

# Package: glpkAPI (via r-universe)

August 27, 2024

**Type** Package

**Title** R Interface to C API of GLPK

**Version** 1.3.4

**Date** 2022-11-10

**Depends** R (>= 2.6.0)

**Imports** methods

**Description** R Interface to C API of GLPK, depends on GLPK Version >= 4.42.

**SystemRequirements** GLPK (>= 4.42)

**License** GPL-3 | file LICENSE

**LazyLoad** yes

**Collate** generics.R glpk\_ptrClass.R glpk.R glpkAPI.R zzz.R

**NeedsCompilation** yes

**Date/Publication** 2022-11-10 16:10:12 UTC

**Author** Mihail Anton [cre], Mayo Roettger [ctb], Gabriel Gelius-Dietrich [aut], Louis Luangkesorn [ctb]

**Maintainer** Mihail Anton <mihail.anton@chalmers.se>

**Repository** <https://mihai-sysbio.r-universe.dev>

**RemoteUrl** <https://github.com/cran/glpkAPI>

**RemoteRef** HEAD

**RemoteSha** de7900110b92cda50a16b5b6ec317be57f56c3de

## Contents

glpkAPI-package . . . . .	5
addColsGLPK . . . . .	6
addRowsGLPK . . . . .	7
advBasisGLPK . . . . .	8
bfExistsGLPK . . . . .	9
bfUpdatedGLPK . . . . .	10

checkDupGLPK . . . . .	11
copyProbGLPK . . . . .	12
cpxBasisGLPK . . . . .	13
createIndexGLPK . . . . .	13
delColsGLPK . . . . .	14
deleteIndexGLPK . . . . .	15
delProbGLPK . . . . .	16
delRowsGLPK . . . . .	16
eraseProbGLPK . . . . .	17
factorizeGLPK . . . . .	18
findColGLPK . . . . .	19
findRowGLPK . . . . .	20
getBfcpGLPK . . . . .	21
getBheadGLPK . . . . .	22
getCbindGLPK . . . . .	23
getColDualGLPK . . . . .	24
getColDualIptGLPK . . . . .	25
getColKindGLPK . . . . .	26
getColLowBndGLPK . . . . .	27
getColNameGLPK . . . . .	28
getColPrimGLPK . . . . .	29
getColPrimIptGLPK . . . . .	30
getColsDualGLPK . . . . .	31
getColsDualIptGLPK . . . . .	31
getColsKindGLPK . . . . .	32
getColsLowBndsGLPK . . . . .	33
getColsPrimGLPK . . . . .	34
getColsPrimIptGLPK . . . . .	34
getColsStatGLPK . . . . .	35
getColStatGLPK . . . . .	36
getColsUppBndsGLPK . . . . .	37
getColTypeGLPK . . . . .	38
getColUppBndGLPK . . . . .	39
getDualStatGLPK . . . . .	40
getInteriorParmGLPK . . . . .	41
getMatColGLPK . . . . .	42
getMatRowGLPK . . . . .	43
getMIPParmGLPK . . . . .	44
getNumBinGLPK . . . . .	45
getNumColsGLPK . . . . .	45
getNumIntGLPK . . . . .	46
getNumNnzGLPK . . . . .	47
getNumRowsGLPK . . . . .	48
getObjCoefGLPK . . . . .	48
getObjCoefsGLPK . . . . .	49
getObjDirGLPK . . . . .	50
getObjNameGLPK . . . . .	51
getObjValGLPK . . . . .	52

getObjValIptGLPK . . . . .	52
getPrimStatGLPK . . . . .	53
getProbNameGLPK . . . . .	54
getRbindGLPK . . . . .	55
getRiiGLPK . . . . .	56
getRowDualGLPK . . . . .	57
getRowDualIptGLPK . . . . .	58
getRowLowBndGLPK . . . . .	59
getRowNameGLPK . . . . .	60
getRowPrimGLPK . . . . .	61
getRowPrimIptGLPK . . . . .	62
getRowsDualGLPK . . . . .	63
getRowsDualIptGLPK . . . . .	63
getRowsLowBndsGLPK . . . . .	64
getRowsPrimGLPK . . . . .	65
getRowsPrimIptGLPK . . . . .	66
getRowsStatGLPK . . . . .	66
getRowStatGLPK . . . . .	67
getRowsTypesGLPK . . . . .	68
getRowsUppBndsGLPK . . . . .	69
getRowTypeGLPK . . . . .	70
getRowUppBndGLPK . . . . .	71
getSimplexParmGLPK . . . . .	72
getSjjGLPK . . . . .	73
getSolStatGLPK . . . . .	74
getSolStatIptGLPK . . . . .	75
getUnbndRayGLPK . . . . .	76
glpkConstants . . . . .	77
glpkPtr-class . . . . .	83
initProbGLPK . . . . .	84
loadMatrixGLPK . . . . .	85
mipColsValGLPK . . . . .	86
mipColValGLPK . . . . .	87
mipObjValGLPK . . . . .	88
mipRowsValGLPK . . . . .	88
mipRowValGLPK . . . . .	89
mipStatusGLPK . . . . .	90
mplAllocWkspGLPK . . . . .	91
mplBuildProbGLPK . . . . .	92
mplFreeWkspGLPK . . . . .	93
mplGenerateGLPK . . . . .	94
mplPostsolveGLPK . . . . .	95
mplReadDataGLPK . . . . .	96
mplReadModelGLPK . . . . .	97
printIptGLPK . . . . .	98
printMIPGLPK . . . . .	99
printRangesGLPK . . . . .	100
printSolGLPK . . . . .	101

readIptGLPK . . . . .	102
readLPGLPK . . . . .	103
readMIPGLPK . . . . .	104
readMPSGLPK . . . . .	105
readProbGLPK . . . . .	106
readSolGLPK . . . . .	107
return_codeGLPK . . . . .	108
scaleProbGLPK . . . . .	108
setBfcpGLPK . . . . .	109
setColBndGLPK . . . . .	110
setColKindGLPK . . . . .	111
setColNameGLPK . . . . .	112
setColsBndsGLPK . . . . .	113
setColsBndsObjCoefsGLPK . . . . .	114
setColsKindGLPK . . . . .	115
setColsNamesGLPK . . . . .	116
setColStatGLPK . . . . .	117
setDefaultIptParmGLPK . . . . .	118
setDefaultMIPParmGLPK . . . . .	118
setDefaultSmpParmGLPK . . . . .	119
setInteriorParmGLPK . . . . .	120
setMatColGLPK . . . . .	121
setMatRowGLPK . . . . .	122
setMIPParmGLPK . . . . .	123
setObjCoefGLPK . . . . .	124
setObjCoefsGLPK . . . . .	125
setObjDirGLPK . . . . .	126
setObjNameGLPK . . . . .	127
setProbNameGLPK . . . . .	128
setRhsZeroGLPK . . . . .	129
setRiiGLPK . . . . .	129
setRowBndGLPK . . . . .	130
setRowNameGLPK . . . . .	131
setRowsBndsGLPK . . . . .	132
setRowsNamesGLPK . . . . .	133
setRowStatGLPK . . . . .	134
setSimplexParmGLPK . . . . .	135
setSjjGLPK . . . . .	136
solveInteriorGLPK . . . . .	137
solveMIPGLPK . . . . .	138
solveSimplexExactGLPK . . . . .	139
solveSimplexGLPK . . . . .	140
sortMatrixGLPK . . . . .	141
status_codeGLPK . . . . .	141
stdBasisGLPK . . . . .	142
termOutGLPK . . . . .	143
unscaleProbGLPK . . . . .	144
versionGLPK . . . . .	144

warmUpGLPK . . . . .	145
writeIptGLPK . . . . .	146
writeLPGLPK . . . . .	147
writeMIPGLPK . . . . .	148
writeMPSGLPK . . . . .	149
writeProbGLPK . . . . .	150
writeSolGLPK . . . . .	151

<b>Index</b>	<b>152</b>
--------------	------------

---

glpkAPI-package	<i>R Interface to C API of GLPK</i>
-----------------	-------------------------------------

---

## Description

A low level interface to the GNU Linear Programming Kit (GLPK).

