</td>
<td>52</td>
</tr>
<tr>
<td>CPX_KAPPA_SUSPICIOUS</td>
<td>53</td>
</tr>
<tr>
<td>CPX_KAPPA_UNSTABLE</td>
<td>54</td>
</tr>
<tr>
<td>CPX_KAPPA_ILLLPOSED</td>
<td>55</td>
</tr>
<tr>
<td>CPX_KAPPA_MAX</td>
<td>56</td>
</tr>
<tr>
<td>CPX_KAPPA_ATTENTION</td>
<td>57</td>
</tr>
</tbody>
</table>

## feasopt options

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_FEASOPT_MIN_SUM</td>
<td>0</td>
</tr>
<tr>
<td>CPX_FEASOPT_OPT_SUM</td>
<td>1</td>
</tr>
<tr>
<td>CPX_FEASOPT_MIN_INF</td>
<td>2</td>
</tr>
<tr>
<td>CPX_FEASOPT_OPT_INF</td>
<td>3</td>
</tr>
<tr>
<td>CPX_FEASOPT_MIN_QUAD</td>
<td>4</td>
</tr>
<tr>
<td>CPX_FEASOPT_OPT_QUAD</td>
<td>5</td>
</tr>
</tbody>
</table>
**File**: barconst.h

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_STAT_OPTIMAL_FACE_UNBOUNDED</td>
<td>20</td>
</tr>
<tr>
<td>CPX_STAT_ABORT_PRIM_OBJ_LIM</td>
<td>21</td>
</tr>
<tr>
<td>CPX_STAT_ABORT_DUAL_OBJ_LIM</td>
<td>22</td>
</tr>
<tr>
<td>CPX_STAT_FIRSTORDER</td>
<td>24</td>
</tr>
</tbody>
</table>

**Barrier parameters**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_PARAM_BARDSTART</td>
<td>3001</td>
</tr>
<tr>
<td>CPX_PARAM_BAREPCOMP</td>
<td>3002</td>
</tr>
<tr>
<td>CPX_PARAM_BARGROWTH</td>
<td>3003</td>
</tr>
<tr>
<td>CPX_PARAM_BAROBJRNG</td>
<td>3004</td>
</tr>
<tr>
<td>CPX_PARAM_BARPSTART</td>
<td>3005</td>
</tr>
<tr>
<td>CPX_PARAM_BARALG</td>
<td>3007</td>
</tr>
<tr>
<td>CPX_PARAM_BARCOLNZ</td>
<td>3009</td>
</tr>
<tr>
<td>CPX_PARAM_BARDISPLAY</td>
<td>3010</td>
</tr>
<tr>
<td>CPX_PARAM_BARITLIM</td>
<td>3012</td>
</tr>
<tr>
<td>CPX_PARAM_BARMAXCOR</td>
<td>3013</td>
</tr>
<tr>
<td>CPX_PARAM_BARORDER</td>
<td>3014</td>
</tr>
<tr>
<td>CPX_PARAM_BARSTARTALG</td>
<td>3017</td>
</tr>
<tr>
<td>CPX_PARAM_BARCROSSALG</td>
<td>3018</td>
</tr>
<tr>
<td>CPX_PARAM_BARQCPEPCOMP</td>
<td>3020</td>
</tr>
</tbody>
</table>

**Optimizing Problems**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_BARORDER_AUTO</td>
<td>0</td>
</tr>
<tr>
<td>CPX_BARORDER_AMD</td>
<td>1</td>
</tr>
<tr>
<td>CPX_BARORDER_AMF</td>
<td>2</td>
</tr>
<tr>
<td>CPX_BARORDER_ND</td>
<td>3</td>
</tr>
</tbody>
</table>

**MIP emphasis settings**

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_MIPEMPHASIS_BALANCED</td>
<td>0</td>
</tr>
<tr>
<td>CPX_MIPEMPHASIS_FEASIBILITY</td>
<td>1</td>
</tr>
<tr>
<td>CPX_MIPEMPHASIS_OPTIMALITY</td>
<td>2</td>
</tr>
<tr>
<td>CPX_MIPEMPHASIS_BESTBOUND</td>
<td>3</td>
</tr>
</tbody>
</table>
Values for sostype and branch type

- CPX_TYPE_VAR "0"
- CPX_TYPE_SOS1 "1"
- CPX_TYPE_SOS2 "2"
- CPX_TYPE_USER "X"
- CPX_TYPE_ANY "A"

Variable selection values

- CPX_VARSEL_MININFEAS -1
- CPX_VARSEL_DEFAULT 0
- CPX_VARSEL_MAXINFEAS 1
- CPX_VARSEL_PSEUDO 2
- CPX_VARSEL_STRONG 3
- CPX_VARSEL_PSEUDOREDUCED 4

Node selection values

- CPX_NODESEL_DFS 0
- CPX_NODESEL_BESTBOUND 1
- CPX_NODESEL_BESTEST 2
- CPX_NODESEL_BESTEST_ALT 3

Values for generated priority order

- CPX_MIPORDER_COST 1
- CPX_MIPORDER_BOUNDS 2
- CPX_MIPORDER_SCALED_COST 3
Values for direction array

CPX_BRANCH_GLOBAL 0
CPX_BRANCH_DOWN -1
CPX_BRANCH_UP 1

Values for CPX_PARAM_BRDIR

CPX_BDIR_DOWN -1
CPX_BDIR_AUTO 0
CPX_BDIR_UP 1

Values for CPX_PARAM_MIPSEARCH

CPX_MIPSEARCH_AUTO 0
CPX_MIPSEARCH_TRADITIONAL 1
CPX_MIPSEARCH_DYNAMIC 2

Values for CPX_PARAM_MIPKAPPASTATS

CPX_MIPKAPPA_OFF -1
CPX_MIPKAPPA_AUTO 0
CPX_MIPKAPPA_SAMPLE 1
CPX_MIPKAPPA_FULL 2

Effort levels for MIP starts

CPX_MIPSTART_AUTO 0
CPX_MIPSTART_CHECKFEAS 1
CPX_MIPSTART_SOLVEFIXED 2
CPX_MIPSTART_SOLVEMIP 3
MIP Problem status codes

CPXMIP_OPTIMAL 101
CPXMIP_OPTIMAL_TOL 102
CPXMIP_INFEASIBLE 103
CPXMIP_SOL_LIM 104
CPXMIP_NODE_LIM_FEAS 105
CPXMIP_NODE_LIM_INFEAS 106
CPXMIP_TIME_LIM_FEAS 107
CPXMIP_TIME_LIM_INFEAS 108
CPXMIP_FAIL_FEAS 109
CPXMIP_FAIL_INFEAS 110
CPXMIP_MEM_LIM_FEAS 111
CPXMIP_MEM_LIM_INFEAS 112
CPXMIP_ABORT_FEAS 113
CPXMIP_ABORT_INFEAS 114
CPXMIP_OPTIMAL_INFEAS 115
CPXMIP_FAIL_FEAS_NO_TREE 116
CPXMIP_FAIL_INFEAS_NO_TREE 117
CPXMIP_UNBOUNDED 118
CPXMIP_INFORUNBD 119
CPXMIP_FEASIBLE_RELAXED_SUM 120
CPXMIP_OPTIMAL_RELAXED_SUM 121
CPXMIP_FEASIBLE_RELAXED_INF 122
CPXMIP_OPTIMAL_RELAXED_INF 123
CPXMIP_FEASIBLE_RELAXED_QUAD 124
CPXMIP_OPTIMAL_RELAXED_QUAD 125
CPXMIP_ABORT_RELAXED 126
CPXMIP_FEASIBLE 127
CPXMIP_POPULATESOL_LIM 128
CPXMIP_OPTIMAL_POPULATED 129
CPXMIP_OPTIMAL_POPULATED_TOL 130
CPXMIP_DETTIME_LIM_FEAS 131
CPXMIP_DETTIME_LIM_INFEAS 132

Valid purgeable values for adding usercuts and lazyconstraints

CPX_USECUT_FORCE 0
CPX_USECUT_PURGE 1
For CPXgetnodeintfeas

- CPX_INTEGER_FEASIBLE 0
- CPX_INTEGER_INFEASIBLE 1
- CPX_IMPLIED_INTEGER_FEASIBLE 2

MIP Parameter numbers

- CPX_PARAM_BRDIR 2001
- CPX_PARAM_BTTOL 2002
- CPX_PARAM_CLIQUES 2003
- CPX_PARAM_COEREDIND 2004
- CPX_PARAM_COVERS 2005
- CPX_PARAM_CUTLO 2006
- CPX_PARAM_CUTUP 2007
- CPX_PARAM_EPAGAP 2008
- CPX_PARAM_EPGAP 2009
- CPX_PARAM_EPINT 2010
- CPX_PARAM_MIPDISPLAY 2012
- CPX_PARAM_MIPINTERVAL 2013
- CPX_PARAM_INTSOLLIM 2015
- CPX_PARAM_NODEFILEIND 2016
- CPX_PARAM_NODELIM 2017
- CPX_PARAM_NODESEL 2018
- CPX_PARAM_OBJDIF 2019
- CPX_PARAM_MIPORDIND 2020
- CPX_PARAM_RELOBJDIF 2022
- CPX_PARAM_STARTALG 2025
- CPX_PARAM_SUBALG 2026
- CPX_PARAM_TRELIM 2027
- CPX_PARAM_VARSEL 2028
- CPX_PARAM_BNDSTRENIND 2029
- CPX_PARAM_HEURFREQ 2031
- CPX_PARAM_MIPORDTYPE 2032
- CPX_PARAM_CUTSFAC 2033
- CPX_PARAM_RELAXPREIND 2034
- CPX_PARAM_PRESLVLND 2037
- CPX_PARAM_BBINTERVAL 2039
- CPX_PARAM_FLOWCOVERS 2040
CPX_PARAM_IMPLBD  2041
CPX_PARAM_PROBE  2042
CPX_PARAM_GUBCOVERS  2044
CPX_PARAM_STRONGCANDLIM  2045
CPX_PARAM_STRONGITLIM  2046
CPX_PARAM_FRACCAND  2048
CPX_PARAM_FRACCUTS  2049
CPX_PARAM_FRACPASS  2050
CPX_PARAM_FLOWPATHS  2051
CPX_PARAM_MIRCUTS  2052
CPX_PARAM_DISJCUTS  2053
CPX_PARAM_AGGCUTLIM  2054
CPX_PARAM_MIPCBLREDLP  2055
CPX_PARAM_CUTPASS  2056
CPX_PARAM_MIPEMPHASIS  2058
CPX_PARAM_SYMMETRY  2059
CPX_PARAM_DIVETYPE  2060
CPX_PARAM_RINSHEUR  2061
CPX_PARAM_SUBMIPNODELIM  2062
CPX_PARAM_LBHEUR  2063
CPX_PARAM_REPEATPRESOLVE  2064
CPX_PARAM_PROBETIME  2065
CPX_PARAM_POLISHTIME  2066
CPX_PARAM_REPAIRTRIES  2067
CPX_PARAM_EPLIN  2068
CPX_PARAM_EPRELAX  2073
CPX_PARAM_FPHEUR  2098
CPX_PARAM_EACHCUTLIM  2102
CPX_PARAM_SOLNPOLCAPACITY  2103
CPX_PARAM_SOLNPOLREPLACE  2104
CPX_PARAM_SOLNPOLGAP  2105
CPX_PARAM_SOLNPOLLAGAP  2106
CPX_PARAM_SOLNPOLINTENSITY  2107
CPX_PARAM_POPULATELIM  2108
CPX_PARAM_MIPSEARCH  2109
CPX_PARAM_MIQCPSTRAT  2110
CPX_PARAM_ZEROHALFCUTS  2111
CPX_PARAM_POLISHAFTEREPAGAP  2126
CPX_PARAM_POLISHAFTEREPEGAP  2127
CPX_PARAM_POLISHAFTERNODE  2128
CPX_PARAM_POLISHAFTERINTSOL  2129
CPX_PARAM_POLISHAFTERTIME  2130
CPX_PARAM_MCFCUTS  2134
CPX_PARAM_MIPKAPPASTATS  2137
CPX_PARAM_AUXROOTTHREADS  2139
CPX_PARAM_INTSOLFILEPREFIX  2143
CPX_PARAM_PROBETIME  2150
CPX_PARAM_POLISHAFTERDETTIME  2151
### cplexConstants

- **CPX_PARAM_LANDPCUTS**: 2152
- **CPX_PARAM_RAMPUPDURATION**: 2163
- **CPX_PARAM_RAMPUPDEETILIM**: 2164
- **CPX_PARAM_RAMPUPTILIM**: 2165

#### Values for CPX_PARAM_SOLNPOOLREPLACE

<table>
<thead>
<tr>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_SOLNPOOL_FIFO</td>
<td>0</td>
</tr>
<tr>
<td>CPX_SOLNPOOL_OBJ</td>
<td>1</td>
</tr>
<tr>
<td>CPX_SOLNPOOL_DIV</td>
<td>2</td>
</tr>
<tr>
<td>CPX_SOLNPOOL_FILTER_DIVERSITY</td>
<td>1</td>
</tr>
<tr>
<td>CPX_SOLNPOOL_FILTER_RANGE</td>
<td>2</td>
</tr>
</tbody>
</table>

#### File: gcconst.h

- **CPX_CON_LOWER_BOUND**: 1
- **CPX_CON_UPPER_BOUND**: 2
- **CPX_CON_LINEAR**: 3
- **CPX_CON_QUADRATIC**: 4
- **CPX_CON_SOS**: 5
- **CPX_CON_INDICATOR**: 6

#### internal types

- **CPX_CON_MINEXPR**: 7
- **CPX_CON_MAXEXPR**: 8
- **CPX_CON_PWL**: 9
- **CPX_CON_ABS**: 9
- **CPX_CON_DISJUNCT**: 10
- **CPX_CON_INDDISJUNCT**: 11
- **CPX_CON_SETVAR**: 12
- **CPX_CON_SETVARMEMBER**: 13
- **CPX_CON_SETVARCARD**: 14
- **CPX_CON_SETVARSUM**: 15
- **CPX_CON_SETVARMIN**: 16
- **CPX_CON_SETVARMAX**: 17
- **CPX_CON_SETVARSUBSET**: 18
cplexConstants

CPX_CON_SETVARDOMAIN 19
CPX_CON_SETVARUNION 20
CPX_CON_SETVARINTERSECTION 21
CPX_CON_SETVARNULLINTERSECT 22
CPX_CON_SETVARINTERSECT 23
CPX_CON_SETVAREQ 24
CPX_CON_SETVAREQEQCST 25
CPX_CON_SETVARNEQ 26
CPX_CON_LAST_CONTYPE 27

Network parameters

CPX_PARAM_NETITLIM 5001
CPX_PARAM_NETEPOPT 5002
CPX_PARAM_NETEPRHS 5003
CPX_PARAM_NETPPIIND 5004
CPX_PARAM_NETDISPLAY 5005

NETOPT display values

CPXNET_NO_DISPLAY_OBJECTIVE 0
CPXNET_TRUE_OBJECTIVE 1
CPXNET_PENALIZED_OBJECTIVE 2

NETOPT pricing parameters

CPXNET_PRICE_AUTO 0
CPXNET_PRICE_PARTIAL 1
CPXNET_PRICE_MULT_PART 2
CPXNET_PRICE_SORT_MULT_PART 3

Copying data

CPX_PARAM_QPNZREADLIM 4001
Specify how to calculate duals for QCPs

CPX_PARAM_CALCQCPDUALS 4003

presolve

CPX_PARAM_QPMAKEPSDIND 4010

Error codes

Callable library miscellaneous routines

CPXERR_NEGATIVE_SURPLUS 1207
CPXERR_NO_SENSIT 1260

new parameter names introduced in IBM ILOG CPLEX version 12.6

Callable library miscellaneous routines

CPXPARAM_Advance 1001
CPXPARAM_Barrier_Algorithm 3007
CPXPARAM_Barrier_ColNonzeros 3009
CPXPARAM_Barrier_ConvergeTol 3002
CPXPARAM_Barrier_Crossover 3018
CPXPARAM_Barrier_Display 3010
CPXPARAM_Barrier_Limits_Corrections 3013
CPXPARAM_Barrier_Limits_Growth 3003
CPXPARAM_Barrier_Limits_Iteration 3012
CPXPARAM_Barrier_Limits_ObjRange 3004
CPXPARAM_Barrier_Ordering 3014
CPXPARAM_Barrier_QCPConvergeTol 3020
CPXPARAM_Barrier_StartAlg 3017
CPXPARAM_ClockType 1006
CPXPARAM_Conflict_Display 1074
CPXPARAM_DetTimeLimit 1127
CPXPARAM_DistMIP_Rampup_DetTimeLimit 2164
<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPXPARAM_DistMIP_Rampup_Duration</td>
<td>2163</td>
</tr>
<tr>
<td>CPXPARAM_DistMIP_Rampup_TimeLimit</td>
<td>2165</td>
</tr>
<tr>
<td>CPXPARAM_Emphasis_Memory</td>
<td>1082</td>
</tr>
<tr>
<td>CPXPARAM_Emphasis_MIP</td>
<td>2058</td>
</tr>
<tr>
<td>CPXPARAM_Emphasis_Numerical</td>
<td>1083</td>
</tr>
<tr>
<td>CPXPARAM_Feasopt_Mode</td>
<td>1084</td>
</tr>
<tr>
<td>CPXPARAM_Feasopt_Tolerance</td>
<td>2073</td>
</tr>
<tr>
<td>CPXPARAM_LPMethod</td>
<td>1062</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_Cliques</td>
<td>2003</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_Covers</td>
<td>2005</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_Disjunctive</td>
<td>2053</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_FlowCovers</td>
<td>2040</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_Gomory</td>
<td>2049</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_GUBCovers</td>
<td>2044</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_Implied</td>
<td>2041</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_LiftProj</td>
<td>2152</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_MFCut</td>
<td>2134</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_MIRCut</td>
<td>2052</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_PathCut</td>
<td>2051</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Cuts_ZeroHalfCut</td>
<td>2111</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Display</td>
<td>2012</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Interval</td>
<td>2013</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_AggForCut</td>
<td>2054</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_AuxRootThreads</td>
<td>2139</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_CutPasses</td>
<td>2056</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_CutsFactor</td>
<td>2033</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_EachCutLimit</td>
<td>2102</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_GomoryCand</td>
<td>2048</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_GomoryPass</td>
<td>2050</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_Nodes</td>
<td>2017</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_PolishTime</td>
<td>2066</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_Populate</td>
<td>2108</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_ProbeDetTime</td>
<td>2150</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_ProbeTime</td>
<td>2065</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_RepairTries</td>
<td>2067</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_Solutions</td>
<td>2015</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_StrongCand</td>
<td>2045</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_StrongIt</td>
<td>2046</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_SubMIPNodeLim</td>
<td>2062</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_TreeMemory</td>
<td>2027</td>
</tr>
<tr>
<td>CPXPARAM_MIP_OrderType</td>
<td>2032</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_AbsMIPGap</td>
<td>2126</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_DetTime</td>
<td>2151</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_MIPGap</td>
<td>2127</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_Nodes</td>
<td>2128</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_Solutions</td>
<td>2129</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_Time</td>
<td>2130</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Pool_AbsGap</td>
<td>2106</td>
</tr>
<tr>
<td>cplexConstants</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Pool_Capacity           2103</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Pool_Intensity          2107</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Pool_RelGap             2105</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Pool_Replace            2104</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_Backtrack      2002</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_BBInterval     2039</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_Branch         2001</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_CallbackReducedLP 2055</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_Dive           2060</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_File           2016</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_FPHeur         2098</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_HeuristicFreq  2031</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_KappaStats     2137</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_LBHeur         2063</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_MIQCPStrat     2110</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_NodeSelect     2018</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_Order          2020</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_PresolveNode   2037</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_Probe          2042</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_RINSHeur       2061</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_Search         2109</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_StartAlgorithm 2025</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_SubAlgorithm   2026</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Strategy_VariableSelect 2028</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Tolerances_AbsMIPGap    2008</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Tolerances_Integrality  2010</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Tolerances_LowerCutoff  2006</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Tolerances_MIPGap       2009</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Tolerances_ObjDifference 2019</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Tolerances_RelObjDifference 2022</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_MIP_Tolerances_UpperCutoff  2007</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Network_Display             5105</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Network_Iterations          5001</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Network_NetFind             1022</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Network_Pricing             5004</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Network_Tolerances_Feasibility 5003</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Network_Tolerances_Optimality 5002</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Output_CloneLog             1132</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Output_IntSolFilePrefix     2143</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Output_MPSLong              1081</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Output_WriteLevel           1114</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Parallel                   1109</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Aggregator    1003</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_BoundStrength 2029</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_CoeffReduce   2004</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Dependency    1008</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Dual         1044</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Fill         1002</td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Linear</td>
<td>1058</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_NumPass</td>
<td>1052</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Presolve</td>
<td>1030</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_QCPDuals</td>
<td>4003</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_QPMakePSD</td>
<td>4010</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Reduce</td>
<td>1057</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Relax</td>
<td>2034</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_RepeatPresolve</td>
<td>2064</td>
</tr>
<tr>
<td>CPXPARAM_Preprocessing_Symmetry</td>
<td>2059</td>
</tr>
<tr>
<td>CPXPARAM_QPMethod</td>
<td>1063</td>
</tr>
<tr>
<td>CPXPARAM_RandomSeed</td>
<td>1124</td>
</tr>
<tr>
<td>CPXPARAM_Read_APIEncoding</td>
<td>1130</td>
</tr>
<tr>
<td>CPXPARAM_Read_Constraints</td>
<td>1021</td>
</tr>
<tr>
<td>CPXPARAM_Read_DataCheck</td>
<td>1056</td>
</tr>
<tr>
<td>CPXPARAM_Read_FileEncoding</td>
<td>1129</td>
</tr>
<tr>
<td>CPXPARAM_Read_Nonzeros</td>
<td>1024</td>
</tr>
<tr>
<td>CPXPARAM_Read_QPNonzeros</td>
<td>4001</td>
</tr>
<tr>
<td>CPXPARAM_Read_Scale</td>
<td>1034</td>
</tr>
<tr>
<td>CPXPARAM_Read_Variables</td>
<td>1023</td>
</tr>
<tr>
<td>CPXPARAM_ScreenOutput</td>
<td>1035</td>
</tr>
<tr>
<td>CPXPARAM_Sifting_Algorithm</td>
<td>1077</td>
</tr>
<tr>
<td>CPXPARAM_Sifting_Display</td>
<td>1076</td>
</tr>
<tr>
<td>CPXPARAM_Sifting_Iterations</td>
<td>1078</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Crash</td>
<td>1007</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_DGradient</td>
<td>1009</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Display</td>
<td>1019</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Limits_Iterations</td>
<td>1020</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Limits_LowerObj</td>
<td>1025</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Limits_Perturbation</td>
<td>1028</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Limits_Singularity</td>
<td>1037</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Limits_UpperObj</td>
<td>1026</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Perturbation_Constant</td>
<td>1015</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Perturbation_Indicator</td>
<td>1027</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_PGradient</td>
<td>1029</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Pricing</td>
<td>1010</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Refactor</td>
<td>1031</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Tolerances_Feasibility</td>
<td>1016</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Tolerances_Markowitz</td>
<td>1013</td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Tolerances_Optimality</td>
<td>1014</td>
</tr>
<tr>
<td>CPXPARAM_SolutionTarget</td>
<td>1131</td>
</tr>
<tr>
<td>CPXPARAM_Threads</td>
<td>1067</td>
</tr>
<tr>
<td>CPXPARAM_Timelimit</td>
<td>1039</td>
</tr>
<tr>
<td>CPXPARAM_Tune_DetTimeLimit</td>
<td>1139</td>
</tr>
<tr>
<td>CPXPARAM_Tune_Display</td>
<td>1113</td>
</tr>
<tr>
<td>CPXPARAM_Tune_Measure</td>
<td>1110</td>
</tr>
<tr>
<td>CPXPARAM_Tune_Repeat</td>
<td>1111</td>
</tr>
<tr>
<td>CPXPARAM_Tune_Timelimit</td>
<td>1112</td>
</tr>
<tr>
<td>CPXPARAM_WorkDir</td>
<td>1064</td>
</tr>
</tbody>
</table>
Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

return_codeCPLEX, status_codeCPLEX, getParmValCPLEX

cplexError-class

Class "cplexError"

Description

Objects of class cplexerr are returned by various functions of cplexAPI, in order to distinguish a status (error) code from a successful result.

Objects from the Class

Objects can be created by calls of the form cplexError(err), with err being an error code of IBM ILOG CPLEX.

Slots

ernum: Object of class "integer" containing the error code.

Methods

err signature(object = "cplexError"): Prints an error message string corresponding to the error code.
errmsg signature(object = "cplexError"): Returns an error message string corresponding to the error code.
ernum signature(object = "cplexError"): Gets the error code.
ernum<- signature(object = "cplexError"): Sets the error code.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
cplexPtr-class

References

cplexPtr-class  Class "cplexPtr"

Description
Structure of the class "cplexPtr". Objects of that class are used to hold pointers to C structures used by IBM ILOG CPLEX.

Objects from the Class
Objects can be created by calls of the form
env <- openEnvCPLEX() and/or
prob <- initProbCPLEX(env)).

Slots
cplexPtrType: Object of class "character" giving the pointer type.
cplexPointer: Object of class "externalptr" containing the pointer to a C structure.

Methods
isCPLEXchanPointer signature(object = "cplexPtr") returns TRUE if cplexPointer(object) is a pointer to a CPLEX channel, otherwise FALSE.
isCLEXenvPointer signature(object = "cplexPtr") returns TRUE if cplexPointer(object) is a pointer to a CPLEX environment, otherwise FALSE.
isCPLEXfilePointer signature(object = "cplexPtr") returns TRUE if cplexPointer(object) is a pointer to a CPLEX file, otherwise FALSE.
isCPLEXprobPointer signature(object = "cplexPtr") returns TRUE if cplexPointer(object) is a pointer to a CPLEX problem object, otherwise FALSE.
isCLEXtermPointer signature(object = "cplexPtr") returns TRUE if cplexPointer(object) is a pointer to a CPLEX termination signal, otherwise FALSE.
isNULLpointerCLEX signature(object = "cplexPtr") returns TRUE if cplexPointer(object) is a NULL pointer, otherwise FALSE.
cplexPointer signature(object = "cplexPtr") gets the cplexPointer slot.
summary signature(object = "cplexPtr") prints a summary of the problem object to the command line. If a solution is available, it prints also information retrieved by solutionCPLEX and solnInfoCPLEX. If no solution is available, it prints the corresponding error message. The method returns invisibly NULL. The CPLEX environment pointer is needed as second argument env to summary.
cplexPtrType signature(object = "cplexPtr") gets the cplexPtrType slot.
cplexPtrType<- signature(object = "cplexPtr") sets the cplexPtrType slot.
Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
openEnvCplex and initProbCplex

---

delColsCplex Delete all Columns in a Specified Range

Description
Low level interface function to the IBM ILOG CPLEX function CPXdelcols. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
delColsCplex(env, lp, begin, end)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>env</td>
<td>An object of class &quot;cplexPtr&quot; as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.</td>
</tr>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
<tr>
<td>begin</td>
<td>Integer value, numeric index of the first column to be deleted.</td>
</tr>
<tr>
<td>end</td>
<td>Integer value, numeric index of the last column to be deleted.</td>
</tr>
</tbody>
</table>

Details
Interface to the C function delCols which calls the CPLEX function CPXdelcols.

Value
Zero if successful, otherwise nonzero.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
References


---

delFpDestCPLEX Remove a File from the List of Message Destinations for a Channel

Description

Low level interface function to the IBM ILOG CPLEX function CPXdelfpdest. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

delFpDestCPLEX(env, newch, cpfile)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- newch: A pointer to the channel for which destinations are to be deleted as returned by CPXaddchannel.
- cpfile: Pointer to an IBM ILOG CPLEX file as returned by openFileCplex.

Details

Interface to the C function delFpDest which calls the CPLEX function CPXdelfpdest.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

addFpDestCPLEX
delIndConstrsCPLEX  
*Delete a Range of Indicator Constraints*

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXdelindconstrs. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```c
delIndConstrsCPLEX(env, lp, begin, end)
```

**Arguments**

- **env**: An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **begin**: An integer that specifies the numeric index of the first indicator constraint to be deleted.
- **end**: An integer that specifies the numeric index of the last indicator constraint to be deleted.

**Details**

Interface to the C function `delIndConstrs` which calls the CPLEX function CPXdelindconstrs.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

Delete a Range MIP Starts

Description

Low level interface function to the IBM ILOG CPLEX function CPLEXdelmipstarts. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

delmipstartscplex(env, lp, begin, end)

Arguments

env 
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp 
An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

begin 
An integer specifying the numeric index of the first MIP start to be deleted.

dend 
An integer specifying the numeric index of the last MIP start to be deleted.

Details

Interface to the C function delMIPstarts which calls the CPLEX function CPLEXdelmipstarts.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

delNamesCPLEX

Remove all Names Assigned to Rows and Columns

Description

Low level interface function to the IBM ILOG CPLEX function CPXdelnames. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

delNamesCPLEX(env, lp)

Arguments

env  
An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp  
An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function delNames which calls the CPLEX function CPXdelnames.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

delProbCplex

Remove Specified CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function cpxfreeprob. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

delProbCplex(env, lp)

Arguments

env | An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp | An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function delProb which calls the CPLEX function CPXfreeprob.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

initProbCplex
delQConstrsCplex  

Delete a Range of Quadratic Constraints

Description

Low level interface function to the IBM ILOG CPLEX function CPXdelqconstrs. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```
delQConstrsCplex(env, lp, begin, end)
```

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **begin**: An integer that specifies the numeric index of the first quadratic constraint to be deleted.
- **end**: An integer that specifies the numeric index of the last quadratic constraint to be deleted.

Details

Interface to the C function delQConstrs which calls the CPLEX function CPXdelqconstrs.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

delRowsCplex

Delete a Range of Rows

Description

Low level interface function to the IBM ILOG CPLEX function CPXdelrows. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

`delRowsCplex(env, lp, begin, end)`

Arguments

- **env**: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **begin**: Integer value, numeric index of the first row to be deleted.
- **end**: Integer value, numeric index of the last row to be deleted.

Details

Interface to the C function `delRows` which calls the CPLEX function CPXdelrows.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Delete a Set of Columns

Description

Low level interface function to the IBM ILOG CPLEX function CPXdelsetcols. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```r
delSetColsCplex(env, lp, delstat)
```

Arguments

- `env` An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `delstat` An array specifying the columns to be deleted.

Details

Interface to the C function `delSetCols` which calls the CPLEX function CPXdelsetcols.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**delSetRowsCPLEX**  
*Delete a Set of Rows*

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXdelsetrows. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```
delSetRowsCLEX(env, lp, delstat)
```

**Arguments**

- **env**  
  An object of class "cplexPtr" as returned by openEnvCLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

- **lp**  
  An object of class "cplexPtr" as returned by initProbCLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

- **delstat**  
  An array specifying the rows to be deleted.

**Details**

Interface to the C function delSetRows which calls the CPLEX function CPXdelsetrows.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

delTerminateCPLEX  Terminate CPLEX gracefully

Description
Low level interface function to the IBM ILOG CPLEX function CPXsetterminate. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
   delTerminateCPLEX(env, tsig)

Arguments
   env   An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
   tsig  Pointer to termination signal as returned by setTerminateCPLEX.

Details
Interface to the C function setTerminate which calls the CPLEX function CPXsetterminate.

Value
Zero if successful, otherwise nonzero.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
setTerminateCPLEX, printTerminateCPLEX, chgTerminateCPLEX
disconnectChannelCPLex

Flush all Message Destinations Associated with a Channel

Description

Low level interface function to the IBM ILOG CPLEX function CPXdisconnectchannel. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

disconnectChannelCPLex(env, newch)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLex. This is basically a pointer to an IBM ILOG CPLEX environment.
newch A pointer to the channel containing the message destinations as returned by CPXaddchannel.

Details

Interface to the C function disconnectChannel which calls the CPLEX function CPXdisconnectchannel.

Value

NULL

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

flushChannelCPLex, flushStdChannelsCPLex, getChannelsCPLex
dualoptCplex  

Find a Problem Solution Using the Dual Simplex Algorithm

Description

Low level interface function to the IBM ILOG CPLEX function CPXdualopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

dualoptCplex(env, lp)

Arguments

env  
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp  
An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function dualOpt which calls the CPLEX function CPXdualopt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

solnInfoCplex, getStatCplex, solutionCplex
dualWriteCPLEX  Write a Dual Formulation of the Current CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXdualwrite. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

dualWriteCPLEX(env, lp, fname)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp  An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

fname  Single character value giving the filename to write to.

Details

Interface to the C function dualWrite which calls the CPLEX function CPXdualwrite.

Value

Zero if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldor.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldor.de>

References

 feasOptCplex  

Compute a Minimum-Cost Relaxation

Description

Low level interface function to the IBM ILOG CPLEX function CPXfeasopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

feasOptCplex(env, lp, rhs = FALSE, rng = FALSE, lb = FALSE, ub = FALSE)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- rhs: If set to FALSE no right hand side value is allowed to be relaxed.
- rng: If set to FALSE no range values are allowed to be relaxed.
- lb: If set to FALSE no lower bound of any variable is allowed to be relaxed.
- ub: If set to FALSE no lower bound of any variable is allowed to be relaxed.

Details

Interface to the C function feasOpt which calls the CPLEX function CPXfeasopt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

solutionCplex, getRowInfeasCplex, getColInfeasCplex, solnInfoCplex, getStatCplex
Description

Low level interface function to the IBM ILOG CPLEX function CPXfputs. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

fileputcplex(cpfile, stuff = "")

Arguments

cpfile A pointer to a file as returned by openFileCPLEX.
stuff A character string to be written to the file.

Details

Interface to the C function fileput which calls the CPLEX function CPXfputs.

Value

A nonnegative value if successful, otherwise EOF.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

closeFileCPLEX, openFileCPLEX
flushChannelCPLEX  
Flush All Message Destinations Associated With a Channel

Description
Low level interface function to the IBM ILOG CPLEX function cpxflushchannel. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
flushChannelCPLEX(env, newch)

Arguments
- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- newch: Pointer to a channel object as returned by addChannelCPLEX.

Details
Interface to the C function flushChannel which calls the CPLEX function CPXflushchannel.

Value
NULL

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
disconnectChannelCPLEX, flushStdChannelsCPLEX, getChannelsCPLEX
flushStdChannels CPLEX  Flushes the Output Buffers of the Four Standard Channels

Description

Low level interface function to the IBM ILOG CPLEX function CPXflushstdchannels. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

flushStdChannelsCPLEX(env)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

Details

Interface to the C function flushStdChannels which calls the CPLEX function CPXflushstdchannels.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

disconnectChannelCPLEX, flushChannelCPLEX, getChannelsCPLEX
freePresolveCplex


desc

Low level interface function to the IBM ILOG CPLEX function CPXfreepresolve. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

freePresolveCplex(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function freePresolve which calls the CPLEX function CPXfreepresolve.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getBaseCPLEX

Access Basis Resident in a CPLEX Problem Object.

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXgetbase. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

getBaseCPLEX(env, lp)

**Arguments**

- **env**: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

**Details**

Interface to the C function getBase which calls the CPLEX function CPXgetbase.

**Value**

If successful a list is returned:

- **cstat**: basis status of the columns in the CPLEX problem object
- **rstat**: basis status of the artificial, slack, or surplus variable associated with each row in the constraint matrix

otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getBestObjValCPLEX

Access the Currently Best Known Bound of all the Remaining Open Nodes in a Branch-And-Cut Tree

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetbestobjval. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getBestObjValCPLEX(env, lp)

Arguments

 env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

 lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getBestObjVal which calls the CPLEX function CPXgetbestobjval.

Value

Objective value if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getChannelsCPLEX Obtain Pointers to the Four Default Channels

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetchannels. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getchannelsCPLEX(env, ptrtype = "cplex_chan")

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- ptrtype: A name for the pointer object.

Details

Interface to the C function getChannels which calls the CPLEX function CPXgetchannels.

Value

If successful a list is returned:

- cpxresults: address of the channel corresponding to cpxresults
- cpxwarning: address of the channel corresponding to cpxwarning
- cpxerror: address of the channel corresponding to cpxerror
- cpxlog: address of the channel corresponding to cpxlog

otherwise an instance of class "cplexError". Each list element is an object of class "cplexPtr".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

disconnectChannelCPLEX, flushChannelCPLEX, flushStdChannelsCPLEX
getChgParmCplex - Get Parameter Numbers for Parameters Which are Not Set at Their Default Values

Description

Low level interface function to the IBM ILOG CPLEX function CPLEXgetchgparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getChgParmCplex(env)

Arguments

env

An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

Details

Interface to the C function getChgParm which calls the CPLEX function CPLEXgetchgparam.