## Details

The package `glpkAPI` provides access to the callable library of the GNU Linear Programming Kit from within R.

## Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

## Examples

```
# load package
library(glpkAPI)

# preparing the model
lp <- initProbGLPK()

# model data
nrows <- 5
ncols <- 8

# constraint matrix
ne <- 14
ia <- c(1, 5, 1, 2, 2, 3, 1, 4, 1, 5, 3, 4, 1, 5)
ja <- c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 7, 8, 8)
ar <- c(3.0, 5.6, 1.0, 2.0, 1.1, 1.0, -2.0, 2.8,
```

```
        -1.0, 1.0, 1.0, -1.2, -1.0, 1.9)

# objective function
obj <- c(1, 0, 0, 0, 2, 0, 0, -1)

# upper and lower bounds of the rows
rlower <- c(2.5, -1000, 4, 1.8, 3)
rupper <- c(1000, 2.1, 4, 5, 15)

# upper and lower bounds of the columns
clower <- c(2.5, 0, 0, 0, 0.5, 0, 0, 0)
cupper <- c(1000, 4.1, 1, 1, 4, 1000, 1000, 4.3)

# direction of optimization
setObjDirGLPK(lp, GLP_MIN)

# add rows and columns
addRowsGLPK(lp, nrows)
addColsGLPK(lp, ncols)

setColsBndsObjCoefsGLPK(lp, c(1:ncols), clower, cupper, obj)
setRowsBndsGLPK(lp, c(1:nrows), rlower, rupper)

# load constraint matrix
loadMatrixGLPK(lp, ne, ia, ja, ar)

# solve lp problem
solveSimplexGLPK(lp)

# retrieve the results
getSolStatGLPK(lp)
getObjValGLPK(lp)
getColsPrimGLPK(lp)

# remove problem object
delProbGLPK(lp)
```

---

addColsGLPK

*Add Columns to a GLPK Problem Object*

---

## Description

Low level interface function to the GLPK function `glp_add_cols`. Consult the GLPK documentation for more detailed information.

## Usage

```
addColsGLPK(lp, ncols)
```

**Arguments**

- lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.
- ncols                The number of columns to add.

**Details**

Interface to the C function `addCols` which calls the GLPK function `glp_add_cols`.

**Value**

The ordinal number of the first new column added to the problem object is returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>

---

addRowsGLPK

*Add Rows to a GLPK Problem Object*

---

**Description**

Low level interface function to the GLPK function `glp_add_rows`. Consult the GLPK documentation for more detailed information.

**Usage**

```
addRowsGLPK(lp, nrows)
```

**Arguments**

- lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.
- nrows                The number of rows to add.

**Details**

Interface to the C function `addRows` which calls the GLPK function `glp_add_rows`.

**Value**

The ordinal number of the first new row added to the problem object is returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>

---

advBasisGLPK

*Construct Advanced Initial LP Basis*

---

**Description**

Low level interface function to the GLPK function `glp_adv_basis`. Consult the GLPK documentation for more detailed information.

**Usage**

```
advBasisGLPK(lp)
```

**Arguments**

`lp` An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `advBasis` which calls the GLPK function `glp_adv_basis`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>



---

bfExistsGLPK	<i>Check if the basis factorization exists</i>
--------------	--

---

### Description

Low level interface function to the GLPK function `glp_bf_exists`. Consult the GLPK documentation for more detailed information.

### Usage

```
bfExistsGLPK(lp)
```

### Arguments

`lp` An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `bfExists` which calls the GLPK function `glp_bf_exists`.

### Value

Returns non-zero if the basis factorization for the specified problem object exists. Otherwise the routine returns zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

bfUpdatedGLPK	<i>Check if the basis factorization has been updated</i>
---------------	--

---

### Description

Low level interface function to the GLPK function `glp_bf_updated`. Consult the GLPK documentation for more detailed information.

### Usage

```
bfUpdatedGLPK(lp)
```

### Arguments

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `bfUpdated` which calls the GLPK function `glp_bf_updated`.

### Value

Returns non-zero if the basis factorization has been updated at least once. Otherwise the routine returns zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

`checkDupGLPK`*Check for Duplicate Elements in Sparse Matrix*

---

**Description**

Low level interface function to the GLPK function `glp_check_dup`. Consult the GLPK documentation for more detailed information.

**Usage**

```
checkDupGLPK(m, n, ne, ia, ja)
```

**Arguments**

<code>m</code>	Number of rows in the matrix.
<code>n</code>	Number of columns in the matrix.
<code>ne</code>	Number of non-zero elements in the matrix.
<code>ia</code>	Row indices of the non-zero elements.
<code>ja</code>	Column indices of the non-zero elements.

**Details**

Interface to the C function `checkDup` which calls the GLPK function `glp_check_dup`.

**Value**

Returns one of the following values:

<code>0</code>	No duplicate elements.
<code>-k</code>	Indices <code>ia[k]</code> or <code>ja[k]</code> are out of range.
<code>+k</code>	Element <code>(ia[k], ja[k])</code> is duplicate.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>

---

`copyProbGLPK`*Copy problem object content*

---

**Description**

Low level interface function to the GLPK function `glp_copy_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
copyProbGLPK(lp, clp, name = GLP_OFF)
```

**Arguments**

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>clp</code>	A pointer to a GLPK problem object (destination).
<code>name</code>	If set to <code>GLP_ON</code> , the routine copies all symbolic names; otherwise ( <code>GLP_OFF</code> ) not.

**Details**

Interface to the C function `copyProb` which calls the GLPK function `glp_copy_prob`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section 'enable/disable flag'.

---

cpxBasisGLPK	<i>Construct Bixby's initial LP basis</i>
--------------	---

---

**Description**

Low level interface function to the GLPK function `glp_cpx_basis`. Consult the GLPK documentation for more detailed information.

**Usage**

```
cpxBasisGLPK(lp)
```

**Arguments**

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `cpxBasis` which calls the GLPK function `glp_cpx_basis`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

createIndexGLPK	<i>Create the Name Index</i>
-----------------	------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_create_index`. Consult the GLPK documentation for more detailed information.

**Usage**

```
createIndexGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `createIndex` which calls the GLPK function `glp_create_index`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

delColsGLPK

*Delete Columns from Problem Object*

---

**Description**

Low level interface function to the GLPK function `glp_del_cols`. Consult the GLPK documentation for more detailed information.

**Usage**

```
delColsGLPK(lp, ncols, j)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

ncols                Number of columns to delete.

j                    Ordinal numbers of columns to delete.

**Details**

Interface to the C function `delCols` which calls the GLPK function `glp_del_cols`.

**Value**

NULL

**Author(s)**

Gabriel Gelius-Dietrich &lt;geliudie@uni-duesseldorf.de&gt;

Maintainer: Mayo Roettger &lt;mayo.roettger@hhu.de&gt;

**References**Based on the package **glpk** by Lopaka Lee.The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

deleteIndexGLPK	<i>Delete the Name Index</i>
-----------------	------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_delete_index`. Consult the GLPK documentation for more detailed information.

**Usage**

```
deleteIndexGLPK(lp)
```

**Arguments**

`lp` An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `deleteIndex` which calls the GLPK function `glp_delete_index`.

**Value**

NULL

**Author(s)**

Gabriel Gelius-Dietrich &lt;geliudie@uni-duesseldorf.de&gt;

Maintainer: Mayo Roettger &lt;mayo.roettger@hhu.de&gt;

**References**Based on the package **glpk** by Lopaka Lee.The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

delProbGLPK                      *Delete Problem Object*

---

**Description**

Low level interface function to the GLPK function `glp_delete_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
delProbGLPK(lp)
```

**Arguments**

`lp`                      An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `delProb` which calls the GLPK function `glp_delete_prob`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

delRowsGLPK                      *Delete Rows from Problem Object*

---

**Description**

Low level interface function to the GLPK function `glp_del_rows`. Consult the GLPK documentation for more detailed information.