Value

A vector containing integer values (unique parameter identifiers) for parameters which are not set at their default values, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
**getCoefCPLEX**  
*Access a Single Constraint Matrix Coefficient*

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXget_coef. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
getCoefCPLEX(env, lp, i, j)
```

**Arguments**

- `env` An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `i` An integer specifying the numeric index of the row.
- `j` An integer specifying the numeric index of the column.

**Details**

Interface to the C function `get_coef` which calls the CPLEX function CPXget_coef.

**Value**

Matrix coefficient value if successful, otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getColIndexCPLEX \hspace{2cm} \textit{Search for the Index Number of the Specified Column}

\textbf{Description}

Low level interface function to the IBM ILOG CPLEX function CPXgetcolindex. Consult the IBM ILOG CPLEX documentation for more detailed information.

\textbf{Usage}

\begin{verbatim}
getColIndexCPLEX(env, lp, cname)
\end{verbatim}

\textbf{Arguments}

- \texttt{env} \hspace{1cm} An object of class "\texttt{cplexPtr}" as returned by \texttt{openEnvCPLEX}. This is basically a pointer to an IBM ILOG CPLEX environment.
- \texttt{lp} \hspace{1cm} An object of class "\texttt{cplexPtr}" as returned by \texttt{initProbCPLEX}. This is basically a pointer to an IBM ILOG CPLEX problem object.
- \texttt{cname} \hspace{1cm} A column name to search for.

\textbf{Details}

Interface to the C function \texttt{getColIndex} which calls the CPLEX function CPXgetcolindex.

\textbf{Value}

Column number if successful, otherwise an instance of class "\texttt{cplexError}".

\textbf{Author(s)}

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

\textbf{References}

The IBM ILOG CPLEX home page at \url{https://www.ibm.com/developerworks/university/academicinitiative/}. 
getColInfeasCPLEX  Compute Infeasibility of a Given Solution for a Range of Variables

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetcolinfeas. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getColInfeasCPLEX(env, lp, begin, end, sol = NULL)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **begin**: An integer specifying the beginning of the range of variables whose infeasibility is to be returned.
- **end**: An integer specifying the end of the range of variables whose infeasibility is to be returned.
- **sol**: The solution whose infeasibility is to be computed.

Details

Interface to the C function getColInfeas which calls the CPLEX function CPXgetcolinfeas.

Value

infeasibility values if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Access a Range of Column Names

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetcolname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getColNameCplex(env, lp, begin, end)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **begin**: An integer specifying the beginning of the range of column names to be returned.
- **end**: An integer specifying the end of the range of column names to be returned.

Details

Interface to the C function getColName which calls the CPLEX function CPXgetcolname.

Value

Column names if successful, otherwise an instance of class "cplexError".

Author(s)

- Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
- Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getColsCPLEX \hspace{1em} \textit{Accesses a Range of Columns of the Constraint Matrix}

\textbf{Description}

Low level interface function to the IBM ILOG CPLEX function CPXgetcols. Consult the IBM ILOG CPLEX documentation for more detailed information.

\textbf{Usage}

\begin{verbatim}
getColsCPLEX(env, lp, begin, end)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
    \item \texttt{env} \hspace{1em} An object of class "\texttt{cplexPtr}" as returned by \texttt{openEnvCPLEX}. This is basically a pointer to an IBM ILOG CPLEX environment.
    \item \texttt{lp} \hspace{1em} An object of class "\texttt{cplexPtr}" as returned by \texttt{initProbCPLEX}. This is basically a pointer to an IBM ILOG CPLEX problem object.
    \item \texttt{begin} \hspace{1em} An integer specifying the beginning of the range of columns to be returned.
    \item \texttt{end} \hspace{1em} An integer specifying the end of the range of columns to be returned.
\end{itemize}

\textbf{Details}

Interface to the C function \texttt{getCols} which calls the CPLEX function CPXgetcols.

\textbf{Value}

If successful a list is returned:

\begin{itemize}
    \item \texttt{matbeg} \hspace{1em} Array that specifies the nonzero elements of the columns. Consult the IBM ILOG CPLEX documentation for more detailed information.
    \item \texttt{matind} \hspace{1em} Array that specifies the nonzero elements of the columns. Consult the IBM ILOG CPLEX documentation for more detailed information.
    \item \texttt{matval} \hspace{1em} Array that specifies the nonzero elements of the columns. Consult the IBM ILOG CPLEX documentation for more detailed information.
\end{itemize}

otherwise an instance of class "\texttt{cplexError}".

\textbf{Author(s)}

Gabriel Gelius-Dietrich <geliudie@uni-duesseldor.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

\textbf{References}

The IBM ILOG CPLEX home page at \url{https://www.ibm.com/developerworks/university/academicinitiative/}.  

getColTypeCplex {Access Types for a Range of Variables}

Description
Low level interface function to the IBM ILOG CPLEX function CPXgetctype. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
getColTypeCplex(env, lp, begin, end)

Arguments
- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- begin: An integer specifying the beginning of the range of the types to be returned.
- end: An integer specifying the end of the range of the types to be returned.

Details
Interface to the C function getColType which calls the CPLEX function CPXgetctype.

Value
Column types if successful, otherwise an instance of class "cplexError".

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
cplexConstants
getConflictCplex

Return Linear Constraints and Variables Belonging to a Conflict

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetconflict. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getConflictCplex(env, lp)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getConflict which calls the CPLEX function CPXgetconflict.

Value

If successful a list is returned:

- confstat: status of the conflict
- confnumrows: number of rows in the conflict
- rowind: indices of the constraints that participate in the conflict
- rowbdstat: conflict status of the rows
- confnumcols: number of columns in the conflict
- colind: indices of the variables that participate in the conflict
- colbdstat: conflict status of the columns

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getConflictExtCplex

Get Conflict Status Codes

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetconflictext. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getConflictExtCplex(env, lp, begin, end)

Arguments

env    An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp     An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
begin  The index of the first group.
end    The index of the last group.

Details

Interface to the C function getConflictExt which calls the CPLEX function CPXgetconflictext.

Value

Specified values denoting the conflict status if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getCutoffCPLEX

Access MIP Cutoff Value Being Used During Mixed Integer Optimization.

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetcutoff. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getcutoffCPLEX(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getCutoff which calls the CPLEX function CPXgetcutoff.

Value

Value of the cutoff if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**getDblParmCplex**

Obtain the Current Value of a CPLEX Parameter of Type Double

**Description**

Low level interface function to the IBM ILOG CPLEX function `cpxgetdblparam`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
getDblParmCplex(env, parm)
```

**Arguments**

- `env`: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `parm`: Constant or reference number of the desired parameter.

**Details**

Interface to the C function `getDblParm` which calls the CPLEX function `CPXgetdblparam`.

**Value**

Parameter value if successful, otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

- `cplexConstants`
getDblQualCplex

Access Double-Valued Information About the Quality of the Current Solution of a Problem

Description
Low level interface function to the IBM ILOG CPLEX function CPXgetdblquality. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
getDblQualCplex(env, lp, w)

Arguments
- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **w**: An Integer specifying the quality value to be retrieved.

Details
Interface to the C function getDblQual which calls the CPLEX function CPXgetdblquality.

Value
Requested quality value if successful, otherwise an instance of class "cplexError".

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
cplexConstants
getDbsCntCplex

Access the Number of Dual Super-Basic Variables in the Current Solution

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetdsbcnt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getDbsCntCplex(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getDbsCnt which calls the CPLEX function CPXgetdsbcnt.

Value

Number of dual super-basic variables if a solution exists, otherwise zero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetdj. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getDjCplex(env, lp, begin, end)

Arguments

env
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp
An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

begin
An integer specifying the beginning of the range of reduced-cost values to be returned.

end
An integer specifying the end of the range of reduced-costs values to be returned.

Details

Interface to the C function getCplex which calls the CPLEX function CPXgetdj.

Value

Reduced costs if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getErrorStrCPLEX  
*Return an Error Message String Corresponding to an Error Code*

**Description**

Low level interface function to the IBM ILOG CPLEX function `CplexGetErrorMessage`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```c
getErrorStrCPLEX(err, env = NULL)
```

**Arguments**

- `err`  
  The error code to be translated.

- `env`  
  An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.

**Details**

Interface to the C function `getErrorStr` which calls the CPLEX function `CplexGetErrorMessage`.

**Value**

A single character value containing the error message string.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

- `getStatStrCPLEX`
getGradCpLEX

Project the Impact of Making Changes to Optimal Variable Values or Objective Function Coefficients

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetgrad. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getGradCpLEX(env, lp, j)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>env</td>
<td>An object of class &quot;cplexPtr&quot; as returned by openEnvCpLEX. This is basically a pointer to an IBM ILOG CPLEX environment.</td>
</tr>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCpLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
<tr>
<td>j</td>
<td>An integer specifying the index of the column of interest.</td>
</tr>
</tbody>
</table>

Details

Interface to the C function getGrad which calls the CPLEX function CPXgetgrad.

Value

If successful a list is returned:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>head</td>
<td>listing of the indices of the basic variables in the order in which they appear in the basis.</td>
</tr>
<tr>
<td>y</td>
<td>coefficients of the j-th column relative to the current basis.</td>
</tr>
</tbody>
</table>

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getIndConstrCplex

getIndConstrCplex \hspace{1cm} \textit{Access a Specified Indicator Constraint on the Variables of a CPLEX Problem Object.}

\textbf{Description}

Low level interface function to the IBM ILOG CPLEX function \texttt{CPXgetindconstr}. Consult the IBM ILOG CPLEX documentation for more detailed information.

\textbf{Usage}

\begin{verbatim}
getIndConstrCplex(env, lp, which)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
  \item \texttt{env} \hspace{1cm} An object of class "\texttt{cplexPtr}" as returned by \texttt{openEnvCplex}. This is basically a pointer to an IBM ILOG CPLEX environment.
  \item \texttt{lp} \hspace{1cm} An object of class "\texttt{cplexPtr}" as returned by \texttt{initProbCplex}. This is basically a pointer to an IBM ILOG CPLEX problem object.
  \item \texttt{which} \hspace{1cm} An integer specifying which indicator constraint to return.
\end{itemize}

\textbf{Details}

Interface to the C function \texttt{getIndConstr} which calls the CPLEX function \texttt{CPXgetindconstr}.

\textbf{Value}

If successful a list is returned:

\begin{itemize}
  \item \texttt{indvar} \hspace{1cm} Index of the binary indicator variable. Consult the IBM ILOG CPLEX documentation for more detailed information.
  \item \texttt{complemented} \hspace{1cm} Boolean value that specifies whether the indicator variable is complemented. Consult the IBM ILOG CPLEX documentation for more detailed information.
  \item \texttt{rhs} \hspace{1cm} Righthand side value of the linear portion of the indicator constraint. Consult the IBM ILOG CPLEX documentation for more detailed information.
  \item \texttt{sense} \hspace{1cm} Sense of the linear portion of the constraint. Consult the IBM ILOG CPLEX documentation for more detailed information.
  \item \texttt{linind} \hspace{1cm} Variable indices of the entries of \texttt{linval}. Consult the IBM ILOG CPLEX documentation for more detailed information.
  \item \texttt{linval} \hspace{1cm} Coefficients of the linear portion of the specified indicator constraint. Consult the IBM ILOG CPLEX documentation for more detailed information.
\end{itemize}

otherwise an instance of class "\texttt{cplexError}".
**getInfoDblParmCplex**

**Author(s)**
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

---

**getDescription**

Obtain Default, Minimum and Maximum Values of a Parameter of Type Double

**Description**
Low level interface function to the IBM ILOG CPLEX function CPLEXinfodblparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**
getInfoDblParmCplex(env, parm)

**Arguments**

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- parm: Constant or reference number of the desired parameter.

**Details**
Interface to the C function getInfoDblParm which calls the CPLEX function CPLEXinfodblparam.

**Value**
If successful a list is returned:

- defvalue: default value
- minvalue: minimum value
- maxvalue: maximum value

otherwise an instance of class "cplexError".

**Author(s)**
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
getInfoIntParmCplex

Obtain Default, Minimum and Maximum Values of a Parameter of Type CPXINT

Description

Low level interface function to the IBM ILOG CPLEX function CPXinfointparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getInfoIntParmCplex(env, parm)

Arguments

| env      | An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment. |
| parm     | Constant or reference number of the desired parameter. |

Details

Interface to the C function getInfoIntParm which calls the CPLEX function CPXinfointparam.

Value

If successful a list is returned:

defvalue  default value
minvalue   minimum value
maxvalue   maximum value

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
getInfoLongParmCplex

References

See Also
cplexConstants

getInfoLongParmCplex Obtain Default, Minimum and Maximum Values of a Parameter of Type CPXLONG

Description
Low level interface function to the IBM ILOG CPLEX function CPXinfolongparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
getInfoLongParmCplex(env, parm)

Arguments
env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
parm Constant or reference number of the desired parameter.

Details
Interface to the C function getInfoLongParm which calls the CPLEX function CPXinfolongparam.

Value
If successful a list is returned:
defvalue default value
minvalue minimum value
maxvalue maximum value

otherwise an instance of class "cplexError".

Note
In order to get a 64 bit integer value from CPXinfolongparam, datatype numeric is used. All return values will be numeric.
getInfoStrParmCPLEX

Obtain Default Value of a String Parameter

Description
Low level interface function to the IBM ILOG CPLEX function CPXinfostrparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
getInfoStrParmCPLEX(env, parm)

Arguments
env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
parm Constant or reference number of the desired parameter.

Details
Interface to the C function getInfoStrParm which calls the CPLEX function CPXinfostrparam.

Value
A single character value.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References
**getIntParmCplex**

Obtain the Current Value of a CPLEX Parameter of Type CPXINT

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXgetintparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
getIntParmCplex(env, parm)
```

**Arguments**

- `env`: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `parm`: Constant or reference number of the desired parameter.

**Details**

Interface to the C function `getIntParm` which calls the CPLEX function CPXgetintparam.

**Value**

Parameter value if successful, otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

`cplexConstants`
**getIntQualCplex**

*Access Integer-Valued Information About the Quality of the Current Solution of a Problem*

**Description**

Low level interface function to the IBM ILOG CPLEX function `cpxgetintquality`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```c
getIntQualCplex(env, lp, w)
```

**Arguments**

- **env**: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **w**: An Integer specifying the quality value to be retrieved.

**Details**

Interface to the C function `getIntQual` which calls the CPLEX function `CPXgetintquality`.

**Value**

Requested quality value if successful, otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getItCntCplex

Access the Total Number of Simplex Iterations to Solve an LP Problem

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetitcnt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getItCntCplex(env, lp)

Arguments

env
An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp
An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getItCnt which calls the CPLEX function CPXgetitcnt.

Value

Total iteration count if solution exists, otherwise zero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getLogFileCPLEX  

**Access log file to Which Messages are Written**

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXgetlogfile. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
getLogFileCPLEX(env, ptrtype = "cplex_file")
```

**Arguments**

- `env`  
  An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.

- `ptrtype`  
  A name for the pointer object.

**Details**

Interface to the C function `getLogFile` which calls the CPLEX function CPXgetlogfile.

**Value**

If successful, a pointer to the CPLEX file is return (an instance of class "cplexPtr"), otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

`setLogFileCPLEX`
**getLongParmCplex**

Obtain Current Value of a Parameter of Type CPXLONG

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXgetlongparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```
getLongParmCplex(env, parm)
```

**Arguments**

- **env**
  An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

- **parm**
  Constant or reference number of the desired parameter.

**Details**

Interface to the C function getLongParm which calls the CPLEX function CPXgetlongparam.

**Value**

Parameter value if successful, otherwise an instance of class "cplexError".

**Note**

In order to get a 64 bit integer value from CPXgetlongparam, datatype numeric is used. The return value will be numeric.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

getIntParmCplex, cplexConstants
getLowBndsIdsCplex

Retrieve Lower Bounds on Variables

Description

The function retrieves the lower bounds on specified variables.

Usage

getLowBndsIdsCplex(env, lp, ind)

Arguments

denv                An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
dlp                 An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
dind                Column indices of variables (remember: first index is 0).

Value

A numeric vector containing the lower bounds on the specified variables. If not successfull an instance of class "cplexError" is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

getLowerBndsCplex
getLowerBndsCplex

Access a Range of Lower Bounds on Variables

Description

Low level interface function to the IBM ILOG CPLEX function cpxgetlb. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getLowerBndsCplex(env, lp, begin, end)

Arguments

env
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp
An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

begin
Beginning of the range of lower bounds to be returned.

dead
End of the range of lower bounds to be returned.

Details

Interface to the C function getLowerBnds which calls the CPLEX function CPXgetlb.

Value

A numeric vector containing the lower bounds on the specified variables. If not successfull an instance of class "cplexError" is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getMethodCPLEX  

Obtain Solution Algorithm

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetmethod. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getmethodcplex(env, lp)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getMethod which calls the CPLEX function CPXgetmethod.

Value

A single integer value specifying the solution algorithm.

Author(s)

- Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
- Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants section “LP/QP solution algorithms”.
getMIPrelGapCPLEX

Access Relative Objective Gap for a MIP Optimization

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetmiprelgap. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getMIPrelGapCPLEX(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getMIPrelGap which calls the CPLEX function CPXgetmiprelgap.

Value

Relative Objective Gap if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

getObjValCPLEX, getBestObjValCPLEX
getMIPstartIndexCplex  

Search for the Index Number of the Specified MIP Start

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetmipstartindex. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getMIPstartIndexCplex(env, lp, iname)

Arguments

ev  An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp  An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
iname  A MIP start name to search for.

Details

Interface to the C function getMIPstartIndex which calls the CPLEX function CPXgetmipstartindex.

Value

Index number of the specified MIP start if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**getMIPstartName**

*Access a Range of Names of MIP Starts*

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXgetmipstartname`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
getMIPstartNameCPLEX(env, lp, begin, end)
```

**Arguments**

- **env**
  - An object of class "`cplexPtr`" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.

- **lp**
  - An object of class "`cplexPtr`" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.

- **begin**
  - An integer specifying the beginning of the range of MIP starts to be returned.

- **end**
  - An integer specifying the end of the range of MIP starts to be returned.

**Details**

Interface to the C function `getMIPstartName` which calls the CPLEX function `CPXgetmipstartname`.

**Value**

Names of the MIP starts if successful, otherwise an instance of class "`cplexError`".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getMIPstartsCPLUS

Access a Range of MIP Starts of a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function cpxGetMipstarts. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getMIPstartsCPLUS(env, lp, begin, end)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLUS. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCPLUS. This is basically a pointer to an IBM ILOG CPLEX problem object.
begin An integer specifying the beginning of the range of MIP starts to be returned.
end An integer specifying the end of the range of MIP starts to be returned.

Details

Interface to the C function getMIPstarts which calls the CPLEX function cpxGetMipstarts.

Value

If successful a list is returned:

beg Array specifying where each of the requested MIP starts begins in the arrays varindices and values. Consult the IBM ILOG CPLEX documentation for more detailed information.
varindices Array containing the numeric indices of the columns corresponding to the variables which are assigned starting values. Consult the IBM ILOG CPLEX documentation for more detailed information.
values Array containing the values of the MIP starts. Consult the IBM ILOG CPLEX documentation for more detailed information.
effortlevel Array containing the effort level for each MIP start requested. Consult the IBM ILOG CPLEX documentation for more detailed information.

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gellius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
getNumColsCPLEX

References


---

getNumColsCPLEX  Access the Number of Columns in the Constraint Matrix

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetnumcols. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getNumColsCPLEX(env, lp)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getNumCols which calls the CPLEX function CPXgetnumcols.

Value

If successful the number of variables is returned. If env or lp do not exist, zero is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getNumMIPstartsCplex

Access the Number of MIP Starts in the CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetnummipstarts. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getNumMIPstartsCplex(env, lp)

Arguments

env
An object of class "cpexpr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp
An object of class "cpexpr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getNumMIPstarts which calls the CPLEX function CPXgetnummipstarts.

Value

If successful the number of MIP starts is returned. If env or lp do not exist, zero is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getNumNnzCPLY

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXgetnumnz. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
getNumNnzCPLY(env, lp)
```

**Arguments**

- `env` An object of class "cpplexPtr" as returned by openEnvCPLY. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cpplexPtr" as returned by initProbCPLY. This is basically a pointer to an IBM ILOG CPLEX problem object.

**Details**

Interface to the C function getNumNnz which calls the CPLEX function CPXgetnumnz.

**Value**

Zero if the problem object or environment does not exist, otherwise the number of nonzero elements.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getNumQConstrsCplex

Return the Number of quadratic constraints.

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetnumqconstrs. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getNumQConstrsCplex(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getNumQConstrs which calls the CPLEX function CPXgetnumqconstrs.

Value

If successful the number of quadratic constraints is returned. If env or lp do not exist, zero is returned.

Author(s)

Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getNumQPnzCPLex

Return the Number of Nonzeros in the Q Matrix

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetnumqpnz. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getNumQPnzCPLex(env, lp)

Arguments

env
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp
An object of class "cplexPtr" as returned by initProbCPLex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getNumQPtr which calls the CPLEX function CPXgetnumqpnz.

Value

If successful the number of nonzeros in the Q matrix is returned. If env or lp do not exist, zero is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getNumQuadCplex

Return the Number of Variables That Have Quadratic Objective Coefficients

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetnumquad. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getNumQuadCplex(env, lp)

Arguments

env  
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp  
An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getNumQuad which calls the CPLEX function CPXgetnumquad.

Value

If successful the number of variables that have quadratic objective coefficients is returned. If env or lp do not exist, zero is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getNumRowsCplex

Access the Number of Rows in the Constraint Matrix

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetnumcols. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

ggetNumRowsCplex(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getNumRows which calls the CPLEX function CPXgetnumrows.

Value

If successful the number of rows is returned. If env or lp do not exist, zero is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getObjCplex

Access a Range of Objective Function Coefficients of a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetobj. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getObjCplex(env, lp, begin, end)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **begin**: An integer specifying the beginning of the range of objective function coefficients to be returned.
- **end**: An integer specifying the end of the range of objective function coefficients to be returned.

Details

Interface to the C function getobj which calls the CPLEX function CPXgetobj.

Value

Specified objective coefficients if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Access the Direction of Optimization

Description
Low level interface function to the IBM ILOG CPLEX function CPXgetobjsen. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
getObjDirCPLEX(env, lp)

Arguments
- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details
Interface to the C function getObjDir which calls the CPLEX function CPXgetobjsen.

Value
Zero if the problem object or environment does not exist, otherwise nonzero.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
cplexConstants
getObjNameCPLEX

Access the Name of the Objective Row of a CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetobjname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getObjNameCPLEX(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getObjName which calls the CPLEX function CPXgetobjname.

Value

Name of the objective row if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Objective Offset Between the Original Problem and the Presolved Problem.

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetobjoffset. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getObjOffsetCplex(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getObjOffset which calls the CPLEX function CPXgetobjoffset.

Value

Objective offset value if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXgetobjval. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

getObjValCPLEX(env, lp)

**Arguments**

- **env**: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

**Details**

Interface to the C function getObjVal which calls the CPLEX function CPXgetobjval.

**Value**

Objective value if successful, otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getOrderCPLEX

Access MIP Priority Order Information

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetorder. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getAddressCPLEX(env, lp)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp   An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getorder which calls the CPLEX function CPXgetorder.

Value

If successful a list is returned:
indices  indices of the variables in the order
priority  priority values
direction preferred branching directions

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
getParmNameCPLEX

Obtain the Name of a CPLEX Parameter, Given the Symbolic Constant

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetparamname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getParmNameCPLEX(env, wparm)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

wparm Constant or reference number of the desired parameter.

Details

Interface to the C function getParmName which calls the CPLEX function CPXgetparamname.

Value

A single character value.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
getParmNumCplex

Obtain the Reference Number of a CPLEX Parameter

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetparamnum. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getParmNumCplex(env, nparm)

Arguments

env                      An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
parm                      A single character value containing the name of the parameter.

Details

Interface to the C function getParmNum which calls the CPLEX function CPXgetparamnum.

Value

A single integer value.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
getParmTypeCPLEX  

Obtain the Type of a CPLEX Parameter

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetparamtype. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getParmTypeCPLEX(env, parm)

Arguments

env  
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

parm  
Constant or reference number of the desired parameter.

Details

Interface to the C function getParamType which calls the CPLEX function CPXgetparamtype.

Value

A single integer value if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
getParmValCPLEX

Values and Names of Parameters Having Non-Default Values

Description
The function getParmValCPLEX retrieves the names and actual values of all IBM ILOG CPLEX parameters, which do not have their default values.

Usage
getParmValCPLEX(env)

Arguments
env
An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

Value
Either a list containing all non-default parameters and their values or NULL.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
cplexConstants and getChgParmCPLEX

getPhase1CntCPLEX

Access Number of Phase I Iterations

Description
Low level interface function to the IBM ILOG CPLEX function CPXgetphase1cnt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
getPhase1CntCPLEX(env, lp)
getPiCplex

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getPhase1Cnt which calls the CPLEX function CPXgetphase1cnt.

Value

Zero if no solution exists, otherwise Phase I iteration count.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


getiCplex Access Dual Values for a Range of Constraints

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetpi. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getiCplex(env, lp, begin, end)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

begin An integer specifying the beginning of the range of dual values to be returned.

dend An integer specifying the end of the range of dual values to be returned.
Details

Interface to the C function getPi which calls the CPLEX function CPXgetpi.

Value

Values of the dual variables if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


getPreStatCPLEX

Access Presolve Status Information for Columns and Rows

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetprestat. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getPreStatCPLEX(env, lp)

Arguments

<table>
<thead>
<tr>
<th>env</th>
<th>An object of class &quot;cplexPtr&quot; as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
</tbody>
</table>

Details

Interface to the C function getPreStat which calls the CPLEX function CPXgetprestat.
Value

If successful a list is returned:

- `prestat` status of the presolved problem
- `pcstat` presolve status values of the columns
- `prstat` presolve status values of the rows
- `ocstat` presolve status values of the columns of the presolved problem
- `orstat` presolve status values of the rows of the presolved problem

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants

getProbNameCplex

Access Problem Name

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetprobname. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getProbNameCplex(env, lp)

Arguments

- `env` An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getProbName which calls the CPLEX function CPXgetprobname.
getProbTypeCplex

Value
Name of the problem if successful, otherwise an instance of class "cplexError".

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getProbTypeCplex
Access Problem Type

Description
Low level interface function to the IBM ILOG CPLEX function CPXgetprobtype. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
getProbTypeCplex(env, lp)

Arguments
env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details
Interface to the C function getProbType which calls the CPLEX function CPXgetprobtype.

Value
A single integer value specifying the problem type.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
getProbVarCplex

References


See Also

chgProbTypeCplex, cplexConstants section “Problem Types”.

getProbVarCplex

Access the Solution Values for a Range of Problem Variables

Description

Low level interface function to the IBM ILOG CPLEX function cpxgetx. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getProbVarCplex(env, lp, begin, end)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>env</td>
<td>An object of class &quot;cplexPtr&quot; as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.</td>
</tr>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
<tr>
<td>begin</td>
<td>An integer specifying the beginning of the range of variable values to be returned.</td>
</tr>
<tr>
<td>end</td>
<td>An integer specifying the end of the range of variable values to be returned.</td>
</tr>
</tbody>
</table>

Details

Interface to the C function getProbVar which calls the CPLEX function CPXgetx.

Value

Values of the primal variables if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getQConstrCPLEX

Access a Specified Quadratic Constraint on the Variables of a CPLEX Problem Object.

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetqconstr. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getQConstrCPLEX(env, lp, which)

Arguments

<table>
<thead>
<tr>
<th>arg</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>env</td>
<td>An object of class &quot;cplexPtr&quot; as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.</td>
</tr>
<tr>
<td>lp</td>
<td>An object of class &quot;cplexPtr&quot; as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.</td>
</tr>
<tr>
<td>which</td>
<td>An integer specifying which quadratic constraint to return.</td>
</tr>
</tbody>
</table>

Details

Interface to the C function getQConstr which calls the CPLEX function CPXgetqconstr.

Value

If successful a list is returned:

<table>
<thead>
<tr>
<th>arg</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rhs</td>
<td>Righthand-side value of the quadratic constraint. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>sense</td>
<td>Character specifying the sense of the constraint. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>linind</td>
<td>Variable indices of the entries of linval. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>linval</td>
<td>Linear coefficients of the specified constraint. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>quadrow</td>
<td>Variable indices of the entries of quadval. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>quadcol</td>
<td>Variable indices of the entries of quadval. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
<tr>
<td>quadval</td>
<td>Quadratic coefficients of the specified constraint. Consult the IBM ILOG CPLEX documentation for more detailed information.</td>
</tr>
</tbody>
</table>

otherwise an instance of class "cplexError".
getQPcoefCplex

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


---

getQPcoefCplex \hspace{1cm} Access the Quadratic Coefficient in the Matrix \( Q \)

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXgetqcoef. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

getQPcoefCplex(env, lp, i, j)

**Arguments**

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **i**: The row number in \( Q \).
- **j**: The row column in \( Q \).

**Details**

Interface to the C function getQPcoef which calls the CPLEX function CPXgetqcoef.

**Value**

Specified quadratic coefficient in the matrix \( Q \) if successful, otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getQuadCplex

Access a Range of Columns of the Matrix Q of a Model With a Quadratic Objective Function

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetquad. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getQuadCplex(env, lp, begin, end)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
begin An integer specifying the beginning of the range of columns to be returned.
end An integer specifying the end of the range of columns to be returned.

Details

Interface to the C function getQuad which calls the CPLEX function CPXgetquad.

Value

If successful a list is returned:

qmatbeg Array that specifies the nonzero elements of the columns. Consult the IBM ILOG CPLEX documentation for more detailed information.
qmatind Array that specifies the nonzero elements of the columns. Consult the IBM ILOG CPLEX documentation for more detailed information.
qmatval Array that specifies the nonzero elements of the columns. Consult the IBM ILOG CPLEX documentation for more detailed information.

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getRedLpCplex

Get a Pointer for the Presolved Problem

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetredlp. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getRedLpCplex(env, lp, ptrtype = "cplex_prob")

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- ptrtype: A name for the pointer object.

Details

Interface to the C function getRedLp which calls the CPLEX function CPXgetredlp.

Value

Pointer for the presolved problem if successful (an instance of class "cplexPtr"), otherwise an instance of class "cplexError" or NULL.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getRhsCplex

Access Righthand Side Coefficients for a Range of Constraints

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetrhs. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getRhsCplex(env, lp, begin, end)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

begin An integer specifying the beginning of the range of righthand side terms to be returned.

end An integer specifying the end of the range of righthand side terms to be returned.

Details

Interface to the C function getRhs which calls the CPLEX function CPXgetrhs.

Value

Specified righthand side coefficients if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getRngValCplex

Accesses Right-hand Side Coefficients Range Coefficients

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetrngval. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getRngValCplex(env, lp, begin, end)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- begin: An integer specifying the beginning of the set of rows for which RHS range coefficients are returned.
- end: An integer specifying the end of the set of rows for which RHS range coefficients are returned.

Details

Interface to the C function getRngVal which calls the CPLEX function CPXgetrngval.

Value

Specified RHS range coefficients if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**getRowIndexCPLEX**  
*Search for the Index Number of a Specified Row*

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXgetrowindex`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```
getRowIndexCPLEX(env, lp, rname)
```

**Arguments**

- `env` An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `rname` A row name to search for.

**Details**

Interface to the C function `getRowIndex` which calls the CPLEX function `CPXgetrowindex`.

**Value**

Specified row index if successful, otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getRowInfeasCPLEX

Compute Infeasibility of a Given Solution for a Range of Linear Constraints

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetrowinfeas. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getRowInfeasCPLEX(env, lp, begin, end, sol = NULL)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
begin An integer specifying the beginning of the range of linear constraints whose infeasibility is to be returned.
end An integer specifying the beginning of the range of linear constraints whose infeasibility is to be returned.
sol The solution whose infeasibility is to be computed.

Details

Interface to the C function getRowInfeas which calls the CPLEX function CPXgetrowinfeas.

Value

Infeasibility values if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**getRowNameCplex**  
*Access a Range of Row Names*

---

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXgetRowname`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```c
getRowNameCplex(env, lp, begin, end)
```

**Arguments**

- `env`: An object of class `cplexPtr` as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp`: An object of class `cplexPtr` as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `begin`: An integer specifying the beginning of the range of row names to be returned.
- `end`: An integer specifying the end of the range of row names to be returned.

**Details**

Interface to the C function `getRowName` which calls the CPLEX function `CPXgetRowname`.

**Value**

Specified row names if successful, otherwise an instance of class `cplexError`.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

getRowsCPLEX

Accesses a Range of Rows of the Constraint Matrix

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetrows. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getRowsCPLEX(env, lp, begin, end)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- begin: An integer specifying the beginning of the range of rows to be returned.
- end: An integer specifying the end of the range of rows to be returned.

Details

Interface to the C function getRows which calls the CPLEX function CPXgetrows.

Value

If successful a list is returned:

- matbeg: Array that specifies the nonzero elements of the rows. Consult the IBM ILOG CPLEX documentation for more detailed information.
- matind: Array that specifies the nonzero elements of the rows. Consult the IBM ILOG CPLEX documentation for more detailed information.
- matval: Array that specifies the nonzero elements of the rows. Consult the IBM ILOG CPLEX documentation for more detailed information.

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getSenseCplex

Access the Sense for a Range of Constraints in a CPLEX Problem Object.

Description

Low level interface function to the IBM ILOG CPLEX function cpxgetsense. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getSenseCplex(env, lp, begin, end)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

begin An integer specifying the beginning of the range of constraint senses to be returned.

der An integer specifying the end of the range of constraint senses to be returned.

Details

Interface to the C function getSense which calls the CPLEX function cpxgetsense.

Value

Specified constraint senses if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getSiftItCntCPLEX \hspace{1em} \textit{Access Total Number of Sifting Iterations}

\section*{Description}

Low level interface function to the IBM ILOG CPLEX function \texttt{CPXgetsiftitcnt}. Consult the IBM ILOG CPLEX documentation for more detailed information.

\section*{Usage}

\begin{verbatim}
getsiftitcntcplex(env, lp)
\end{verbatim}

\section*{Arguments}

\begin{itemize}
  \item \texttt{env} \hspace{1em} An object of class "\texttt{cplexPtr}" as returned by \texttt{openEnvCPLEX}. This is basically a pointer to an IBM ILOG CPLEX environment.
  \item \texttt{lp} \hspace{1em} An object of class "\texttt{cplexPtr}" as returned by \texttt{initProbCPLEX}. This is basically a pointer to an IBM ILOG CPLEX problem object.
\end{itemize}

\section*{Details}

Interface to the C function \texttt{getSiftItCnt} which calls the CPLEX function \texttt{CPXgetsiftitcnt}.

\section*{Value}

Zero if no solution exists, otherwise nonzero the total iteration count.

\section*{Author(s)}

Gabriel Gelius-Dietrich \texttt{<geliudie@uni-duesseldorf.de>}

Maintainer: Claus Jonathan Fritzemeier \texttt{<clausjonathan.fritzemeier@uni-duesseldorf.de>}

\section*{References}

getSiftPase1CntCPLEX

Access Number of Phase I Sifting Iterations

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetsiftphase1cnt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getSiftPase1CntCPLEX(env, lp)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getSiftPase1Cnt which calls the CPLEX function CPXgetsiftphase1cnt.

Value

Zero if no solution exists, otherwise nonzero the Phase I iteration count.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getSlackCPLEX \hspace{1em} \textit{Accesses Slack Values for a Range of Linear Constraints}

\section*{Description}
Low level interface function to the IBM ILOG CPLEX function \texttt{CXPgetslack}. Consult the IBM ILOG CPLEX documentation for more detailed information.

\section*{Usage}
\begin{verbatim}
getSlackCPLEX(env, lp, begin, end)
\end{verbatim}

\section*{Arguments}
\begin{itemize}
\item \texttt{env} \hspace{1em} An object of class "\texttt{cplexPtr}" as returned by \texttt{openEnvCPLEX}. This is basically a pointer to an IBM ILOG CPLEX environment.
\item \texttt{lp} \hspace{1em} An object of class "\texttt{cplexPtr}" as returned by \texttt{initProbCPLEX}. This is basically a pointer to an IBM ILOG CPLEX problem object.
\item \texttt{begin} \hspace{1em} An integer specifying the beginning of the range of slack values to be returned.
\item \texttt{end} \hspace{1em} An integer specifying the end of the range of slack values to be returned.
\end{itemize}

\section*{Details}
Interface to the C function \texttt{getSlack} which calls the CPLEX function \texttt{CXPgetslack}.

\section*{Value}
Specified slack or surplus variables if successful, otherwise an instance of class "\texttt{cplexError}".

\section*{Author(s)}
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

\section*{References}
getStatCPLEX

Access the Solution Status of the Problem

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetstat. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getStatCPLEX(env, lp)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getStat which calls the CPLEX function CPXgetstat.

Value

A single integer value giving the solution status.

Author(s)

- Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
  - Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants section “Values returned for stat by solution”.
getStatStrCPLEX  

Return an Status Message String Corresponding to an Status Code

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetstatstring. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getStatStrCPLEX(env, stat)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
stat The status code to be translated.

Details

Interface to the C function getStatStr which calls the CPLEX function CPXgetstatstring.

Value

A single character value containing the status message string.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

getErrorStrCPLEX
getStrParmCplex

Obtain the Current Value of a CPLEX String Parameter

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetstrparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getStrParmCplex(env, parm)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
parm Constant or reference number of the desired parameter.

Details

Interface to the C function getStrParm which calls the CPLEX function CPXgetstrparam.

Value

A single character value.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
getSubMethodCplex

Accesses Solution Method of the Last Subproblem Optimization

Description

Low level interface function to the IBM ILOG CPLEX function `CPXgetsubmethod`. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```r
getSubMethodCplex(env, lp)
```

Arguments

- `env` An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function `getSubMethod` which calls the CPLEX function `CPXgetsubmethod`.

Value

Integer value specifying the solution method.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
getSubStatCplex

Access Solution Status of the Last Subproblem Optimization

Description

Low level interface function to the IBM ILOG CPLEX function CPXgetsubstat. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getSubStatCplex(env, lp)

Arguments

env
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp
An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function getSubStat which calls the CPLEX function CPXgetsubstat.

Value

Zero if no solution exists, nonzero otherwise.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getTimeCplex \hspace{1em} Get a Time Stamp

Description

Low level interface function to the IBM ILOG CPLEX function CPXfclose. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```r
getimeCplex(env)
```

Arguments

- `env` An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.

Details

Interface to the C function `getTime` which calls the CPLEX function CPXgettime.

Value

If successful a single numeric value, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

`fileputCplex`, `openFileCplex`
getUppBndsIdsCPLEX  Retrieve Upper Bounds on Variables

Description

The function retrieves the upper bounds on specified variables.

Usage

getUppBndsIdsCPLEX(env, lp, ind)

Arguments

env          An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp           An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
ind          Column indices of variables (remember: first index is 0).

Value

A numeric vector containing the upper bounds on the specified variables. If not successfull an instance of class "cplexError" is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

getUpperBndsCPLEX
getUpperBndsCPLEX

Access a Range of Upper Bounds on Variables

Description

Low level interface function to the IBM ILOG CPLEX function `cpxgetub`. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```
getUpperBndsCPLEX(env, lp, begin, end)
```

Arguments

- **env**: An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **begin**: Beginning of the range of upper bounds to be returned.
- **end**: End of the range of upper bounds to be returned.

Details

Interface to the C function `getUpperBnds` which calls the CPLEX function `CPXgetub`.

Value

A numeric vector containing the lower bounds on the specified variables. If not successfull an instance of class "cplexError" is returned.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

getVersionCPLEX  

Get Version Number of the CPLEX Library.

Description

Low level interface function to the IBM ILOG CPLEX function CPXversion. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

getVersionCPLEX(env)

Arguments

| env       | An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment. |

Details

Interface to the C function getVersion which calls the CPLEX function getVerboseCPLEX.

Value

Single character string specifying the version of the cplex library or NULL if the environment does not exist.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Solve the Specified Problem by the CPLEX Barrier Optimizer

Description

Low level interface function to the IBM ILOG CPLEX function CPXhybbaropt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

   hybbaroptCPlEX(env, lp, method)

Arguments

   env  An object of class "cplexPtr" as returned by openEnvCPlEX. This is basically a pointer to an IBM ILOG CPLEX environment.

   lp   An object of class "cplexPtr" as returned by initProbCPlEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

   method A single integer value giving the crossover method to be implemented.

Details

Interface to the C function hybbaropt which calls the CPLEX function CPXhybbaropt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

solnInfoCPlEX, getStatCPlEX, solutionCPlEX, cplexConstants section “LP/QP solution algorithms”.