**Usage**

```
delRowsGLPK(lp, nrows, i)
```



**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
nrows	Number of rows to delete.
i	Ordinal numbers of rows to delete.

**Details**

Interface to the C function `delRows` which calls the GLPK function `glp_del_rows`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

eraseProbGLPK

*Erase problem object content*

---

**Description**

Low level interface function to the GLPK function `glp_erase_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
eraseProbGLPK(lp)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
----	---

**Details**

Interface to the C function `eraseProb` which calls the GLPK function `glp_erase_prob`.

**Value**

NULL

**Author(s)**

Gabriel Gelius-Dietrich &lt;geliudie@uni-duesseldorf.de&gt;

Maintainer: Mayo Roettger &lt;mayo.roettger@hhu.de&gt;

**References**Based on the package **glpk** by Lopaka Lee.The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

factorizeGLPK*Compute the basis factorization*

---

**Description**

Low level interface function to the GLPK function `glp_factorize`. Consult the GLPK documentation for more detailed information.

**Usage**`factorizeGLPK(lp)`**Arguments**

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `factorize` which calls the GLPK function `glp_factorize`.

**Value**

Returns zero if the basis factorization has been successfully computed. Otherwise the routine returns non-zero.

**Author(s)**

Gabriel Gelius-Dietrich &lt;geliudie@uni-duesseldorf.de&gt;

Maintainer: Mayo Roettger &lt;mayo.roettger@hhu.de&gt;

**References**Based on the package **glpk** by Lopaka Lee.The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘return codes’.

---

findColGLPK	<i>Find Column by its Name</i>
-------------	--------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_find_col`. Consult the GLPK documentation for more detailed information.

**Usage**

```
findColGLPK(lp, cname)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
cname	A column name.

**Details**

Interface to the C function `findCol` which calls the GLPK function `glp_find_column`.

**Value**

Returns the ordinal number of a column, which is assigned the specified `cname`.

**Note**

Before calling `findColGLPK` for the first time on a problem object `lp`, an index has to be created via a call to [createIndexGLPK](#).

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

`findRowGLPK`*Find Row by its Name*

---

**Description**

Low level interface function to the GLPK function `glp_find_row`. Consult the GLPK documentation for more detailed information.

**Usage**

```
findRowGLPK(lp, rname)
```

**Arguments**

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>rname</code>	A row name.

**Details**

Interface to the C function `findRow` which calls the GLPK function `glp_find_row`.

**Value**

Returns the ordinal number of a row, which is assigned the specified name.

**Note**

Before calling `findRowGLPK` for the first time on a problem object `lp`, an index has to be created via a call to `createIndexGLPK`.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getBfcpGLPK	<i>Retrieve Basis Factorization Control parameters</i>
-------------	--

---

### Description

Returns the names and values of members in the structure `glp_bfcp`. Consult the GLPK documentation for more detailed information.

### Usage

```
getBfcpGLPK(lp)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
-----------------	--

### Details

Interface to the C function `getBfcp`.

### Value

The function returns a list.

integer	The names and corresponding values of all integer control parameters in <code>glp_bfcp</code> .
double	The names and corresponding values of all double control parameters in <code>glp_bfcp</code> .

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section ‘Control Parameters’.

---

getBheadGLPK	<i>Retrieve Basis Header Information</i>
--------------	--

---

### Description

Low level interface function to the GLPK function `glp_get_bhead`. Consult the GLPK documentation for more detailed information.

### Usage

```
getBheadGLPK(lp, k)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
k	Index of the basic variable.

### Details

Interface to the C function `getBhead` which calls the GLPK function `glp_get_bhead`.

### Value

Index of the auxiliary/structural variable.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getCbindGLPK	<i>Retrieve Column Index in the Basis Header</i>
--------------	--

---

### Description

Low level interface function to the GLPK function `glp_get_col_bind`. Consult the GLPK documentation for more detailed information.

### Usage

```
getCbindGLPK(lp, j)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Structural variable <code>j</code> .

### Details

Interface to the C function `getCbind` which calls the GLPK function `glp_get_col_bind`.

### Value

Index of the basic variable.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColDualGLPK	<i>Retrieve Column Dual Value</i>
----------------	-----------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_col_dual`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColDualGLPK(lp, j)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .

### Details

Interface to the C function `getColDual` which calls the GLPK function `glp_get_col_dual`.

### Value

Column dual value

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.



---

getColDualIptGLPK      *Retrieve Column Dual Value*

---

### Description

Low level interface function to the GLPK function `glp_ipt_col_dual`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColDualIptGLPK(lp, j)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .

### Details

Interface to the C function `getColDualIpt` which calls the GLPK function `glp_ipt_col_dual`.

### Value

Column dual value

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColKindGLPK	<i>Retrieve Column Kind</i>
----------------	-----------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_col_kind`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColKindGLPK(lp, j)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

### Details

Interface to the C function `getColKind` which calls the GLPK function `glp_get_col_kind`.

### Value

Column Kind

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColLowBndGLPK	<i>Retrieve Column Lower Bound</i>
------------------	------------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_col_lb`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColLowBndGLPK(lp, j)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .

### Details

Interface to the C function `getColLowBnd` which calls the GLPK function `glp_get_col_lb`.

### Value

The lower bound of the `j`-th column (the corresponding structural variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColNameGLPK	<i>Retrieve Column Name</i>
----------------	-----------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_col_name`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColNameGLPK(lp, j)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

### Details

Interface to the C function `getColName` which calls the GLPK function `glp_get_col_name`.

### Value

The assigned name of the j-th column is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColPrimGLPK	<i>Retrieve Column Primal Value</i>
----------------	-------------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_col_prim`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColPrimGLPK(lp, j)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

### Details

Interface to the C function `getColPrim` which calls the GLPK function `glp_get_col_prim`.

### Value

The primal value of the j-th column (the corresponding structural variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColPrimIptGLPK      *Retrieve Column Primal Value*

---

### Description

Low level interface function to the GLPK function `glp_ipt_col_prim`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColPrimIptGLPK(lp, j)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .

### Details

Interface to the C function `getColPrimIpt` which calls the GLPK function `glp_ipt_col_prim`.

### Value

The primal value of the `j`-th column (the corresponding structural variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColsDualGLPK      *Retrieve Column Dual Value of all Columns*

---

**Description**

This is an advanced version of [getColDualGLPK](#).

**Usage**

```
getColsDualGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getColsDual` which calls the GLPK function `glp_get_col_dual`.

**Value**

The column dual values of all columns (structural variables) are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColsDualIptGLPK      *Retrieve Column Dual Value of all Columns*

---

**Description**

This is an advanced version of [getColDualIptGLPK](#).

**Usage**

```
getColsDualIptGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getColDualIpt` which calls the GLPK function `glp_ipt_col_dual`.

**Value**

The column dual values of all columns are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColsKindGLPK            *Retrieve Column Kind*

---

**Description**

This is an advanced version of `getColKindGLPK`.