Description

Low level interface function to the IBM ILOG CPLEX function CPXhybnetopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

\[\text{hybnetoptCPLEX}(\text{env}, \text{lp}, \text{method})\]

Arguments

- **env**: An object of class "cplexPtr" as returned by \text{openEnvCPLEX}. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by \text{initProbCPLEX}. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **method**: A single integer value giving the type of simplex method to follow the network optimization.

Details

Interface to the C function hybnetopt which calls the CPLEX function CPXhybnetopt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

The IBM ILOG CPLEX home page at https://www.ibm.com/developerworks/university/academicinitiative.\

See Also

\text{solnInfoCLEX, getStatCPLEX, solutionCPLEX, cplexConstants} section “LP/QP solution algorithms”.

initProbCplex

Create a CPLEX Problem Object in the CPLEX Environment

Description

Low level interface function to the IBM ILOG CPLEX function CPXcreateprob. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

initProbCplex(env, pname = "CPLEX_PROB", ptrtype = "cplex_prob")

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **pname**: A single character string containing the name of the problem object.
- **ptrtype**: A name for the pointer object.

Details

Interface to the C function CPXcreate which calls the CPLEX function CPXcreateprob.

Value

If successful, a pointer to the CPLEX problem object is returned (an instance of class "cplexPtr"), otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

delProbCplex
## lpoptCPLEX
*Find a Solution to a Problem Using One of the CPLEX Linear Optimizers*

### Description
Low level interface function to the IBM ILOG CPLEX function CPXlpopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

### Usage
```
lpoptCPLEX(env, lp)
```

### Arguments
- `env` An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.

### Details
Interface to the C function `lpopt` which calls the CPLEX function CPXlpopt.

### Value
Zero if successful, otherwise nonzero.

### Author(s)
- Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
- Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

### References

### See Also
- `solnInfoCPLEX`, `getStatCPLEX`, `solutionCPLEX`
Find a Solution to a Mixed Integer Program

Description

Low level interface function to the IBM ILOG CPLEX function CPXmipopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

mipoptCplex(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function mipopt which calls the CPLEX function CPXmipopt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

solnInfoCplex, getStatCplex, solutionCplex
newColsCPLEX

Add Empty Columns to a Specified CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXnewcols. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

newColsCPLEX(env, lp, ncols,
  obj = NULL, lb = NULL, ub = NULL,
  xctype = NULL, cnames = NULL)

Arguments

env     An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp      An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
ncols   Number of variables to add.
obj     Objective function coefficients.
lb      Lower bounds on the new variables.
ub      Upper bounds on the new variables.
xctype  Type of the new variables.
cnames  Names of the new variables.

Details

Interface to the C function newCols which calls the CPLEX function CPXnewcols.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

newRowsCPLEX

Add Empty Constraints to a Specified CPLEX Problem Object

Description

Low level interface function to the IBM ILOG CPLEX function CPXnewrows. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```r
newRowsCPLEX(env, lp, nrows, rhs = NULL, sense = NULL, rngval = NULL, rnames = NULL)
```

Arguments

- `env`: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp`: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `nrows`: Number of new rows.
- `rhs`: Right hand side term for each new constraint.
- `sense`: Sense of each new constraint (see IBM ILOG CPLEX documentation for possible values).
- `rngval`: Range values for each new constraint.
- `rnames`: Names for the new rows.

Details

Interface to the C function newRows which calls the CPLEX function CPXnewrows.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

**objcCPLEX**

*Access Upper and Lower Sensitivity Ranges for Objective Function Coefficients*

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXobjsa`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
objcCPLEX(env, lp, begin, end)
```

**Arguments**

- `env` An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCPLEX`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `begin` Beginning of the range of ranges to be returned.
- `end` End of the range of ranges to be returned.

**Details**

Interface to the C function `objsa` which calls the CPLEX function `CPXobjsa`.

**Value**

If successful a list is returned:

- `lower` lower range values
- `upper` upper range values

otherwise an instance of class "cplexError".

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

openEnvCplex

Initialize a CPLEX Environment

Description

Low level interface function to the IBM ILOG CPLEX function CPXopenCplex. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

openEnvCplex(ptrtype = "cplex.env")

Arguments

ptrtype A name for the pointer object.

Details

Interface to the C function openEnv which calls the CPLEX function CPXopenCplex.

Value

If successful, a pointer to the CPLEX environment is returnd (an instance of class "cplexPtr"), otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

closeEnvCplex
openFileCplex  

**Open a File**

**Description**

Low level interface function to the IBM ILOG CPLEX function cpxfopen. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
openFileCplex(fname, ftype = "w", ptrtype = "cplex_file")
```

**Arguments**

- `fname`  
  Character string giving the file name to be opened.

- `ftype`  
  Character string according to the syntax of the standard C function `fopen`.

- `ptrtype`  
  A name for the pointer object.

**Details**

Interface to the C function openFile which calls the CPLEX function cpxfopen.

**Value**

A pointer to the log file (an instance of class "cplexPtr") or NULL.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

closeFileCplex, fileputcplex
openProbCPLEX  

Create new CPLEX Environment And New CPLEX Problem Object

Description
The function openProbCPLEX creates a new CPLEX environment and a new CPLEX problem object.

Usage
```c
openProbCPLEX(pname = "CPLEX_PROB",
ptrtypeENV = "cplex_env",
ptrtypePROB = "cplex_prob")
```

Arguments
- **pname**: A single character string containing the name of the problem object.
- **ptrtypeENV**: A name for the IBM ILOG CPLEX environment pointer object.
- **ptrtypePROB**: A name for the IBM ILOG CPLEX problem pointer object.

Details
Interface to the C functions openenv and initprob calling CPLEX functions CPXopenCplex and CPXcreateprob.

Value
- **env**: A pointer to the CPLEX environment as returned by `openEnvCPLEX`.
- **lp**: A pointer to the CPLEX problem object as returned by `initProbCPLEX`.

If `openEnvCPLEX()` fails, env will be of class "cplexError" and lp will be NULL. Each list element is an object of class "cplexPtr".

Author(s)
- Gabriel Gelius-Dietrich <heliudie@uni-duesseldorf.de>
- Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
- closeProbCPLEX, openEnvCPLEX, initProbCPLEX
ordWriteCplex  Write Priority Order to ORD File

Description

Low level interface function to the IBM ILOG CPLEX function CPXordwrite. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

ordWriteCplex(env, lp, fname)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

fname Filename.

Details

Interface to the C function ordWrite which calls the CPLEX function CPXordwrite.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
preslvWriteCplex  Write a Presolved Version of the Problem to File

Description

Low level interface function to the IBM ILOG CPLEX function CPXpreslvwrite. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

preslvWriteCplex(env, lp, fname)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp   An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
fname Single character value giving the file name to write to.

Details

Interface to the C function preslvWrite which calls the CPLEX function CPXpreslvWrite.

Value

If successful a dingle numeric value containing the objective value difference between the original problem and the presolved problem, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

readCopyProbCplex
presolveCplex

Perform Presolve

Description

Low level interface function to the IBM ILOG CPLEX function CPXpresolve. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

presolveCplex(env, lp, method)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- method: A single integer value specifying the optimization algorithm to be used to solve the problem after the presolve is completed.

Details

Interface to the C function presolve which calls the CPLEX function CPXpresolve.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants section “LP/QP solution algorithms”.
primoptCPLEX  Find a Solution to a Problem Using the Primal Simplex Method

Description

Low level interface function to the IBM ILOG CPLEX function CPXprimopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

primoptCPLEX(env, lp)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp  An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function primopt which calls the CPLEX function CPXprimopt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

solnInfoCPLEX, getStatCPLEX, solutionCPLEX
printTerminateCPLEX  Print Termination Signal

Description

The function chgTerminateCPLEX prints termination signal.

Usage

printTerminateCPLEX(env)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

Value

NULL

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

setTerminateCPLEX, delTerminateCPLEX, chgTerminateCPLEX

qpoptCPLEX  Find a Solution to a Continuous Quadratic Program

Description

Low level interface function to the IBM ILOG CPLEX function CPXqpopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

qpoptCPLEX(env, lp)
Arguments

env  An object of class "cplexPtr" as returned by openEnvCPELEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp   An object of class "cplexPtr" as returned by initProbCPELEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function qpopt which calls the CPLEX function CPXqpopt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

solnInfoCPELEX, getStatCPELEX, solutionCPELEX

Description

Low level interface function to the IBM ILOG CPLEX function CPXreadcopybase. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

readCopyBaseCPELEX(env, lp, fname)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPELEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp   An object of class "cplexPtr" as returned by initProbCPELEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

fname Single character value giving the filename to read from.
**readCopyMIPstartsCplex**

**Details**

Interface to the C function `readCopyBase` which calls the CPLEX function `CPXreadcopybase`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


---

**readCopyMIPstartsCplex**

*Read a File in the Format MST*

**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXreadcopymipstarts`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```
readCopyMIPstartsCplex(env, lp, fname)
```

**Arguments**

- **env**
  An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.

- **lp**
  An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.

- **fname**
  Name of the file to read from.

**Details**

Interface to the C function `readCopyMIPstarts` which calls the CPLEX function `CPXreadcopymipstarts`.

**Value**

Zero if successful, otherwise nonzero.
readCopyOrderCplex

Read ORD File

Description
Low level interface function to the IBM ILOG CPLEX function CPXreadcopyorder. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
readCopyOrderCplex(env, lp, fname)

Arguments
env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
fname Single character value giving the filename to read from.

Details
Interface to the C function readCopyOrder which calls the CPLEX function CPXreadcopyorder.

Value
Zero if successful, otherwise nonzero.

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References
**Description**

Low level interface function to the IBM ILOG CPLEX function `CPXreadcopyparam`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
readCopyParmCplex(env, fname)
```

**Arguments**

- `env` An object of class "`cplexPtr`" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `fname` Filename.

**Details**

Interface to the C function `readCopyParm` which calls the CPLEX function `CPXreadcopyparam`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

cplexConstants
readCopyProbCplex

**Description**

Low level interface function to the IBM ILOG CPLEX function `cpxreadcopyprob`. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```c
readCopyProbCplex(env, lp, fname, ftype = NULL)
```

**Arguments**

- **env**: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **fname**: Single character value giving the filename to read from.
- **ftype**: Single character value giving the type of the file to read from.

**Details**

Interface to the C function `readCopyProb` which calls the CPLEX function `CPXreadcopyprob`.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

Description

Low level interface function to the IBM ILOG CPLEX function CPXreadcopySol. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

readCopySolCPLEX(env, lp, fname)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

fname Single character value giving the filename to read from.

Details

Interface to the C function readCopySol which calls the CPLEX function CPXreadcopySol.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

refineConflictCPLEX  Identify a Minimal Conflict for the Infeasibility of the Linear Constraints and the Variable Bounds

Description

Low level interface function to the IBM ILOG CPLEX function CPXrefineconflict. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

refineConflictCPLEX(env, lp)

Arguments

env        An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
lp        An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function refineConflict which calls the CPLEX function CPXrefineconflict.

Value

If successful a list is returned:

confnumrows number of linear constraints in the conflict
confnumcols number of variable bounds in the conflict

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

getConflictCPLEX
refineConflictExtCplex

Identify a Minimal Conflict

Description

Low level interface function to the IBM ILOG CPLEX function CPXrefineconflictext. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

```c
refineConflictExtCplex(env, lp, grpcnt, concnt,
                        grppref, grpbeg, grpind, grptype)
```

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **lp**: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- **grpcnt**: The number of constraint groups to be considered.
- **concnt**: Length of arrays grpind and grptype.
- **grppref**: Preferences for the groups.
- **grpbeg**: The constraint indices. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **grpind**: The constraint indices. Consult the IBM ILOG CPLEX documentation for more detailed information.
- **grptype**: The constraint indices. Consult the IBM ILOG CPLEX documentation for more detailed information.

Details

Interface to the C function refineConflictExt which calls the CPLEX function CPXrefineconflictext.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Refine a Conflict in Order to Determine Why a Given MIP Start is Not Feasible

Description

Low level interface function to the IBM ILOG CPLEX function CPXrefinemipstartconflict. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

refinemipstartconflictcplexHenvL lpL mipstartindexI

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- mipstartindex: The index of the MIP start.

Details

Interface to the C function refineMIPstartConflict which calls the CPLEX function CPXrefinemipstartconflict.

Value

If successful a list is returned:

- confnumrows: number of linear constraints in the conflict
- confnumcols: number of variable bounds in the conflict

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

getConflictCplex
refineMIPstartConflictExtCPLEX

Identify a Minimal Conflict

Description

Low level interface function to the IBM ILOG CPLEX function CPXrefinemipstartconflictext. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

refineMIPstartConflictExtCPLEX(env, lp, mipstartindex, grpcnt, concnt, grppref, grpbeg, grpind, grptype)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

mipstartindex The index of the MIP start.

grpcnt The number of constraint groups to be considered.

concnt Length of arrays grpind and grptype.

grppref Preferences for the groups.

grpbeg The constraint indices. Consult the IBM ILOG CPLEX documentation for more detailed information.

grpind The constraint indices. Consult the IBM ILOG CPLEX documentation for more detailed information.

grptype The constraint indices. Consult the IBM ILOG CPLEX documentation for more detailed information.

Details

Interface to the C function refineMIPstartConflictExt which calls the CPLEX function CPXrefinemipstartconflictext.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
return_codeCPLEX

Translates a IBM ILOG CPLEX Return Code into a Human Readable String

Description

Translates a IBM ILOG CPLEX return code into a human readable string.

Usage

`return_codeCPLEX(code)`

Arguments

- `code` Return (error) code from IBM ILOG CPLEX.

Value

An error message string corresponding to an return (error) code.

Author(s)

- Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
- Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

- cplexConstants
Access Upper and Lower Sensitivity Ranges for Righthand Side Values of a Range of Constraints

Description
Low level interface function to the IBM ILOG CPLEX function CPXrhssa. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
```
rhsSaCplex(env, lp, begin, end)
```

Arguments
- `env`: An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp`: An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `begin`: Beginning of the range of ranges to be returned.
- `end`: End of the range of ranges to be returned.

Details
Interface to the C function `rhsSa` which calls the CPLEX function CPXrhssa.

Value
If successful a list is returned:

- `lower`: righthand side lower range values
- `upper`: righthand side upper range values

otherwise an instance of class "cplexError".

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References
setDblParmCplex

Set the Value of a CPLEX Parameter of Type Double

Description

Low level interface function to the IBM ILOG CPLEX function CPXsetdblparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

setDblParmCplex(env, parm, value)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- parm: Constant or reference number of the desired parameter.
- value: The new value of the parameter.

Details

Interface to the C function setDblParm which calls the CPLEX function CPXsetdblparam.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
**setDefaultParmCplex**

_Reset All CPLEX Parameters And Settings to Default Values_

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXsetdefaults. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
setDefaultParmCplex(env)
```

**Arguments**

- `env` An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.

**Details**

Interface to the C function `setDefaultParm` which calls the CPLEX function CPXsetdefaults.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

cplexConstants
setIntParmCplex

Set the Value of a CPLEX Parameter of Type CPXINT

Description

Low level interface function to the IBM ILOG CPLEX function CPXsetintparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

setIntParmCplex(env, parm, value)

Arguments

- **env**: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- **parm**: Constant or reference number of the desired parameter.
- **value**: The new value of the parameter (integer value).

Details

Interface to the C function setIntParm which calls the CPLEX function CPXsetintparam.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
**setLogFileCPLEX**

Modifies the log file to which Messages are Written

**Description**

Low level interface function to the IBM ILOG CPLEX function CPXsetlogfile. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```c
setLogFileCPLEX(env, cpfile = NULL)
```

**Arguments**

- `env` An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `cpfile` A pointer to a file as returned by `openFileCPLEX`.

**Details**

Interface to the C function `getLogFile` which calls the CPLEX function CPXgetlogfile.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

- `getLogFileCPLEX`
setLongParmCPLEX

Set the Value of a Parameter of Type CPXLONG

Description

Low level interface function to the IBM ILOG CPLEX function CPXsetlongparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

setLongParmCPLEX(env, parm, value)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- parm: Constant or reference number of the desired parameter.
- value: New value for the parameter.

Details

Interface to the C function setLongParm which calls the CPLEX function CPXsetlongparam.

Value

Zero if successful, otherwise nonzero.

Note

In order to transfer a 64 bit integer value to CPXsetlongparam, datatype numeric is used. Parameter value is a numeric value.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

setIntParmCPLEX, cplexConstants
setObjDirCplex

Change the Sense of the Optimization for a Problem

Description
Low level interface function to the IBM ILOG CPLEX function CPLEXchgbjsen. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage
```
setObjDirCplex(env, lp, lpdire)
```

Arguments
- `env`: An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp`: An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `lpdir`: A single integer value specifying the sense of the problem.

Details
Interface to the C function setObjDir which calls the CPLEX function CPXchgbjsen.

Value
null

Author(s)
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also
cplexConstants section “Generic constants”.
setStrParmCPLEX  

*Set the Value of a CPLEX String Parameter*

**Description**

High level interface function to the IBM ILOG CPLEX function CPXsetstrparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**

```r
setStrParmCPLEX(env, parm, value)
```

**Arguments**

- `env`  
  An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.

- `parm`  
  Constant or reference number of the desired parameter.

- `value`  
  The new value of the parameter (character value).

**Details**

Interface to the C function `setStrParm` which calls the CPLEX function CPXsetstrparam.

**Value**

Zero if successful, otherwise nonzero.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**


**See Also**

- `cplexConstants`
**setTerminateCPLEX**  
*Release Termination Signal*

**Description**
Low level interface function to the IBM ILOG CPLEX function `CPXsetterminate` with argument `terminate_p` set to NULL. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**
```
setTerminateCPLEX(env, ptrtype = "cplex_term")
```

**Arguments**
- `env`: An object of class "cplexPtr" as returned by `openEnvCPLEX`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `ptrtype`: A name for the pointer object.

**Details**
Interface to the C function `delTerminate` which calls the CPLEX function `CPXsetterminate` with argument `terminate_p` set to NULL.

**Value**
If successful, a pointer to a termination signal is returned, otherwise an instance of class "cplexError".

**Author(s)**
- Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
- Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

**See Also**
- `delTerminateCPLEX`, `printTerminateCPLEX`, `chgTerminateCPLEX`
siftoptCPLEX  Solve a Reduced Model

Description

Low level interface function to the IBM ILOG CPLEX function CPXsiftopt. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

siftoptCPLEX(env, lp)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp   An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function siftopt which calls the CPLEX function CPXsiftopt.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Access Solution Information

Description

Low level interface function to the IBM ILOG CPLEX function CPXsolninfo. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

solnInfoCPLEX(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function solnInfo which calls the CPLEX function CPXsolninfo.

Value

If successful a list is returned:

method Integer value specifying the method to produce the current solution.

type Integer value specifying the type of current solution.

primal_feasible Integer value specifying if the current solution is known to be primal feasible.

dual_feasible Integer value specifying if the current solution is known to be dual feasible.

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants, solutionCPLEX
solutionCPLEX

Access Solution Values Produced by Optimization Routines

Description

Low level interface function to the IBM ILOG CPLEX function CPXsolution. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

solutionCPLEX(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function solution which calls the CPLEX function CPXsolution.

Value

If successful a list is returned:

lpstat result of the optimization
objval objective function value
x values of the variables for the problem
pi values of the dual variables
slack values of the slack or surplus variables
dj values of the reduced costs

otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

See Also

cplexConstants, solnInfoCPLEX

---

solWriteCPLEX Write a Solution File

Description

Low level interface function to the IBM ILOG CPLEX function CPXsolwrite. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

solWriteCPLEX(env, lp, fname)

Arguments

env An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

fname Single character value giving the filename to write to.

Details

Interface to the C function solWrite which calls the CPLEX function CPXsolwrite.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

getProbTypeCPLEX, cplexConstants section “Problem Types”.
status_codeCPLEX  Translates an IBM ILOG CPLEX Status Value into a Human Readable String

Description

Translates a IBM ILOG CPLEX status code into a human readable string.

Usage

status_codeCPLEX(env, code)

Arguments

env  An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
code  Status code from IBM ILOG CPLEX as returned by getStatCPLEX.

Value

A character string corresponding to the value of an IBM ILOG CPLEX status code as returned by getStatCPLEX.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants, getStatStrCPLEX
**tightenBndsCplex**  
*Change the Lower or Upper Bounds on a Set of Variables of a Problem*

**Description**
Low level interface function to the IBM ILOG CPLEX function CPXtightenbds. Consult the IBM ILOG CPLEX documentation for more detailed information.

**Usage**
```r
tightenBndsCplex(env, lp, ncols, ind, lu, bd)
```

**Arguments**
- `env` An object of class "cplexPtr" as returned by `openEnvCplex`. This is basically a pointer to an IBM ILOG CPLEX environment.
- `lp` An object of class "cplexPtr" as returned by `initProbCplex`. This is basically a pointer to an IBM ILOG CPLEX problem object.
- `ncols` Number of bounds to be changed.
- `ind` Indices of bounds to be changed.
- `lu` A character vector, specifying whether an entry in `bd` is a upper or a lower bound on variable `ind[j]`.
- `bd` Values of the lower or upper bounds of the variables present in `ind`.

**Details**
Interface to the C function `tightenBnds` which calls the CPLEX function CPXtightenbds.

**Value**
Zero if successful, otherwise nonzero.

**Author(s)**
Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

**References**

**See Also**
chgBndsCplex
tuneParmCPLEX

Tune Parameters of the Environment For Improved Optimizer Performance

Description

Low level interface function to the IBM ILOG CPLEX function CPXtuneparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

tuneParmCPLEX(env, lp,
    nIntP = 0, intP = NULL, intPv = NULL,
    nDb1P = 0, db1P = NULL, db1Pv = NULL,
    nStrP = 0, strP = NULL, strPv = NULL)

Arguments

env       An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.

lp        An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.

nIntP     Number of integer parameters to be fixed during tuning.

intP      Parameter numbers of the integer parameters which remain fixed.

intPv     Values for the parameters listed in intP.

nDb1P     Number of double parameters to be fixed during tuning.

db1P      Parameter numbers of the double parameters which remain fixed.

db1Pv     Values for the parameters listed in db1P.

nStrP     Number of string parameters to be fixed during tuning.

strP      Parameter numbers of the string parameters which remain fixed.

strPv     Values for the parameters listed in strP.

Details

Interface to the C function tuneParam which calls the CPLEX function CPXtuneparam.

Value

Zero if successful, otherwise an instance of class "cplexError".

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>
unscaleProbCplex

References


unscaleProbCplex Remove Any Scaling Applied to the Resident Problem

Description

Low level interface function to the IBM ILOG CPLEX function CPXunscaleprob. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

unscaleProbCplex(env, lp)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

Details

Interface to the C function unscaleProb which calls the CPLEX function CPXunscaleprob.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

writeMIPstartsCplex  

Write a Range of MIP Starts to a File in MST Format

Description

Low level interface function to the IBM ILOG CPLEX function CPXwritemipstarts. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

writeMIPstartsCplex(env, lp, fname, begin, end)

Arguments

env  
An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.

lp  
An object of class "cplexPtr" as returned by initProbCplex. This is basically a pointer to an IBM ILOG CPLEX problem object.

fname  
Filename to write to.

begin  
An integer specifying the beginning of the range of MIP starts to be written.

end  
An integer specifying the end of the range of MIP starts to be written.

Details

Interface to the C function writeMIPstarts which calls the CPLEX function CPXwritemipstarts.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>

Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

writeParmCplex

Write Names and Current Settings of CPLEX Parameters to File

Description

Low level interface function to the IBM ILOG CPLEX function CPXwriteparam. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

writeParmCplex(env, fname)

Arguments

env An object of class "cplexPtr" as returned by openEnvCplex. This is basically a pointer to an IBM ILOG CPLEX environment.
fname Filename.

Details

Interface to the C function writeParm which calls the CPLEX function CPXwriteparam.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References


See Also

cplexConstants
Description

Low level interface function to the IBM ILOG CPLEX function CPXwriteprob. Consult the IBM ILOG CPLEX documentation for more detailed information.

Usage

writeProbCPLEX(env, lp, fname, ftype = NULL)

Arguments

- env: An object of class "cplexPtr" as returned by openEnvCPLEX. This is basically a pointer to an IBM ILOG CPLEX environment.
- lp: An object of class "cplexPtr" as returned by initProbCPLEX. This is basically a pointer to an IBM ILOG CPLEX problem object.
- fname: Single character value giving the file name to write to.
- ftype: Single character value giving the type of the file to write to.

Details

Interface to the C function writeProb which calls the CPLEX function CPXwriteprob.

Value

Zero if successful, otherwise nonzero.

Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>
Maintainer: Claus Jonathan Fritzemeier <clausjonathan.fritzemeier@uni-duesseldorf.de>

References

Index

*Topic optimize
  - addColsCplex, 7
  - addFpDestCplex, 9
  - addIndConstrCplex, 10
  - addMIPstartsCplex, 11
  - addQConstrCplex, 12
  - addRowsCplex, 13
  - baroptCplex, 14
  - baseWriteCplex, 15
  - basicPresolveCplex, 16
  - boundSaCplex, 17
  - checkAddColsCplex, 18
  - checkAddRowsCplex, 19
  - checkChgCoefListCplex, 20
  - checkCopyColTypeCplex, 21
  - checkCopyLpCplex, 22
  - checkCopyLpwNamesCplex, 23
  - checkCopyQPsepCplex, 25
  - checkCopyQuadCplex, 26
  - checkValsCplex, 27
  - chgBndsCplex, 28
  - chgCoefCplex, 29
  - chgCoefListCplex, 30
  - chgColNameCplex, 31
  - chgColsBndsCplex, 32
  - chgColTypeCplex, 33
  - chgMIPstartsCplex, 34
  - chgNameCplex, 35
  - chgObjCplex, 36
  - chgProbNameCplex, 37
  - chgProbTypeCplex, 38
  - chgQPcoefCplex, 39
  - chgRhsCplex, 40
  - chgRngValCplex, 41
  - chgRowNameCplex, 42
  - chgSenseCplex, 43
  - chgTerminateCplex, 44
  - cleanupCoefCplex, 44
  - cloneProbCplex, 45
  - closeEnvCplex, 46
  - closeFileCplex, 47
  - closeProbCplex, 48
  - clpWriteCplex, 49
  - completeIpCplex, 50
  - copyBaseCplex, 51
  - copyColTypeCplex, 52
  - copyLpCplex, 53
  - copyLpwNamesCplex, 54
  - copyObjNameCplex, 55
  - copyOrderCplex, 56
  - copyPartBaseCplex, 57
  - copyQPsepCplex, 58
  - copyQuadCplex, 59
  - copyStartCplex, 60
  - cplexAPI-package, 6
  - cplexConstants, 61
  - cplexError-class, 82
  - cplexPtr-class, 83
  - delColsCplex, 84
  - delFpDestCplex, 85
  - delIndConstrsCplex, 86
  - delMIPstartsCplex, 87
  - delNamesCplex, 88
  - delProbCplex, 89
  - delQConstrsCplex, 90
  - delRowsCplex, 91
  - delSetColsCplex, 92
  - delSetRowsCplex, 93
  - delTerminateCplex, 94
  - disconnectChannelCplex, 95
  - dualoptCplex, 96
  - dualWriteCplex, 97
  - feasOptCplex, 98
  - fileputCplex, 99
  - flushChannelCplex, 100
  - flushStdChannelsCplex, 101
  - freePresolveCplex, 102
  - getBaseCplex, 103
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>getBestObjValCPLEX</td>
<td>104</td>
</tr>
<tr>
<td>getChannelsCPLEX</td>
<td>105</td>
</tr>
<tr>
<td>getChgParmCPLEX</td>
<td>106</td>
</tr>
<tr>
<td>getCoefCPLEX</td>
<td>107</td>
</tr>
<tr>
<td>getColIndexCPLEX</td>
<td>108</td>
</tr>
<tr>
<td>getColInfeasCPLEX</td>
<td>109</td>
</tr>
<tr>
<td>getColNameCPLEX</td>
<td>110</td>
</tr>
<tr>
<td>getColsCPLEX</td>
<td>111</td>
</tr>
<tr>
<td>getColTypeCPLEX</td>
<td>112</td>
</tr>
<tr>
<td>getConflictcplex</td>
<td>113</td>
</tr>
<tr>
<td>getConflictextCPLEX</td>
<td>114</td>
</tr>
<tr>
<td>getCutoffCPLEX</td>
<td>115</td>
</tr>
<tr>
<td>getDbllParmCPLEX</td>
<td>116</td>
</tr>
<tr>
<td>getDbllQualCPLEX</td>
<td>117</td>
</tr>
<tr>
<td>getDbssCntCPLEX</td>
<td>118</td>
</tr>
<tr>
<td>getDjCPLEX</td>
<td>119</td>
</tr>
<tr>
<td>getErrorStrCPLEX</td>
<td>120</td>
</tr>
<tr>
<td>getGradCPLEX</td>
<td>121</td>
</tr>
<tr>
<td>getIndConstrCPLEX</td>
<td>122</td>
</tr>
<tr>
<td>getInfoDbllParmCPLEX</td>
<td>123</td>
</tr>
<tr>
<td>getInfoIntParmCPLEX</td>
<td>124</td>
</tr>
<tr>
<td>getInfoLongParmCPLEX</td>
<td>125</td>
</tr>
<tr>
<td>getInfoStrParmCPLEX</td>
<td>126</td>
</tr>
<tr>
<td>getIntParmCPLEX</td>
<td>127</td>
</tr>
<tr>
<td>getIntQualCPLEX</td>
<td>128</td>
</tr>
<tr>
<td>getItCntCPLEX</td>
<td>129</td>
</tr>
<tr>
<td>getLogFileCPLEX</td>
<td>130</td>
</tr>
<tr>
<td>getLongParmCPLEX</td>
<td>131</td>
</tr>
<tr>
<td>getLowBndsIdsCPLEX</td>
<td>132</td>
</tr>
<tr>
<td>getLowerBndsCPLEX</td>
<td>133</td>
</tr>
<tr>
<td>getMethodCPLEX</td>
<td>134</td>
</tr>
<tr>
<td>getMIPrelGapCPLEX</td>
<td>135</td>
</tr>
<tr>
<td>getMIPstartIndexCPLEX</td>
<td>136</td>
</tr>
<tr>
<td>getMIPstartNameCPLEX</td>
<td>137</td>
</tr>
<tr>
<td>getMIPstartsCPLEX</td>
<td>138</td>
</tr>
<tr>
<td>getNumColsCPLEX</td>
<td>139</td>
</tr>
<tr>
<td>getNumMIPstartsCPLEX</td>
<td>140</td>
</tr>
<tr>
<td>getNumNnzCPLEX</td>
<td>141</td>
</tr>
<tr>
<td>getNumQConstrsCPLEX</td>
<td>142</td>
</tr>
<tr>
<td>getNumQnzCPLEX</td>
<td>143</td>
</tr>
<tr>
<td>getNumQuadCPLEX</td>
<td>144</td>
</tr>
<tr>
<td>getNumRowsCPLEX</td>
<td>145</td>
</tr>
<tr>
<td>getObjCPLEX</td>
<td>146</td>
</tr>
<tr>
<td>getObjDirCPLEX</td>
<td>147</td>
</tr>
<tr>
<td>getObjNameCPLEX</td>
<td>148</td>
</tr>
<tr>
<td>getObjOffsetCPLEX</td>
<td>149</td>
</tr>
<tr>
<td>getObjValCPLEX</td>
<td>150</td>
</tr>
<tr>
<td>getOrderCPLEX</td>
<td>151</td>
</tr>
<tr>
<td>getParmNameCPLEX</td>
<td>152</td>
</tr>
<tr>
<td>getParmNumCPLEX</td>
<td>153</td>
</tr>
<tr>
<td>getParmTypeCPLEX</td>
<td>154</td>
</tr>
<tr>
<td>getParmValCPLEX</td>
<td>155</td>
</tr>
<tr>
<td>getPhase1CntCPLEX</td>
<td>155</td>
</tr>
<tr>
<td>getPiCPLEX</td>
<td>156</td>
</tr>
<tr>
<td>getPreStatCPLEX</td>
<td>157</td>
</tr>
<tr>
<td>getProbNameCPLEX</td>
<td>158</td>
</tr>
<tr>
<td>getProbTypeCPLEX</td>
<td>159</td>
</tr>
<tr>
<td>getProbVarCPLEX</td>
<td>160</td>
</tr>
<tr>
<td>getQConstrCPLEX</td>
<td>161</td>
</tr>
<tr>
<td>getQPcofCPLEX</td>
<td>162</td>
</tr>
<tr>
<td>getQuadCPLEX</td>
<td>163</td>
</tr>
<tr>
<td>getRedLpCPLEX</td>
<td>164</td>
</tr>
<tr>
<td>getRhsCPLEX</td>
<td>165</td>
</tr>
<tr>
<td>getRngValCPLEX</td>
<td>166</td>
</tr>
<tr>
<td>getRowIndexCPLEX</td>
<td>167</td>
</tr>
<tr>
<td>getRowInfeasCPLEX</td>
<td>168</td>
</tr>
<tr>
<td>getRowNameCPLEX</td>
<td>169</td>
</tr>
<tr>
<td>getRowsCPLEX</td>
<td>170</td>
</tr>
<tr>
<td>getSenseCPLEX</td>
<td>171</td>
</tr>
<tr>
<td>getSiftItCntCPLEX</td>
<td>172</td>
</tr>
<tr>
<td>getSiftPase1CntCPLEX</td>
<td>173</td>
</tr>
<tr>
<td>getSlackCPLEX</td>
<td>174</td>
</tr>
<tr>
<td>getStatCPLEX</td>
<td>175</td>
</tr>
<tr>
<td>getStatStrCPLEX</td>
<td>176</td>
</tr>
<tr>
<td>getStrParmCPLEX</td>
<td>177</td>
</tr>
<tr>
<td>getSubMethodCPLEX</td>
<td>178</td>
</tr>
<tr>
<td>getSubStatCPLEX</td>
<td>179</td>
</tr>
<tr>
<td>getTimeCPLEX</td>
<td>180</td>
</tr>
<tr>
<td>getUppBndsIdsCPLEX</td>
<td>181</td>
</tr>
<tr>
<td>getUpperBndsCPLEX</td>
<td>182</td>
</tr>
<tr>
<td>getVersionCPLEX</td>
<td>183</td>
</tr>
<tr>
<td>hybbaroptCPLEX</td>
<td>184</td>
</tr>
<tr>
<td>hybnetoptCPLEX</td>
<td>185</td>
</tr>
<tr>
<td>initProbCPLEX</td>
<td>186</td>
</tr>
<tr>
<td>lpoptCPLEX</td>
<td>187</td>
</tr>
<tr>
<td>mipoptCPLEX</td>
<td>188</td>
</tr>
<tr>
<td>newColsCPLEX</td>
<td>189</td>
</tr>
<tr>
<td>newRowsCPLEX</td>
<td>190</td>
</tr>
<tr>
<td>objSaCPLEX</td>
<td>191</td>
</tr>
<tr>
<td>openEnvCPLEX</td>
<td>192</td>
</tr>
<tr>
<td>openFileCPLEX</td>
<td>193</td>
</tr>
<tr>
<td>openProbCPLEX</td>
<td>194</td>
</tr>
<tr>
<td>ordWriteCPLEX</td>
<td>195</td>
</tr>
<tr>
<td>preslvWriteCPLEX</td>
<td>196</td>
</tr>
<tr>
<td>presolveCPLEX</td>
<td>197</td>
</tr>
<tr>
<td>primaloptCPLEX</td>
<td>198</td>
</tr>
</tbody>
</table>
printTerminateCPLEX, 199
qoptCPLEX, 199
readCopyBaseCPLEX, 200
readCopyMIPstartsCPLEX, 201
readCopyOrderCPLEX, 202
readCopyParmCPLEX, 203
readCopyProbCPLEX, 204
readCopySolCPLEX, 205
refineConflictCPLEX, 206
refineConflictExtCPLEX, 207
refineMIPstartConflictCPLEX, 208
refineMIPstartConflictExtCPLEX, 209
return_codeCPLEX, 210
rhsSaCPLEX, 211
setDb1ParmCPLEX, 212
setDefaultParmCPLEX, 213
setIntParmCPLEX, 214
setLogFileCPLEX, 215
setLongParmCPLEX, 216
setObjDirCPLEX, 217
setStrParmCPLEX, 218
setTerminateCPLEX, 219
siftoptCPLEX, 220
solnInfoCPLEX, 221
solutionCPLEX, 222
solWriteCPLEX, 223
status_codeCPLEX, 224
tightenBndsCPLEX, 225
tuneParmCPLEX, 226
unscaleProbCPLEX, 227
writeMIPstartsCPLEX, 228
writeParmCPLEX, 229
writeProbCPLEX, 230

*Topic package
  cplexAPI-package, 6
  cplexConstants(cplexConstants), 61
  cplexAPI(cplexAPI-package), 6
  cplexConstants, 38, 61, 106, 112, 114, 116.