**Usage**

```
getColsKindGLPK(lp, j)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

j                     Vector of column numbers.

**Details**

Interface to the C function `getColsKind` which calls the GLPK function `glp_get_col_ub`.

**Value**

The column kinds of all specified columns (j) are returned.



**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColsLowBndsGLPK      *Retrieve Lower Bounds of Specified Columns*

---

**Description**

This is an advanced version of [getColLowBndGLPK](#). Here, *j* can be an integer vector.

**Usage**

```
getColsLowBndsGLPK(lp, j)
```

**Arguments**

*lp*                    An object of class "[glpkPtr](#)" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

*j*                     Vector of column numbers.

**Details**

Interface to the C function `getColsLowBnds` which calls the GLPK function `glp_get_col_lb`.

**Value**

The lower bounds of all specified columns (*j*) (the corresponding structural variables) are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

getColsPrimGLPK      *Retrieve all Column Primal Values*

---

**Description**

This is an advanced version of [getColPrimGLPK](#).

**Usage**

```
getColsPrimGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getColsPrim` which calls the GLPK functions `glp_get_col_prim` and `glp_get_num_cols`.

**Value**

Returns all values of the structural variables as a numeric vector.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColsPrimIptGLPK      *Retrieve all Column Primal Values*

---

**Description**

This is an advanced version of [getColPrimGLPK](#).

**Usage**

```
getColsPrimIptGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getColsPrimIpt` which calls the GLPK functions `glp_igt_col_prim` and `glp_get_num_cols`.

**Value**

Returns all values of the structural variables as a numeric vector.

**Author(s)**

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>  
Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.  
The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColsStatGLPK                    *Retrieve Column Status of all Columns*

---

**Description**

This is an advanced version of [getColStatGLPK](#).

**Usage**

```
getColsStatGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getColsStat` which calls the GLPK function `glp_get_col_stat`.

**Value**

The column status of all columns are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColStatGLPK	<i>Retrieve Column Status</i>
----------------	-------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_get_col_stat`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getColStatGLPK(lp, j)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

**Details**

Interface to the C function `getColStat` which calls the GLPK function `glp_get_col_stat`.

**Value**

Column status

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section 'LP/MIP problem object'.

---

getColsUppBndsGLPK      *Retrieve Upper Bounds of Specified Columns*

---

### Description

This is an advanced version of [getColUppBndGLPK](#). Here, *j* can be an integer vector.

### Usage

```
getColsUppBndsGLPK(lp, j)
```

### Arguments

<i>lp</i>	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>j</i>	Vector of column numbers.

### Details

Interface to the C function `getColsUppBnds` which calls the GLPK function `glp_get_col_ub`.

### Value

The upper bounds of all specified columns (*j*) (the corresponding structural variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getColTypeGLPK      *Retrieve Column Type*

---

### Description

Low level interface function to the GLPK function `glp_get_col_type`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColTypeGLPK(lp, j)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

### Details

Interface to the C function `getColType` which calls the GLPK function `glp_get_col_type`.

### Value

The type of the j-th column (the corresponding structural variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'LP/MIP problem object'.

---

getColUppBndGLPK	<i>Retrieve Column Upper Bound</i>
------------------	------------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_col_ub`. Consult the GLPK documentation for more detailed information.

### Usage

```
getColUppBndGLPK(lp, j)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .

### Details

Interface to the C function `getColUppBnd` which calls the GLPK function `glp_get_col_ub`.

### Value

The upper bound of the `j`-th column (the corresponding structural variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getDualStatGLPK      *Retrieve Status of Dual Basic Solution*

---

### Description

Low level interface function to the GLPK function `glp_get_dual_stat`. Consult the GLPK documentation for more detailed information.

### Usage

```
getDualStatGLPK(lp)
```

### Arguments

`lp`                    An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `getDualStat` which calls the GLPK function `glp_get_dual_stat`.

### Value

Status of dual basic solution

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'LP/MIP problem object'.



---

getInteriorParmGLPK    *Retrives the Control Parameters for the Interior-point Method.*

---

## Description

Returns the names and values of members in the structure `glp_iptcp`. Consult the GLPK documentation for more detailed information.

## Usage

```
getInteriorParmGLPK()
```

## Details

Interface to the C function `getInteriorParm`.

## Value

The function returns a list.

integer            The names and corresponding values of all integer control parameters in `glp_iptcp`.

## Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

## See Also

[glpkConstants](#), section ‘Control Parameters’.

---

getMatColGLPK                      *Retrieves Column j of the Constraint Matrix.*

---

### Description

Low level interface function to the GLPK function `glp_get_mat_col`. Consult the GLPK documentation for more detailed information.

### Usage

```
getMatColGLPK(lp, j)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

### Details

Interface to the C function `getMatCol` which calls the GLPK functions `glp_get_num_rows` and `glp_get_mat_col`.

### Value

Returns NULL or a list containing the non zero elements of column j:

nnz	number of non zero elements in column j
index	row indices of the non zero elements in column j
value	numerical values of the non zero elements in column j

### Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>  
Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getMatRowGLPK	<i>Retrieves Row i of the Constraint Matrix.</i>
---------------	--

---

### Description

Low level interface function to the GLPK function `glp_get_mat_row`. Consult the GLPK documentation for more detailed information.

### Usage

```
getMatRowGLPK(lp, i)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
i	Row number i.

### Details

Interface to the C function `getMatRow` which calls the GLPK functions `glp_get_num_cols` and `glp_get_mat_row`.

### Value

Returns NULL or a list containing the non zero elements of row i:

nnz	number of non zero elements in row i
index	column indices of the non zero elements in row i
value	numerical values of the non zero elements in row i

### Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>  
Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getMIPParamGLPK	<i>Retrives the Control Parameters for MIP.</i>
-----------------	---

---

### Description

Returns the names and values of members in the structure `glp_iocp`. Consult the GLPK documentation for more detailed information.

### Usage

```
getMIPParamGLPK()
```

### Details

Interface to the C function `getMIPParam`.

### Value

The function returns a list.

integer            The names and corresponding values of all integer control parameters in `glp_iocp`.

double            The names and corresponding values of all double control parameters in `glp_iocp`.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section ‘Control Parameters’.

---

getNumBinGLPK	<i>Retrieve Number of Binary Columns</i>
---------------	--

---

**Description**

Low level interface function to the GLPK function `glp_get_num_bin`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getNumBinGLPK(lp)
```

**Arguments**

`lp` An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getNumBin` which calls the GLPK function `glp_get_num_bin`.

**Value**

Number of binary columns.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getNumColsGLPK	<i>Retrieve Number of Columns</i>
----------------	-----------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_get_num_cols`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getNumColsGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getNumCols` which calls the GLPK function `glp_get_num_cols`.

**Value**

Returns the current number of columns in the specified problem object.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getNumIntGLPK

*Retrieve Number of Integer Columns*

---

**Description**

Low level interface function to the GLPK function `glp_get_num_int`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getNumIntGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getNumInt` which calls the GLPK function `glp_get_num_int`.

**Value**

Number of integer columns.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getNumNnzGLPK

*Retrieve the Number of Constraint Coefficients*

---

**Description**

Low level interface function to the GLPK function `glp_get_num_nz`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getNumNnzGLPK(lp)
```

**Arguments**

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getNumNnz` which calls the GLPK function `glp_get_num_nz`.

**Value**

Returns the number of non-zero elements in the constraint matrix of the specified problem object.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getNumRowsGLPK      *Retrieve Number of Rows*

---

**Description**

Low level interface function to the GLPK function `glp_get_num_rows`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getNumRowsGLPK(lp)
```

**Arguments**

`lp`      An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getNumRows` which calls the GLPK function `glp_get_num_rows`.