addColsCPLEX, 7, 14, 19
addFpDestCPLEX, 9, 85
addIndConstrCPLEX, 10
addMIPstartsCPLEX, 11
addQConstrCPLEX, 12
addRowsCPLEX, 8, 13, 20
baroptCPLEX, 14
baseWriteCPLEX, 15
basicPresolveCPLEX, 16
boundsSaCPLEX, 17

checkAddColsCPLEX, 8, 18
checkAddRowsCPLEX, 14, 19
checkChgCoefListCPLEX, 20
checkCopyColTypeCPLEX, 21
checkCopyLpCPLEX, 22
checkCopyLpWNamesCPLEX, 23
checkCopyQPsepCPLEX, 25
checkCopyQuadCPLEX, 26
checkValsCPLEX, 27
chgBndsCPLEX, 28, 32, 225
chgCoefCPLEX, 29, 39
chgCoefListCPLEX, 27, 30
chgColNameCPLEX, 31
chgColsBndsCPLEX, 32
chgColTypeCPLEX, 33
chgMIPstartsCPLEX, 34
chgNameCPLEX, 35
chgObjCPLEX, 29, 36
chgProbNameCPLEX, 37
chgProbTypeCPLEX, 38, 160
chgQPCoeffCPLEX, 39
chgRhsCPLEX, 29, 40
chgRngValCPLEX, 14, 29, 41
chgRowNameCPLEX, 42
chgSenseCPLEX, 43
chgTerminateCPLEX, 44, 94, 199, 219
cleanupCplexCplex, 44
closeProbCPLEX, 45
closeEnvCPLEX, 46, 192
closefileCPLEX, 47, 99, 193
closeProbCPLEX, 48, 194
clpWriteCPLEX, 49
completemipCPLEX, 50
constantsCPLEX(cplexConstants), 61
copyBaseCPLEX, 51
copyColTypeCPLEX, 22, 52
copyLpCPLEX, 14, 23, 53
copyLpWNamesCPLEX, 25, 54
copyObjNameCPLEX, 55
copyOrderCPLEX, 56
copyPartBaseCPLEX, 57
copyQPsepCPLEX, 25, 58
copyQuadCPLEX, 27, 59
copyStartCPLEX, 60
cplex_Constants(cplexConstants), 61
cplexAPI(cplexAPI-package), 6
cplexAPI-package, 6
cplexConstants, 38, 61, 106, 112, 114, 116.
| 117, 124–127, 131, 134, 147, 151–155, 158, 160, 175, 177, 178, 184, 185, 190, 195, 197, 203, 210, 212–214, 216–218, 221, 223, 224, 229 |
| cplexError (cplexError-class), 82 |
| cplexPointer (cplexPtr-class), 83 |
| cplexPointer, cplexPtr-method (cplexPtr-class), 83 |
| cplexPtr (cplexPtr-class), 83 |
| cplexPtr-class, 83 |
| cplexPtrType (cplexPtr-class), 83 |
| cplexPtrType, cplexPtr-method (cplexPtr-class), 83 |
| cplexPtrType< (cplexPtr-class), 83 |
| cplexPtrType<, cplexPtr-method (cplexPtr-class), 83 |
| CPX_ALG_AUTOMATIC (cplexConstants), 61 |
| CPX_ALG_BAROPT (cplexConstants), 61 |
| CPX_ALG_BARRIER (cplexConstants), 61 |
| CPX_ALG_CONCURRENT (cplexConstants), 61 |
| CPX_ALG_DUAL (cplexConstants), 61 |
| CPX_ALG_FEASOPT (cplexConstants), 61 |
| CPX_ALG_MIP (cplexConstants), 61 |
| CPX_ALG_NET (cplexConstants), 61 |
| CPX_ALG_POINTS (cplexConstants), 61 |
| CPX_ALG_PIVOT (cplexConstants), 61 |
| CPX_ALG_PIVOTOUT (cplexConstants), 61 |
| CPX_ALG_PIVOTIN (cplexConstants), 61 |
| CPX_ALG_PRIMAL (cplexConstants), 61 |
| CPX_ALG_ROBUST (cplexConstants), 61 |
| CPX_ALG_SIFTING (cplexConstants), 61 |
| CPX_AT_LOWER (cplexConstants), 61 |
| CPX_AT_UPPER (cplexConstants), 61 |
| CPX_BARORDER_AMD (cplexConstants), 61 |
| CPX_BARORDER_AMF (cplexConstants), 61 |
| CPX_BARORDER_AUTO (cplexConstants), 61 |
| CPX_BARORDER_ND (cplexConstants), 61 |
| CPX_BASIC (cplexConstants), 61 |
| CPX_BASIC_SOLN (cplexConstants), 61 |
| CPX_BINARY (cplexConstants), 61 |
| CPX_BRANCH_DOWN (cplexConstants), 61 |
| CPX_BRANCH_GLOBAL (cplexConstants), 61 |
| CPX_BRANCH_UP (cplexConstants), 61 |
| CPX_BRDIR_AUTO (cplexConstants), 61 |
| CPX_BRDIR_DOWN (cplexConstants), 61 |
| CPX_BRDIR_UP (cplexConstants), 61 |
| CPX_CON_ABS (cplexConstants), 61 |
| CPX_CON_Disjunctive (cplexConstants), 61 |
| CPX_CON_INDI_DISJUNCTIVE (cplexConstants), 61 |
| CPX_CON_INDICATOR (cplexConstants), 61 |
| CPX_CON_LAST_CONTYPE (cplexConstants), 61 |
| CPX_CON_LINEAR (cplexConstants), 61 |
| CPX_CON_LOWER_BOUND (cplexConstants), 61 |
| CPX_CON_MAXEXPR (cplexConstants), 61 |
| CPX_CON_MINEXPR (cplexConstants), 61 |
| CPX_CON_PWL (cplexConstants), 61 |
| CPX_CON_QUADRATIC (cplexConstants), 61 |
| CPX_CON_SETVAR (cplexConstants), 61 |
| CPX_CON_SETVARCARD (cplexConstants), 61 |
| CPX_CON_SETVARDOMAIN (cplexConstants), 61 |
| CPX_CON_SETVAREQ (cplexConstants), 61 |
| CPX_CON_SETVARINTERSECT (cplexConstants), 61 |
| CPX_CON_SETVARINTERSECTION (cplexConstants), 61 |
| CPX_CON_SETVARMAX (cplexConstants), 61 |
| CPX_CON_SETVARMEMBER (cplexConstants), 61 |
| CPX_CON_SETVARMIN (cplexConstants), 61 |
| CPX_CON_SETVARNEQ (cplexConstants), 61 |
| CPX_CON_SETVARNEQ_GAST (cplexConstants), 61 |
| CPX_CON_SETVARNULLINTERSECT (cplexConstants), 61 |
| CPX_CON_SETVARSUBSET (cplexConstants), 61 |
| CPX_CON_SETVARSUM (cplexConstants), 61 |
| CPX_CON_SETVARUNION (cplexConstants), 61 |
| CPX_CON_SOC (cplexConstants), 61 |
| CPX_CON_UPPERBOUND (cplexConstants), 61 |
| CPX_CON_UPPERBOUND (cplexConstants), 61 |
| CPX_CONFLICT_EXCLUDED (cplexConstants), 61 |
| CPX_CONFLICT_LB (cplexConstants), 61 |
| CPX_CONFLICT_MEMBER (cplexConstants), 61 |
INDEX

CPX_CONFLICT_POSSIBLE_LB (cplexConstants), 61
CPX_CONFLICT_POSSIBLE_MEMBER (cplexConstants), 61
CPX_CONFLICT_POSSIBLE UB (cplexConstants), 61
CPX_CONFLICT UB (cplexConstants), 61
CPX_CONTINUOUS (cplexConstants), 61
CPX_DPRIIND_AUTO (cplexConstants), 61
CPX_DPRIIND_DEVEX (cplexConstants), 61
CPX_DPRIIND_FULL (cplexConstants), 61
CPX_DPRIIND_FULL_STEEP (cplexConstants), 61
CPX_DPRIIND_STEEPQSTART (cplexConstants), 61
CPX_DPRIIND_STEEP (cplexConstants), 61
CPX_DPRIIND_STEEPQSTART (cplexConstants), 61
CPX_DUAL_OBJ (cplexConstants), 61
CPX_EXACT_KAPPA (cplexConstants), 61
CPX_FEASOPT_MIN_INF (cplexConstants), 61
CPX_FEASOPT_MIN_QUAD (cplexConstants), 61
CPX_FEASOPT_MIN_SUM (cplexConstants), 61
CPX_FEASOPT_OPT_INF (cplexConstants), 61
CPX_FEASOPT_OPT_QUAD (cplexConstants), 61
CPX_FEASOPT_OPT_SUM (cplexConstants), 61
CPX_FREE_SUPER (cplexConstants), 61
CPX_IMPLIED_INTEGER_FEASIBLE (cplexConstants), 61
CPX_INFBOUND (cplexConstants), 61
CPX_INTEGER (cplexConstants), 61
CPX_INTEGER_FEASIBLE (cplexConstants), 61
CPX_INTEGER_INFEASIBLE (cplexConstants), 61
CPX_KAPPA (cplexConstants), 61
CPX_KAPPA_ATTENTION (cplexConstants), 61
CPX_KAPPA_IllPOSED (cplexConstants), 61
CPX_KAPPA_MAX (cplexConstants), 61
CPX_KAPPA_STABLE (cplexConstants), 61
CPX_KAPPA_SUSPICIOUS (cplexConstants), 61
CPX_KAPPA_UNSTABLE (cplexConstants), 61
CPX_MAX (cplexConstants), 61
CPX_MAX_COMP_SLACK (cplexConstants), 61
CPX_MAX_DUAL_INFEAS (cplexConstants), 61
CPX_MAX_DUAL_RESIDUAL (cplexConstants), 61
CPX_MAX_INDSLACK_INFEAS (cplexConstants), 61
CPX_MAX_INT_INFEAS (cplexConstants), 61
CPX_MAX_PI (cplexConstants), 61
CPX_MAX_PRIMAL_INFEAS (cplexConstants), 61
CPX_MAX_PRIMAL_RESIDUAL (cplexConstants), 61
CPX_MAX_QCPRIMAL_RESIDUAL (cplexConstants), 61
CPX_MAX_QCSLACK (cplexConstants), 61
CPX_MAX_QCSLACK_INFEAS (cplexConstants), 61
CPX_MAX_RED_COST (cplexConstants), 61
CPX_MAX_SCALED_DUAL_INFEAS (cplexConstants), 61
CPX_MAX_SCALED_DUAL_RESIDUAL (cplexConstants), 61
CPX_MAX_SCALED_PI (cplexConstants), 61
CPX_MAX_SCALED_PRIMAL_INFEAS (cplexConstants), 61
CPX_MAX_SCALED_PRIMAL_RESIDUAL (cplexConstants), 61
CPX_MAX_SCALED_RED_COST (cplexConstants), 61
CPX_MAX_SCALED_SLACK (cplexConstants), 61
CPX_MAX_SCALED_X (cplexConstants), 61
CPX_MAX_SLACK (cplexConstants), 61
CPX_MAX_X (cplexConstants), 61
CPX_MIN (cplexConstants), 61
CPX_MIPEMPHASIS_BALANCED (cplexConstants), 61
CPX_MIPEMPHASIS_BESTBOUND (cplexConstants), 61
CPX_MIPEMPHASIS_FEASIBILITY (cplexConstants), 61
CPX_MIPEMPHASIS_HIDDENFEAS (cplexConstants), 61
CPX_MIPEMPHASIS_OPTIMALITY (cplexConstants), 61
CPX_MIPKAPPA_AUTO (cplexConstants), 61
CPX_MIPKAPPA_FULL (cplexConstants), 61
CPX_MIPKAPPA_OFF (cplexConstants), 61
CPX_MIPKAPPA_SAMPLE (cplexConstants), 61
CPX_MIPORDER_BOUNDS (cplexConstants), 61
CPX_MIPORDER_COST (cplexConstants), 61
CPX_MIPORDER_SCALED_COST
INDEX

CPX_PARAM_EPRHS (cplexConstants), 61
CPX_PARAM_EPRHS_H (cplexConstants), 61
CPX_PARAM_EPRHSH (cplexConstants), 61
CPX_PARAM_EPRELAX (cplexConstants), 61
CPX_PARAM_EPPER (cplexConstants), 61
CPX_PARAM_EPRHS (cplexConstants), 61
CPX_PARAM_FASTMIP (cplexConstants), 61
CPX_PARAM_FEASOPTMODE (cplexConstants), 61
CPX_PARAM_FILEENCODING (cplexConstants), 61
CPX_PARAM_FLOWCOVERS (cplexConstants), 61
CPX_PARAM_FLOWPATHS (cplexConstants), 61
CPX_PARAM_FPHEUR (cplexConstants), 61
CPX_PARAM_FRACCAND (cplexConstants), 61
CPX_PARAM_FRACCOVERS (cplexConstants), 61
CPX_PARAM_FRACPASS (cplexConstants), 61
CPX_PARAM_SUBCOVERS (cplexConstants), 61
CPX_PARAM_HUERFREQ (cplexConstants), 61
CPX_PARAM_INTEGRALFILEPREFIX (cplexConstants), 61
CPX_PARAM_INTEGRALFILEPREFIX (cplexConstants), 61
CPX_PARAM_INTSOLFILEPREFIX (cplexConstants), 61
CPX_PARAM_INTSOLLIM (cplexConstants), 61
CPX_PARAM_OTOLIM (cplexConstants), 61
CPX_PARAM_LBHEUR (cplexConstants), 61
CPX_PARAM_LPMETHOD (cplexConstants), 61
CPX_PARAM_MCFCUTS (cplexConstants), 61
CPX_PARAM_MEMORYEMPHASIS (cplexConstants), 61
CPX_PARAM_MIPCBREDLDP (cplexConstants), 61
CPX_PARAM_MIPDISPLAY (cplexConstants), 61
CPX_PARAM_MIPEMPHASIS (cplexConstants), 61
CPX_PARAM_MIPINTERVAL (cplexConstants), 61
CPX_PARAM_MIPKAPPASTATS (cplexConstants), 61
CPX_PARAM_MIPORDIND (cplexConstants), 61
CPX_PARAM_MIPORDTYPE (cplexConstants), 61
CPX_PARAM_MIPSEARCH (cplexConstants), 61
CPX_PARAM_MIQCPSTRAT (cplexConstants), 61
CPX_PARAM_MIRCUTS (cplexConstants), 61
CPX_PARAM_MPSLONGNUM (cplexConstants), 61
CPX_PARAM_NETDISPLAY (cplexConstants), 61
CPX_PARAM_NETDISPLAY (cplexConstants), 61
CPX_PARAM_NETEPOPT (cplexConstants), 61
CPX_PARAM_NETPRHS (cplexConstants), 61
CPX_PARAM_NETFIND (cplexConstants), 61
CPX_PARAM_NETITLIM (cplexConstants), 61
CPX_PARAM_NETPRIIND (cplexConstants), 61
CPX_PARAM_NODFILEINDEX (cplexConstants), 61
CPX_PARAM_NODELIM (cplexConstants), 61
CPX_PARAM_NODESEL (cplexConstants), 61
CPX_PARAM_NUMERICALEMPHASIS (cplexConstants), 61
CPX_PARAM_NZREADLIM (cplexConstants), 61
CPX_PARAM_OBJDISPLAY (cplexConstants), 61
CPX_PARAM_OBJLIM (cplexConstants), 61
CPX_PARAM_OBJLIM (cplexConstants), 61
CPX_PARAM_PARALLELMODE (cplexConstants), 61
CPX_PARAM_PERIND (cplexConstants), 61
CPX_PARAM_PERLIM (cplexConstants), 61
CPX_PARAM_POLISHAFTERDETTIME (cplexConstants), 61
CPX_PARAM_POLISHAFTERDETTIME (cplexConstants), 61
CPX_PARAM_POLISHAFTEREPAGAP (cplexConstants), 61
CPX_PARAM_POLISHAFTEREPAGAP (cplexConstants), 61
CPX_PARAM_POLISHAFTERINTSOL (cplexConstants), 61
CPX_PARAM_POLISHAFTERINTSOL (cplexConstants), 61
CPX_PARAM_POLISHAFTERNODE (cplexConstants), 61
CPX_PARAM_POLISHAFTERNODE (cplexConstants), 61
CPX_PARAM_POLISHAFTERTIME (cplexConstants), 61
CPX_PARAM_POLISHAFTERTIME (cplexConstants), 61
CPX_PARAM_POPULATELIM (cplexConstants), 61
CPX_PARAM_PPRIIND (cplexConstants), 61
CPX_PARAM_PREDUAL (cplexConstants), 61
CPX_PARAM_PREIND (cplexConstants), 61
CPX_PARAM_PRELINEAR (cplexConstants), 61
CPX_PARAM_PREPASS (cplexConstants), 61
CPX_PARAM_PRESLIND (cplexConstants), 61
CPX_PARAM_PRICELIM (cplexConstants), 61
CPX_PARAM_PROBE (cplexConstants), 61
CPX_PARAM_PROBEDETTIME
  (cplexConstants), 61
CPX_PARAM_PROBETIME (cplexConstants), 61
CPX_PARAM_QPMAKEPSDIND
  (cplexConstants), 61
CPX_PARAM_QPMETHOD (cplexConstants), 61
CPX_PARAM_QPNZREADLIM (cplexConstants), 61
CPX_PARAM_QPMAKEPSDIND (cplexConstants), 61
CPX_PARAM_QPMAKETIME (cplexConstants), 61
CPX_PARAM_QPNZREADLIM (cplexConstants), 61
CPX_PARAM_RAMPUPDETTILIM
  (cplexConstants), 61
CPX_PARAM_RAMPUPDURATION
  (cplexConstants), 61
CPX_PARAM_RAMPUPTILIM (cplexConstants), 61
CPX_PARAM_RANDOMSEED (cplexConstants), 61
CPX_PARAM_REDUCE (cplexConstants), 61
CPX_PARAM_REINV (cplexConstants), 61
CPX_PARAM_RELAXPREIND (cplexConstants), 61
CPX_PARAM_RELOBJDIF (cplexConstants), 61
CPX_PARAM_REPAIRTRIES (cplexConstants), 61
CPX_PARAM_REPEATPRESOLVE
  (cplexConstants), 61
CPX_PARAM_REVERSEIND (cplexConstants), 61
CPX_PARAM_RFILEMUL (cplexConstants), 61
CPX_PARAM_RINSHEUR (cplexConstants), 61
CPX_PARAM_ROWREADLIM (cplexConstants), 61
CPX_PARAM_SCAIND (cplexConstants), 61
CPX_PARAM_SCRIND (cplexConstants), 61
CPX_PARAM_SETTALG (cplexConstants), 61
CPX_PARAM_SFDISPLAY (cplexConstants), 61
CPX_PARAM_SFTTILIM (cplexConstants), 61
CPX_PARAM_SDTDISPLAY (cplexConstants), 61
CPX_PARAM_SIMDISPLAY (cplexConstants), 61
CPX_PARAM_SINGLIM (cplexConstants), 61
CPX_PARAM_SINGTOL (cplexConstants), 61
CPX_PARAM_SOLNPOLLAGAP
  (cplexConstants), 61
CPX_PARAM_SOLNPOLLCAPACITY
  (cplexConstants), 61
CPX_PARAM_SOLNPOLLGAP (cplexConstants), 61
CPX_PARAM_SOLNPOLLINTENSITY
  (cplexConstants), 61
CPX_PARAM_SOLNPOLLREPLACE
  (cplexConstants), 61
CPX_PARAM_SOLUTIONTARGET
  (cplexConstants), 61
CPX_PARAM_STARTALG (cplexConstants), 61
CPX_PARAM_STRONGCANDLIM
  (cplexConstants), 61
CPX_PARAM_STRONGITLIM (cplexConstants), 61
CPX_PARAM_SUBALG (cplexConstants), 61
CPX_PARAM_SUBMIPNODELIM
  (cplexConstants), 61
CPX_PARAM_SYMMETRY (cplexConstants), 61
CPX_PARAM_THREADS (cplexConstants), 61
CPX_PARAM_TILIM (cplexConstants), 61
CPX_PARAM_TRELIM (cplexConstants), 61
CPX_PARAM_TUNINGDETTILIM
  (cplexConstants), 61
CPX_PARAM_TUNINGDISPLAY
  (cplexConstants), 61
CPX_PARAM_TUNINGMEASURE
  (cplexConstants), 61
CPX_PARAM_TUNINGREPEAT
  (cplexConstants), 61
CPX_PARAM_TUNINGREPEATEAT
  (cplexConstants), 61
CPX_PARAM_TUNINGTILIM (cplexConstants), 61
CPX_PARAM_VARSEL (cplexConstants), 61
CPX_PARAM_WRITELEVEL
  (cplexConstants), 61
CPX_PARAM_TYPE_DOUBLE (cplexConstants), 61
CPX_PARAM_TYPE_INT (cplexConstants), 61
CPX_PARAM_TYPE_LONG (cplexConstants), 61
CPX_PARAM_TYPE_NONE (cplexConstants), 61
CPX_PARAM_TYPE_STRING (cplexConstants), 61
CPX_PARAMTYPE_DOUBLE (cplexConstants), 61
CPX_PARAMTYPE_INT (cplexConstants), 61
CPX_PARAMTYPE_LONG (cplexConstants), 61
CPX_PARAMTYPE_NONE (cplexConstants), 61
CPX_PARAMTYPE_STRING (cplexConstants), 61
CPX_PPRIIND_AUTO (cplexConstants), 61
CPX_PPRIIND_DEVEX (cplexConstants), 61
CPX_PPRIIND_FULL (cplexConstants), 61
CPX_PPRIIND_PARTIAL (cplexConstants), 61
CPX_PPRIIND_STEEP (cplexConstants), 61
INDEX