**Value**

Returns the current number of rows in the specified problem object.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getObjCoefGLPK      *Retrieve Objective Coefficient or Constant Term*

---

**Description**

Low level interface function to the GLPK function `glp_get_obj_coef`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getObjCoefGLPK(lp, j)
```



**Arguments**

- lp            An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.
- j             Column number j.

**Details**

Interface to the C function `getObjCoef` which calls the GLPK function `glp_get_obj_coef`.

**Value**

The objective coefficient at the j-th column (the corresponding structural variable) is returned. If j is 0, the constant term "shift" of the objective function is returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getObjCoefsGLPK	<i>Retrieve Objective Coefficients at Specified Columns and/or Constant Term</i>
-----------------	--

---

**Description**

This is an advanced version of `getObjCoefGLPK`. Here, j can be an integer vector.

**Usage**

```
getObjCoefsGLPK(lp, j)
```

**Arguments**

- lp            An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.
- j             Vector of column numbers.

**Details**

Interface to the C function `getObjCoef` which calls the GLPK function `glp_get_obj_coef`.

**Value**

The objective coefficient at all specified columns (*j*) (the corresponding structural variable) is returned. If *j* is 0, the constant term “shift” of the objective function is returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getObjDirGLPK

*Retrieve Optimization Direction Flag*

---

**Description**

Low level interface function to the GLPK function `glp_get_obj_dir`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getObjDirGLPK(lp)
```

**Arguments**

`lp` An object of class “`glpkPtr`” as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getObjDir` which calls the GLPK function `glp_get_obj_dir`.

**Value**

Returns the optimization direction flag.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section 'LP/MIP problem object'.

---

getObjNameGLPK

*Retrieve Objective Function Name*

---

**Description**

Low level interface function to the GLPK function `glp_get_obj_name`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getObjNameGLPK(lp)
```

**Arguments**

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getObjName` which calls the GLPK function `glp_get_obj_name`.

**Value**

The assigned name of the objective function is returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getObjValGLPK            *Retrieve Objective Value*

---

**Description**

Low level interface function to the GLPK function `glp_get_obj_val`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getObjValGLPK(lp)
```

**Arguments**

`lp`            An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getObjVal` which calls the GLPK function `glp_get_obj_val`.

**Value**

Returns the current value of the objective function.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getObjValIptGLPK            *Retrieve Objective Value*

---

**Description**

Low level interface function to the GLPK function `glp_ipt_obj_val`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getObjValIptGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getObjValIpt` which calls the GLPK function `glp_ipt_obj_val`.

**Value**

Returns the current value of the objective function.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getPrimStatGLPK                    *Retrieve Status of Primal Basic Solution*

---

**Description**

Low level interface function to the GLPK function `glp_get_prim_stat`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getPrimStatGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getPrimStat` which calls the GLPK function `glp_get_prim_stat`.

**Value**

Status of primal basic solution

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘LP/MIP problem object’.

---

getProbNameGLPK	<i>Retrieve Problem Name</i>
-----------------	------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_get_prob_name`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getProbNameGLPK(lp)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
----	--

**Details**

Interface to the C function `getProbName` which calls the GLPK function `glp_get_prob_name`.

**Value**

The assigned name of the problem is returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRbindGLPK	<i>Retrieve Row Index in the Basis Header</i>
--------------	---

---

### Description

Low level interface function to the GLPK function `glp_get_row_bind`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRbindGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Auxiliary variable <code>i</code> .

### Details

Interface to the C function `getRbind` which calls the GLPK function `glp_get_row_bind`.

### Value

Index of the basic variable.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRiiGLPK	<i>Retrieve row scale factor</i>
------------	----------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_rii`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRiiGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRii` which calls the GLPK function `glp_get_rii`.

### Value

Returns the current scale factor `$r_ii` for row `i` of the specified problem object.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.



---

getRowDualGLPK	<i>Retrieve Row Dual Value</i>
----------------	--------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_row_dual`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowDualGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowDual` which calls the GLPK function `glp_get_row_dual`.

### Value

Row dual value

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowDualIptGLPK      *Retrieve Row Dual Value*

---

### Description

Low level interface function to the GLPK function `glp_ipt_row_dual`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowDualIptGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowDualIpt` which calls the GLPK function `glp_ipt_row_dual`.

### Value

Row dual value

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowLowBndGLPK	<i>Retrieve Row Lower Bound</i>
------------------	---------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_row_lb`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowLowBndGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowLowBnd` which calls the GLPK function `glp_get_row_lb`.

### Value

The lower bound of the `i`-th row (the corresponding auxiliary variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowNameGLPK	<i>Retrieve Row Name</i>
----------------	--------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_row_name`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowNameGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowName` which calls the GLPK function `glp_get_row_name`.

### Value

The assigned name of the `i`-th row is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowPrimGLPK	<i>Retrieve Row Primal Value</i>
----------------	----------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_row_prim`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowPrimGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowPrim` which calls the GLPK function `glp_get_row_prim`.

### Value

Row primal value

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowPrimIptGLPK      *Retrieve Row Primal Value*

---

### Description

Low level interface function to the GLPK function `glp_ipt_row_prim`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowPrimIptGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowPrimIpt` which calls the GLPK function `glp_ipt_row_prim`.

### Value

Row primal value

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowsDualGLPK      *Retrieve Row Dual Values of all Rows*

---

**Description**

This is an advanced version of [getRowDualGLPK](#).

**Usage**

```
getRowsDualGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getRowsDual` which calls the GLPK function `glp_get_row_stat`.

**Value**

The row dual values of all rows are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowsDualIptGLPK      *Retrieve Row Dual Value of all Rows*

---

**Description**

This is an advanced version of [getRowDualIptGLPK](#).

**Usage**

```
getRowsDualIptGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getRowsDualIpt` which calls the GLPK function `glp_ipt_row_dual`.

**Value**

The row dual values of all rows are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowsLowBndsGLPK      *Retrieve Lower Bounds of Specified Rows*

---

**Description**

This is an advanced version of `getRowLowBndGLPK`. Here, `i` can be an integer vector.

**Usage**

```
getRowsLowBndsGLPK(lp, i)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

i                     Vector of row numbers.

**Details**

Interface to the C function `getRowsLowBnds` which calls the GLPK function `glp_get_row_lb`.

**Value**

The lower bounds of all specified columns (`i`) (the corresponding auxiliary variables) are returned.



**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowsPrimGLPK	<i>Retrieve Row Primal Value of all Rows</i>
-----------------	--

---

**Description**

This is an advanced version of [getRowPrimGLPK](#).

**Usage**

```
getRowsPrimGLPK(lp)
```

**Arguments**

lp                   An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getRowsPrim` which calls the GLPK function `glp_get_row_prim`.

**Value**

The row primal values for all rows are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

getRowsPrimIptGLPK      *Retrieve Row Primal Value of all Rows*

---

**Description**

This is an advanced version of [getRowPrimIptGLPK](#).

**Usage**

```
getRowsPrimIptGLPK(lp)
```

**Arguments**

lp                      An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getRowsPrimIpt` which calls the GLPK function `glp_apt_row_prim`.

**Value**

The row primal values of all rows are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowsStatGLPK      *Retrieve Row Status of all Rows*

---

**Description**

This is an advanced version of [getRowStatGLPK](#).

**Usage**

```
getRowsStatGLPK(lp)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `getRowStat` which calls the GLPK function `glp_get_row_stat`.

**Value**

The row status values of all rows are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowStatGLPK

*Retrieve Row Status*

---

**Description**

Low level interface function to the GLPK function `glp_get_row_stat`. Consult the GLPK documentation for more detailed information.

**Usage**

```
getRowStatGLPK(lp, i)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

i                     Row number i.

**Details**

Interface to the C function `getRowStat` which calls the GLPK function `glp_get_row_stat`.

**Value**

Row status

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘LP/MIP problem object’.

---

getRowsTypesGLPK

*Retrieve Types of Specified Constraints (Rows)*

---

**Description**

This is an advanced version of [getRowTypeGLPK](#). Here, *i* can be an integer vector.

**Usage**

```
getRowsTypesGLPK(lp, i)
```

**Arguments**

<i>lp</i>	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>i</i>	Vector of row numbers.

**Details**

Interface to the C function `getRowsTypes` which calls the GLPK function `glp_get_row_type`.

**Value**

A numeric vector of the same length as *i* giving the constraint type of the specified rows.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘type of auxiliary/structural variable’.

---

getRowsUppBndsGLPK      *Retrieve Upper Bounds of Specified Rows*

---

**Description**

This is an advanced version of [getRowUppBndGLPK](#). Here, *i* can be an integer vector.

**Usage**

```
getRowsUppBndsGLPK(lp, i)
```

**Arguments**

<i>lp</i>	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>i</i>	Vector of row numbers.

**Details**

Interface to the C function `getRowsUppBnds` which calls the GLPK function `glp_get_row_ub`.

**Value**

The upper bounds of all specified columns (*i*) (the corresponding auxiliary variables) are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getRowTypeGLPK      *Retrieve Row Type*

---

### Description

Low level interface function to the GLPK function `glp_get_row_type`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowTypeGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowType` which calls the GLPK function `glp_get_row_type`.

### Value

The type of the `i`-th row (the corresponding auxiliary variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'LP/MIP problem object'.

---

getRowUppBndGLPK	<i>Retrieve Row Upper Bound</i>
------------------	---------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_row_ub`. Consult the GLPK documentation for more detailed information.

### Usage

```
getRowUppBndGLPK(lp, i)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .

### Details

Interface to the C function `getRowUppBnd` which calls the GLPK function `glp_get_row_ub`.

### Value

The upper bound of the `i`-th row (the corresponding auxiliary variable) is returned.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getSimplexParmGLPK      *Retrives the Control Parameters for the Simplex Method.*

---

### Description

Returns the names and values of members in the structure `glp_smcp`. Consult the GLPK documentation for more detailed information.

### Usage

```
getSimplexParmGLPK()
```

### Details

Interface to the C function `getSimplexParm`.

### Value

The function returns a list.

integer            The names and corresponding values of all integer control parameters in `glp_smcp`.

double            The names and corresponding values of all double control parameters in `glp_smcp`.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section ‘Control Parameters’.



---

getSjjGLPK	<i>Retrieve column scale factor</i>
------------	-------------------------------------

---

### Description

Low level interface function to the GLPK function `glp_get_sjj`. Consult the GLPK documentation for more detailed information.

### Usage

```
getSjjGLPK(lp, j)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

### Details

Interface to the C function `getSjj` which calls the GLPK function `glp_get_sjj`.

### Value

Returns the current scale factor `$s_jj` for column j of the specified problem object.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

getSolStatGLPK      *Determine Generic Status of the Basic Solution*

---

### Description

Low level interface function to the GLPK function `glp_get_status`. Consult the GLPK documentation for more detailed information.

### Usage

```
getSolStatGLPK(lp)
```

### Arguments

`lp`                    An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `getSolStat` which calls the GLPK function `glp_get_status`.

### Value

Returns the generic status of the current basic solution for the specified problem object.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'LP/MIP problem object'.

---

getSolStatIptGLPK      *Determine Solution Status*

---

### Description

Low level interface function to the GLPK function `glp_ipt_status`. Consult the GLPK documentation for more detailed information.

### Usage

```
getSolStatIptGLPK(lp)
```

### Arguments

`lp`                    An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `getSolStatIpt` which calls the GLPK function `glp_ipt_status`.

### Value

Returns the generic status of the current basic solution for the specified problem object.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'LP/MIP problem object'.

---

getUnbndRayGLPK      *Determine Variable Causing Unboundedness*

---

### Description

Low level interface function to the GLPK function `glp_get_unbnd_ray`. Consult the GLPK documentation for more detailed information.

### Usage