CPX_PPRIIND_STEEPQSTART
  (cplexConstants), 61
CPX_PRECOL_AGG (cplexConstants), 61
CPX_PRECOL_FIX (cplexConstants), 61
CPX_PRECOL_LOW (cplexConstants), 61
CPX_PRECOL_OTHER (cplexConstants), 61
CPX_PRECOL_UP (cplexConstants), 61
CPX_PREREDEUCE_DUALONLY
  (cplexConstants), 61
CPX_PREREDEUCE_NOPRIMALORDUAL
  (cplexConstants), 61
CPX_PREREDEUCE_PRIMALANDDUAL
  (cplexConstants), 61
CPX_PREREDEUCE_PRIMALONLY
  (cplexConstants), 61
CPX_PREROW_AGG (cplexConstants), 61
CPX_PREROW_RED (cplexConstants), 61
CPX_PRIMAL_OBJ (cplexConstants), 61
CPX_PRIMAL_SOLN (cplexConstants), 61
CPX_SEMICONT (cplexConstants), 61
CPX_SEMIINT (cplexConstants), 61
CPX_SOLNPPOOL_DIV (cplexConstants), 61
CPX_SOLNPPOOL_FIFO (cplexConstants), 61
CPX_SOLNPPOOL_FILTER_DIVERSITY
  (cplexConstants), 61
CPX_SOLNPPOOL_FILTER_RANGE
  (cplexConstants), 61
CPX_SOLNPPOOL_OBJ (cplexConstants), 61
CPX_SOLUTIONTARGET_AUTO
  (cplexConstants), 61
CPX_SOLUTIONTARGET_FIRSTORDER
  (cplexConstants), 61
CPX_SOLUTIONTARGET_OPTIMALCONVEX
  (cplexConstants), 61
CPX_SOLUTIONTARGET_OPTIMALGLOBAL
  (cplexConstants), 61
CPX_STAT_ABORT_DETTIME_LIM
  (cplexConstants), 61
CPX_STAT_ABORT_DUAL_OBJ_LIM
  (cplexConstants), 61
CPX_STAT_ABORT_IT_LIM
  (cplexConstants), 61
CPX_STAT_ABORT_OBJ_LIM
  (cplexConstants), 61
CPX_STAT_ABORT_PRIM_OBJ_LIM
  (cplexConstants), 61
CPX_STAT_ABORT_TIME_LIM
  (cplexConstants), 61
CPX_STAT_ABORT_USER (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_CONTRADICTION
  (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_DETTIME_LIM
  (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_IT_LIM
  (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_MEM_LIM
  (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_NODE_LIM
  (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_OBJ_LIM
  (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_TIME_LIM
  (cplexConstants), 61
CPX_STAT_CONFLICT_ABORT_USER
  (cplexConstants), 61
CPX_STAT_CONFLICT_FEASIBLE
  (cplexConstants), 61
CPX_STAT_CONFLICT_MINIMAL
  (cplexConstants), 61
CPX_STAT_FEASIBLE
  (cplexConstants), 61
CPX_STAT_FEASIBLE_RELAXED_INF
  (cplexConstants), 61
CPX_STAT_FEASIBLE_RELAXED_QUAD
  (cplexConstants), 61
CPX_STAT_FEASIBLE_RELAXED_SUM
  (cplexConstants), 61
CPX_STAT_FIRSTORDER (cplexConstants), 61
CPX_STAT_INFEASIBLE (cplexConstants), 61
CPX_STAT_INFORUNBD (cplexConstants), 61
CPX_STAT_NUM_BEST (cplexConstants), 61
CPX_STAT_OPTIMAL (cplexConstants), 61
CPX_STAT_OPTIMAL_FACE_UNBOUNDED
  (cplexConstants), 61
CPX_STAT_OPTIMAL_INFEAS
  (cplexConstants), 61
CPX_STAT_OPTIMAL_RELAXED_INF
  (cplexConstants), 61
CPX_STAT_OPTIMAL_RELAXED_QUAD
  (cplexConstants), 61
CPX_STAT_OPTIMAL_RELAXED_SUM
  (cplexConstants), 61
CPX_STAT_UNBOUNDED (cplexConstants), 61
CPX_STR_PARAM_MAX (cplexConstants), 61
CPX_SUM_COMP_SLACK (cplexConstants), 61
CPX_SUM_DUAL_INFEAS (cplexConstants), 61
<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPX_MIP_TIME_LIM_INFEAS</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPX_MIP_UNBOUNDED</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPX_MIP_DISPLAY_OBJECTIVE</td>
<td><em>cplexConstants</em>, 61</td>
</tr>
<tr>
<td>CPX_NET_DISPLAY_OBJECTIVE</td>
<td><em>cplexConstants</em>, 61</td>
</tr>
<tr>
<td>CPX_NET_PRICE_AUTO</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPX_PARAM_CUTOFF_MULT</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPX_PARAM_CUTOFF_SORT_MULT</td>
<td><em>cplexConstants</em>, 61</td>
</tr>
<tr>
<td>CPX_PARAM_TRUE_OBJECTIVE</td>
<td><em>cplexConstants</em>, 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_ALGORITHM</td>
<td><em>cplexConstants</em>, 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_COLUMN</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_CONVERGE_TOL</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_CROSSOVER</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_DISPLAY</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_LIMITS_CORRECTIONS</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_LIMITS_GROWTH</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_LIMITS_ITERATION</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_LIMITS_OBJ_RANGE</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_ORDERING</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_QCP_CONVERGE_TOL</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_BARRIER_START_ALGORITHM</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_CLOCK_TYPE</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_CONFLICT_DISPLAY</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_DET_TIME_LIMIT</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_DIST_MIP_RAMPUP_DET_TIME_LIMIT</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_DIST_MIP_RAMPUP_DURATION</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_DIST_MIP_RAMPUP_TIME_LIMIT</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_EMPHASIS_MEMORY</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_EMPHASIS_MIP</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_EMPHASIS_NUMERICAL</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_FEASOPT_MODE</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_FEASOPT_TOLERANCE</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_LP_METHOD</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_CLIQUES</td>
<td><em>cplexConstants</em>, 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_COVERS</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_DISJUNCTIVE</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_FLOW_COVERS</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_GOMORY</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_GUB_COVERS</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_IMPLIED</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_LIFTPROJ</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_MCF</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_MIRC</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_PATH_CUT</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_CUTS_ZERO_HALF</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_DISPLAY</td>
<td>(cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Interval (cplexConstants), 61</td>
<td>CPXPARAM_MIP_PolishAfter_Solutions (cplexConstants), 61</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_AggForCut (cplexConstants), 61</td>
<td>CPXPARAM_MIP_PolishAfter_Time (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_AuxRootThreads (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Pool_AbsGap (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_CutPasses (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Pool_Capacity (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_CutsFactor (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Pool_Intensity (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_EachCutLimit (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Pool_RelGap (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_GomoryCand (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Pool_Replacement (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_GomoryPass (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_Backtrack (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_NODES (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_BBIInterval (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_PolishTime (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy.Branch (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_Populate (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_CallbackReducedLP (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_ProbeDetTime (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_Dive (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_ProbeTime (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_File (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_RepairTries (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_FPHeur (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_Solutions (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_HeuristicFreq (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_StrongCand (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_KappaStats (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_StrongIt (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_LBHeur (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_SubMIPNodeLimit (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_MIQCPStrat (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_Limits_TreeMemory (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_NodeSelect (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_OrderType (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_Order (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter.AbsMIPGap (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_PresolveNode (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_DET (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_Probe (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_MIPGap (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_RINSHeur (cplexConstants), 61</td>
</tr>
<tr>
<td>CPXPARAM_MIP_PolishAfter_Nodes (cplexConstants), 61</td>
<td>CPXPARAM_MIP_Strategy_Search (cplexConstants), 61</td>
</tr>
</tbody>
</table>
CPXPARAM_MIP_Strategy_StartAlgorithm (cplexConstants), 61
CPXPARAM_MIP_Strategy_SubAlgorithm (cplexConstants), 61
CPXPARAM_MIP_Strategy_VariableSelect (cplexConstants), 61
CPXPARAM_MIP_Tolerances_AbsMIPGap (cplexConstants), 61
CPXPARAM_MIP_Tolerances_Integrality (cplexConstants), 61
CPXPARAM_MIP_Tolerances_LowerCutoff (cplexConstants), 61
CPXPARAM_MIP_Tolerances_MIPGap (cplexConstants), 61
CPXPARAM_MIP_Tolerances_ObjDifference (cplexConstants), 61
CPXPARAM_MIP_Tolerances_RelObjDifference (cplexConstants), 61
CPXPARAM_MIP_Tolerances_UpperCutoff (cplexConstants), 61
CPXPARAM_Network_Display (cplexConstants), 61
CPXPARAM_Network_Iterations (cplexConstants), 61
CPXPARAM_Network_NetFind (cplexConstants), 61
CPXPARAM_Network_Pricing (cplexConstants), 61
CPXPARAM_Network_Tolerances_Feasibility (cplexConstants), 61
CPXPARAM_Network_Tolerances_Optimality (cplexConstants), 61
CPXPARAM_Output_CloneLog (cplexConstants), 61
CPXPARAM_Output_IntSolFilePrefix (cplexConstants), 61
CPXPARAM_Output_MPSSlong (cplexConstants), 61
CPXPARAM_Output_WriteLevel (cplexConstants), 61
CPXPARAM_Parallel (cplexConstants), 61
CPXPARAM_Preprocessing_Aggregator (cplexConstants), 61
CPXPARAM_Preprocessing_BoundStrength (cplexConstants), 61
CPXPARAM_Preprocessing_CoeffReduce (cplexConstants), 61
CPXPARAM_Preprocessing_Dependency (cplexConstants), 61
CPXPARAM_Preprocessing_Dual (cplexConstants), 61
CPXPARAM_Preprocessing_Fill (cplexConstants), 61
CPXPARAM_Preprocessing_Linear (cplexConstants), 61
CPXPARAM_Preprocessing_NumPass (cplexConstants), 61
CPXPARAM_Preprocessing_Presolve (cplexConstants), 61
CPXPARAM_Preprocessing_QCPDuals (cplexConstants), 61
CPXPARAM_Preprocessing_QPMakePSD (cplexConstants), 61
CPXPARAM_Preprocessing_Reduce (cplexConstants), 61
CPXPARAM_Preprocessing_Relax (cplexConstants), 61
CPXPARAM_Preprocessing_RepeatPresolve (cplexConstants), 61
CPXPARAM_Preprocessing_Symmetry (cplexConstants), 61
CPXPARAM_QPMethod (cplexConstants), 61
CPXPARAM_Read_APIEncoding (cplexConstants), 61
CPXPARAM_Read_Constraints (cplexConstants), 61
CPXPARAM_Read_DataCheck (cplexConstants), 61
CPXPARAM_Read_FileEncoding (cplexConstants), 61
CPXPARAM_Read_NonZeros (cplexConstants), 61
CPXPARAM_Read_QPNonZeros (cplexConstants), 61
CPXPARAM_Read_Scale (cplexConstants), 61
CPXPARAM_Read_Variables (cplexConstants), 61
CPXPARAM_ScreenOutput (cplexConstants), 61
CPXPARAM_Sifting_Algorithm (cplexConstants), 61
CPXPARAM_Sifting_Display (cplexConstants), 61
CPXPARAM_Sifting_Iterations (cplexConstants), 61
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPXPARAM_Simplex_Crash</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_DGradient</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Display</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SimplexLimits_Iterations</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SimplexLimits_LowerObj</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SimplexLimits_Perturbation</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SimplexLimits_Singularity</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SimplexLimits_UpperObj</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SimplexPerturbation_Constant</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SimplexPerturbation_Indicator</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_PGradient</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Pricing</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Refactor</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Tolerances_Feasibility</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Tolerances_Markowitz</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Simplex_Tolerances_Optimality</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_SolutionTarget</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Threads</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_TimeLimit</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Tune_DetTimeLimit</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Tune_Display</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Tune.Measure</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Tune_Repeat</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_Tune_TimeLimit</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_WorkDir</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPARAM_WorkMem</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPRESLVWRITE</td>
<td>(preslWriteCPLEX), 196</td>
<td></td>
</tr>
<tr>
<td>CPXPRESOLVE</td>
<td>(presolveCPLEX), 197</td>
<td></td>
</tr>
<tr>
<td>CPXPRIMOPT</td>
<td>(primoptCPLEX), 198</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_FIXEDMILP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_FIXEDMIQP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_LP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_MIQP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_MIQCP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_NODELP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_NODEQCP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_PROB_QCP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXPB_QP</td>
<td>(cplexConstants), 61</td>
<td></td>
</tr>
<tr>
<td>CPXQPRESLVWRITE</td>
<td>(qpsolveCPLEX), 199</td>
<td></td>
</tr>
<tr>
<td>CPXREADCOPYBASE</td>
<td>(readCopyBaseCPLEX), 200</td>
<td></td>
</tr>
<tr>
<td>CPXREADCOPYSTARTS</td>
<td>(readCopyMIPstartsCLEX), 201</td>
<td></td>
</tr>
<tr>
<td>CPXREADCOPYORDER</td>
<td>(readCopyOrderCLEX), 202</td>
<td></td>
</tr>
<tr>
<td>CPXREADCOPYPARAM</td>
<td>(readCopyParmCLEX), 203</td>
<td></td>
</tr>
<tr>
<td>CPXREADCOPYPROB</td>
<td>(readCopyProbCLEX), 204</td>
<td></td>
</tr>
<tr>
<td>CPXREADCOPYSOL</td>
<td>(readCopySolCLEX), 205</td>
<td></td>
</tr>
<tr>
<td>CPXRREFINECONFLICT</td>
<td>(refineConflictCLEX), 206</td>
<td></td>
</tr>
<tr>
<td>CPXRREFINECONFLICTEXT</td>
<td>(refineConflictExtCLEX), 207</td>
<td></td>
</tr>
<tr>
<td>CPXRREFINEMIPSTARTCONFLICT</td>
<td>(refineMIPstartConflictCLEX), 208</td>
<td></td>
</tr>
<tr>
<td>CPXRREFINEMIPSTARTCONFLICTTEXT</td>
<td>(refineMIPstartConflictExtCLEX), 209</td>
<td></td>
</tr>
<tr>
<td>CPXRHSSA</td>
<td>(rhsAcPLEX), 211</td>
<td></td>
</tr>
<tr>
<td>CPXSETDBLPARAM</td>
<td>(setdblParmCLEX), 212</td>
<td></td>
</tr>
<tr>
<td>CPXSETDEFAULTS</td>
<td>(setDefaultParmCLEX), 213</td>
<td></td>
</tr>
<tr>
<td>CPXSETINTPARAM</td>
<td>(setIntParmCLEX), 214</td>
<td></td>
</tr>
<tr>
<td>CPXSETLOGFILE</td>
<td>(setLogFileCLEX), 215</td>
<td></td>
</tr>
<tr>
<td>CPXSETLONGPARAM</td>
<td>(setLongParmCLEX), 216</td>
<td></td>
</tr>
<tr>
<td>CPXSETSTRPARAM</td>
<td>(setStrParmCLEX), 218</td>
<td></td>
</tr>
<tr>
<td>CPXSETTERMINATE</td>
<td>(delTerminateCLEX), 94</td>
<td></td>
</tr>
<tr>
<td>CPXSIFTOPT</td>
<td>(siftoptCLEX), 220</td>
<td></td>
</tr>
<tr>
<td>CPXSOLINFO</td>
<td>(solInfoCLEX), 221</td>
<td></td>
</tr>
<tr>
<td>CPXSOLUTION</td>
<td>(solutionCLEX), 222</td>
<td></td>
</tr>
<tr>
<td>CPXSOLWRITE</td>
<td>(solWriteCLEX), 223</td>
<td></td>
</tr>
<tr>
<td>CPXtightenbds</td>
<td>(tightenBndsCLEX), 225</td>
<td></td>
</tr>
</tbody>
</table>
INDEX

getPreStatCplex, 157
getProbNameCplex, 158
getProbTypeCplex, 38, 159, 223
getProbVarCplex, 160
getQConstrCplex, 161
getQpcoefCplex, 162
getQuadCplex, 163
getRedLpCplex, 164
getRhsCplex, 165
getRngValCplex, 166
getRowIndexCplex, 167
getRowInfeasCplex, 98, 168
getRowNameCplex, 169
getRowsCplex, 170
getSenseCplex, 171
getSiftItCntCplex, 172
getSiftPaseQcntCplex, 173
getSlackCplex, 174
getStatCplex, 15, 96, 98, 175, 184, 185, 187, 188, 198, 200, 224
getStatStrCplex, 120, 176, 224
getStrParmCplex, 177
getSubMethodCplex, 178
getSubStatCplex, 179
getTimeCplex, 180
getUppBndsIdsCplex, 181
getUpperBndsCplex, 28, 181, 182
getVersionCplex, 183

hybbaroptCplex, 184
hybnetoptCplex, 185


isCplexChanPointer (cplexPtr-class), 83
isCplexChanPointer, cplexPtr-method (cplexPtr-class), 83
isCplexEnvPointer (cplexPtr-class), 83
isCplexEnvPointer, cplexPtr-method (cplexPtr-class), 83
isCplexFilePointer (cplexPtr-class), 83
isCplexFilePointer, cplexPtr-method (cplexPtr-class), 83

lpoptCplex, 187

mipoptCplex, 188

newColsCplex, 189
newRowsCplex, 190

objSaCplex, 191

openFileCplex, 9, 47, 85, 99, 180, 193, 215
openProbCplex, 48, 194

ordWriteCplex, 195

preslvWriteCplex, 196
presolveCplex, 197

primoptCplex, 198
printTerminateCplex, 199

queryCplex, 199

readCopyBaseCplex, 200
readCopyMIPstartsCplex, 201
readCopyOrderCplex, 202
readCopyParmCplex, 203
readCopyProbCplex, 196, 204
readCopySolCplex, 205
refineConflictCplex, 206
refineConflictExtCplex, 207
refineMIPstartConflictCplex, 208
refineMIPstartConflictExtCplex, 209

return_codeCplex, 82, 210

rhsSaCplex, 211

setDb1ParmCplex, 212
setDefaultParmCplex, 213
setIntParmCplex, 214, 216
setLogFileCplex, 130, 215

setLongParmCplex, 216
setObjDirCplex, 217
setStrParmCplex, 218
setTerminateCplex, 44, 94, 199, 219
siftoptCplex, 220
solnInfoCplex, 15, 83, 96, 98, 184, 185, 187, 188, 198, 200, 221, 223
solutionCplex, 15, 83, 96, 98, 184, 185, 187, 188, 198, 200, 221, 222
solWriteCplex, 223
status_codeCplex, 82, 224
summary, cplexPtr-method
    (cplexPtr-class), 83

tightenBndsCplex, 32, 225
tuneParmCplex, 226

unscaleProbCplex, 227

writeMIPstartsCplex, 228
writeParmCplex, 229
writeProbCplex, 230