```
getUnbndRayGLPK(lp)
```

### Arguments

`lp`                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `getUnbndRay` which calls the GLPK function `glp_get_unbnd_ray`.

### Value

Returns the number `k` of a variable, which causes primal or dual unboundedness.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**Description**

This is a list containing constants used by GLPK. Consult the glpk manual for more information, in particular for the control parameters.

**Control Parameters***Simplex*

MSG_LEV <- 101	Message level for terminal output (default: GLP_MSG_ALL).
METH <- 102	Simplex method option (default: GLP_PRIMAL).
PRICING <- 103	Pricing technique (default: GLP_PT_PSE).
R_TEST <- 104	Ratio test technique (default: GLP_RT_HAR).
IT_LIM <- 105	Simplex iteration limit (default: INT_MAX).
TM_LIM <- 106	Searching time limit, in milliseconds (default: INT_MAX).
OUT_FRQ <- 107	Output frequency, in iterations (default: 500).
OUT_DLY <- 108	Output delay, in milliseconds (default: 0).
PRESOLVE <- 109	LP presolver option (default: GLP_OFF).
TOL_BND <- 201	Tolerance used to check if the basic solution is primal feasible (default: 1e-7).
TOL_DJ <- 202	Tolerance used to check if the basic solution is dual feasible (default: 1e-7).
TOL_PIV <- 203	Tolerance used to choose eligible pivotal elements of the simplex table (default: 1e-10).
OBJ_LL <- 204	Lower limit of the objective function (default: -DBL_MAX).
OBJ_UL <- 205	Upper limit of the objective function (default: DBL_MAX).

The exact simplex method uses only the parameters IT\_LIM and TM\_LIM.

*Interior*

MSG_LEV <- 101	Message level for terminal output (default: GLP_MSG_ALL).
ORD_ALG <- 301	Ordering algorithm used prior to Cholesky factorization (default: GLP_ORD_AMD).

*MIP*

MSG_LEV <- 101	Message level for terminal output (default: GLP_MSG_ALL).
TM_LIM <- 106	Searching time limit, in milliseconds (default: INT_MAX).
OUT_FRQ <- 107	Output frequency, in iterations (default: 5000).
OUT_DLY <- 108	Output delay, in milliseconds (default: 10000).
PRESOLVE <- 109	MIP presolver option (default: GLP_OFF).
BR_TECH <- 601	Branching technique option (default: GLP_BR_DTH).
BT_TECH <- 602	Backtracking technique option (default: GLP_BT_BLB).
PP_TECH <- 603	Preprocessing technique option (default: GLP_PP_ALL).
FP_HEUR <- 604	Feasibility pump heuristic option (default: GLP_OFF).

GMI\_CUTS <- 605 Gomory's mixed integer cut option (default: GLP\_OFF).  
 MIR\_CUTS <- 606 Mixed integer rounding (MIR) cut option (default: GLP\_OFF).  
 COV\_CUTS <- 607 Mixed cover cut option (default: GLP\_OFF).  
 CLQ\_CUTS <- 608 Clique cut option (default: GLP\_OFF).  
 CB\_SIZE <- 609 The number of extra (up to 256) bytes allocated for each node of the branch-and-bound tree to store application data.  
 BINARIZE <- 610 LP presolver option (default: GLP\_OFF).  
 CB\_FUNC <- 651 Use a user defined callback routine glpkCallback which is written in the file 'glpkCallback.c'. This file is located in the directory 'src'.  
 TOL\_INT <- 701 Absolute tolerance used to check if optimal solution to the current LP relaxation is integer feasible (default: 1e-10).  
 TOL\_OBJ <- 702 Relative tolerance used to check if the objective value in optimal solution to the current LP relaxation is near optimal (default: 1e-5).  
 MIP\_GAP <- 703 The relative mip gap tolerance. If the relative mip gap for currently known best integer feasible solution falls below this tolerance, the solver will terminate. (default: 1e-5).

### *Basis Factorization*

TYPE <- 401 Basis factorization type (default: GLP\_BF\_FT).  
 LU\_SIZE <- 402 Initial size of the Sparse Vector Area (default: 0).  
 PIV\_LIM <- 403 computing LU-factorization of the basis matrix (default: 4).  
 SUHL <- 404 computing LU-factorization of the basis matrix (default: GLP\_ON).  
 NFS\_MAX <- 405 Maximal number of additional row-like factors (default: 100).  
 NRS\_MAX <- 406 Maximal number of additional rows and columns (default: 100).  
 RS\_SIZE <- 407 Initial size of the Sparse Vector Area (default: 0).  
 PIV\_TOL <- 501 Threshold pivoting (Markowitz) tolerance (default: 0.10).  
 EPS\_TOL <- 502 Epsilon tolerance (default: 1e-15).  
 MAX\_GRO <- 503 Maximal growth of elements of factor U (default: 1e+10).  
 UPD\_TOL <- 504 Update tolerance (default: 1e-6).

### **LP/MIP problem object**

#### *optimization direction flag*

GLP\_MIN <- 1 minimization  
 GLP\_MAX <- 2 maximization

#### *kind of structural variable*

GLP\_CV <- 1 continuous variable  
 GLP\_IV <- 2 integer variable  
 GLP\_BV <- 3 binary variable

#### *type of auxiliary/structural variable*

GLP\_FR <- 1 free variable  
 GLP\_LO <- 2 variable with lower bound  
 GLP\_UP <- 3 variable with upper bound

GLP\_DB <- 4 double-bounded variable  
 GLP\_FX <- 5 fixed variable

*status of auxiliary/structural variable*

GLP\_BS <- 1 basic variable  
 GLP\_NL <- 2 non-basic variable on lower bound  
 GLP\_NU <- 3 non-basic variable on upper bound  
 GLP\_NF <- 4 non-basic free variable  
 GLP\_NS <- 5 non-basic fixed variable

*scaling options*

GLP\_SF\_GM <- 0x01 perform geometric mean scaling  
 GLP\_SF\_EQ <- 0x10 perform equilibration scaling  
 GLP\_SF\_2N <- 0x20 round scale factors to power of two  
 GLP\_SF\_SKIP <- 0x40 skip if problem is well scaled  
 GLP\_SF\_AUTO <- 0x80 choose scaling options automatically

*solution indicator*

GLP\_SOL <- 1 basic solution  
 GLP\_IPT <- 2 interior-point solution  
 GLP\_MIP <- 3 mixed integer solution

*solution status*

GLP\_UNDEF <- 1 solution is undefined  
 GLP\_FEAS <- 2 solution is feasible  
 GLP\_INFEAS <- 3 solution is infeasible  
 GLP\_NOFEAS <- 4 no feasible solution exists  
 GLP\_OPT <- 5 solution is optimal  
 GLP\_UNBND <- 6 solution is unbounded

**basis factorization control parameters**

*type*

GLP\_BF\_FT <- 0x01 LUF + Forrest-Tomlin  
 GLP\_BF\_BG <- 0x02 LUF + Schur compl. + Bartels-Golub  
 GLP\_BF\_GR <- 0x03 LUF + Schur compl. + Givens rotation  
 GLP\_BF\_LUF <- 0x00 plain LU-factorization

GLP\_BF\_BTFF <- 0x10 block triangular LU-factorization

### simplex method control parameters

*msg\_lev* message level:

GLP\_MSG\_OFF <- 0 no output  
 GLP\_MSG\_ERR <- 1 warning and error messages only  
 GLP\_MSG\_ON <- 2 normal output  
 GLP\_MSG\_ALL <- 3 full output  
 GLP\_MSG\_DBG <- 4 debug output

*meth* simplex method option:

GLP\_PRIMAL <- 1 use primal simplex  
 GLP\_DUALP <- 2 use dual; if it fails, use primal  
 GLP\_DUAL <- 3 use dual simplex

*pricing* pricing technique:

GLP\_PT\_STD <- 0x11 standard (Dantzig rule)  
 GLP\_PT\_PSE <- 0x22 projected steepest edge

*r\_test* ratio test technique:

GLP\_RT\_STD <- 0x11 standard (textbook)  
 GLP\_RT\_HAR <- 0x22 two-pass Harris' ratio test

### interior-point solver control parameters

*ord\_alg* ordering algorithm:

GLP\_ORD\_NONE <- 0 natural (original) ordering  
 GLP\_ORD\_QMD <- 1 quotient minimum degree (QMD)  
 GLP\_ORD\_AMD <- 2 approx. minimum degree (AMD)  
 GLP\_ORD\_SYMAMD <- 3 approx. minimum degree (SYMAMD)

### integer optimizer control parameters

*br\_tech* branching technique:



GLP\_BR\_FFV <- 1 first fractional variable  
GLP\_BR\_LFV <- 2 last fractional variable  
GLP\_BR\_MFV <- 3 most fractional variable  
GLP\_BR\_DTH <- 4 heuristic by Driebeck and Tomlin  
GLP\_BR\_HPC <- 5 hybrid pseudocost

*bt\_tech* backtracking technique:

GLP\_BT\_DFS <- 1 depth first search  
GLP\_BT\_BFS <- 2 breadth first search  
GLP\_BT\_BLB <- 3 best local bound  
GLP\_BT\_BPH <- 4 best projection heuristic

*pp\_tech* preprocessing technique:

GLP\_PP\_NONE <- 0 disable preprocessing  
GLP\_PP\_ROOT <- 1 preprocessing only on root level  
GLP\_PP\_ALL <- 2 preprocessing on all levels

### **additional row attributes**

*the row origin flag*

GLP\_RF\_REG <- 0 regular constraint  
GLP\_RF\_LAZY <- 1 "lazy" constraint  
GLP\_RF\_CUT <- 2 cutting plane constraint

*the row class descriptor class*

GLP\_RF\_GMI <- 1 Gomory's mixed integer cut  
GLP\_RF\_MIR <- 2 mixed integer rounding cut  
GLP\_RF\_COV <- 3 mixed cover cut  
GLP\_RF\_CLQ <- 4 clique cut

### **enable/disable flag**

GLP\_ON <- 1 enable something  
GLP\_OFF <- 0 disable something

**reason codes**

GLP_IROWGEN <- 0x01	request for row generation
GLP_IBINGO <- 0x02	better integer solution found
GLP_IHEUR <- 0x03	request for heuristic solution
GLP_ICUTGEN <- 0x04	request for cut generation
GLP_IBRANCH <- 0x05	request for branching
GLP_ISELECT <- 0x06	request for subproblem selection
GLP_IPREPRO <- 0x07	request for preprocessing

**branch selection indicator**

GLP_NO_BRNCH <- 0	select no branch
GLP_DN_BRNCH <- 1	select down-branch
GLP_UP_BRNCH <- 2	select up-branch

**return codes**

GLP_EBADB <- 0x01	invalid basis
GLP_ESING <- 0x02	singular matrix
GLP_ECOND <- 0x03	ill-conditioned matrix
GLP_EBOUND <- 0x04	invalid bounds
GLP_EFAIL <- 0x05	solver failed
GLP_EOBJLL <- 0x06	objective lower limit reached
GLP_EOBJUL <- 0x07	objective upper limit reached
GLP_EITLIM <- 0x08	iteration limit exceeded
GLP_ETMLIM <- 0x09	time limit exceeded
GLP_ENOPFS <- 0x0A	no primal feasible solution
GLP_ENODFS <- 0x0B	no dual feasible solution
GLP_EROOT <- 0x0C	root LP optimum not provided
GLP_ESTOP <- 0x0D	search terminated by application
GLP_EMIPGAP <- 0x0E	relative mip gap tolerance reached
GLP_ENOFEAS <- 0x0F	no primal/dual feasible solution
GLP_ENOCVG <- 0x10	no convergence
GLP_EINSTAB <- 0x11	numerical instability
GLP_EDATA <- 0x12	invalid data
GLP_ERANGE <- 0x13	result out of range

**condition indicator**

```
GLP_KKT_PE <- 1  primal equalities
GLP_KKT_PB <- 2  primal bounds
GLP_KKT_DE <- 3  dual equalities
GLP_KKT_DB <- 4  dual bounds
GLP_KKT_CS <- 5  complementary slackness
```

**MPS file format**

```
GLP_MPS_DECK <- 1  fixed (ancient)
GLP_MPS_FILE <- 2  free (modern)
```

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[status\\_codeGLPK](#), [return\\_codeGLPK](#)

---

glpkPtr-class

*Class "glpkPtr"*

---

**Description**

Structure of the class "glpkPtr". Objects of that class are used to hold pointers to C structures used by GLPK.

**Objects from the Class**

Objects can be created by calls of the form

```
test <- initProbGLPK() or
```

```
test <- mplaAllocWkspGLPK().
```

**Slots**

`glpkPtrType`: Object of class "character" giving the pointer type.

`glpkPointer`: Object of class "externalptr" containing the pointer to a C structure.

**Methods**

**isGLPKpointer** signature(object = "glpkPtr"): returns TRUE if `glpkPointer(object)` is a pointer to a GLPK problem object, otherwise FALSE.

**isNULLpointerGLPK** signature(object = "glpkPtr"): returns TRUE if `glpkPointer(object)` is a NULL pointer, otherwise FALSE.

**isTRWKSpointer** signature(object = "glpkPtr"): returns TRUE if `glpkPointer(object)` is a pointer to a MathProg translator workspace, otherwise FALSE.

**glpkPointer** signature(object = "glpkPtr"): gets the `glpkPointer` slot.

**glpkPtrType** signature(object = "glpkPtr"): gets the `glpkPtrType` slot.

**glpkPtrType<-** signature(object = "glpkPtr"): sets the `glpkPtrType` slot.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[mplAllocWkspGLPK](#) and [initProbGLPK](#).

**Examples**

```
showClass("glpkPtr")
```

---

initProbGLPK

*Create a GLPK Problem Object*

---

**Description**

Low level interface function to the GLPK function `glp_create_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
initProbGLPK(ptrtype = "glpk_prob")
```

**Arguments**

ptrtype            A name for the pointer to a GLPK problem object.

**Details**

Interface to the C function `initProb` which calls the GLPK function `glp_create_prob`.

**Value**

An instance of class "`glpkPtr`".

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

"`glpkPtr`".

---

loadMatrixGLPK	<i>Load/Replace the Whole Constraint Matrix</i>
----------------	---

---

**Description**

Low level interface function to the GLPK function `glp_load_matrix`. Consult the GLPK documentation for more detailed information.

**Usage**

```
loadMatrixGLPK(lp, ne, ia, ja, ra)
```

**Arguments**

lp	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
ne	Number of non-zero elements.
ia	Row indices of the non-zero elements.
ja	Column indices of the non-zero elements.
ra	The numeric values of the constraint coefficients.

**Details**

Interface to the C function `loadMatrix` which calls the GLPK function `glp_load_matrix`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

mipColsValGLPK

*Retrieve Column Value of all Columns*

---

**Description**

This is an advanced version of [mipColValGLPK](#).

**Usage**

```
mipColsValGLPK(lp)
```

**Arguments**

`lp` An object of class "`glpkPtr`" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `mipColsVal` which calls the GLPK function `glp_mip_col_val`.

**Value**

The column values of all columns are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

mipColValGLPK

*Retrieve Column Value*

---

## Description

Low level interface function to the GLPK function `glp_mip_col_val`. Consult the GLPK documentation for more detailed information.

## Usage

```
mipColValGLPK(lp, j)
```

## Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.

## Details

Interface to the C function `mipColVal` which calls the GLPK function `glp_mip_col_val`.

## Value

Column value of column j.

## Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

mipObjValGLPK	<i>Retrieve Objective Value</i>
---------------	---------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_mip_obj_val`. Consult the GLPK documentation for more detailed information.

**Usage**

```
mipObjValGLPK(lp)
```

**Arguments**

`lp` An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `mipObjVal` which calls the GLPK function `glp_mip_obj_val`.

**Value**

Objective value.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

mipRowsValGLPK	<i>Retrieve Row Value of all Rows</i>
----------------	---------------------------------------

---

**Description**

This is an advanced version of `mipRowValGLPK`.

**Usage**

```
mipRowsValGLPK(lp)
```



**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `mipRowsVal` which calls the GLPK function `glp_mip_row_val`.

**Value**

The row values of all rows are returned.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

mipRowValGLPK

*Retrieve Row Value*

---

**Description**

Low level interface function to the GLPK function `glp_mip_row_val`. Consult the GLPK documentation for more detailed information.

**Usage**

```
mipRowValGLPK(lp, i)
```

**Arguments**

lp                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

i                     Row number i.

**Details**

Interface to the C function `mipRowVal` which calls the GLPK function `glp_mip_row_val`.

**Value**

Row value of row i.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

mipStatusGLPK

*Determine Status of MIP Solution*

---

**Description**

Low level interface function to the GLPK function `glp_mip_status`. Consult the GLPK documentation for more detailed information.

**Usage**

```
mipStatusGLPK(lp)
```

**Arguments**

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `mipStatus` which calls the GLPK function `glp_mip_status`.

**Value**

Status of MIP Solution.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

mplAllocWkspGLPK	<i>Allocate Translator Workspace</i>
------------------	--------------------------------------

---

### Description

Low level interface function to the GLPK function `glp_mpl_alloc_wksp`. Consult the GLPK documentation for more detailed information.

### Usage

```
mplAllocWkspGLPK(ptrtype = "tr_wksp")
```

### Arguments

`ptrtype`            A name for the pointer to a translator workspace.

### Details

Interface to the C function `mplAllocWksp` which calls the GLPK function `glp_mpl_alloc_wksp`.

### Value

An instance of class "`glpkPtr`".

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

"`glpkPtr`".

---

mplBuildProbGLPK      *Build Problem Instance From Model*

---

### Description

Low level interface function to the GLPK function `glp_mpl_build_prob`. Consult the GLPK documentation for more detailed information.

### Usage

```
mplBuildProbGLPK(wk, lp)
```

### Arguments

wk	An object of class "glpkPtr" as returned by <code>mplAllocWkspGLPK</code> . This is basically a pointer to a GLPK translocator workspace.
lp	A pointer to a GLPK problem object.

### Details

Interface to the C function `mplBuildProb` which calls the GLPK function `glp_mpl_build_prob`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`mplAllocWkspGLPK`, `mplFreeWkspGLPK`, `mplGenerateGLPK`, `mplPostsolveGLPK`, `mplReadDataGLPK` and `mplReadModelGLPK`.

---

mplFreeWkspGLPK      *Free Translator Workspace*

---

### Description

Low level interface function to the GLPK function `glp_mpl_free_wksp`. Consult the GLPK documentation for more detailed information.

### Usage

```
mplFreeWkspGLPK(wk)
```

### Arguments

`wk`                    An object of class "glpkPtr" as returned by `mplAllocWkspGLPK`. This is basically a pointer to a GLPK translocator workspace.

### Details

Interface to the C function `mplFreeWksp` which calls the GLPK function `glp_mpl_free_wksp`.

### Value

NULL

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`mplAllocWkspGLPK`, `mplBuildProbGLPK`, `mplGenerateGLPK`, `mplPostsolveGLPK`, `mplReadDataGLPK` and `mplReadModelGLPK`.

---

mplGenerateGLPK      *Generate the Model*

---

### Description

Low level interface function to the GLPK function `glp_mpl_generate`. Consult the GLPK documentation for more detailed information.

### Usage

```
mplGenerateGLPK(wk, fname = NULL)
```

### Arguments

`wk`                    An object of class "`glpkPtr`" as returned by `mplAllocWkspGLPK`. This is basically a pointer to a GLPK translocator workspace.

`fname`                The name of the text file to be written out.

### Details

Interface to the C function `mplGenerate` which calls the GLPK function `glp_mpl_generate`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`mplAllocWkspGLPK`, `mplBuildProbGLPK`, `mplFreeWkspGLPK`, `mplPostsolveGLPK`, `mplReadDataGLPK` and `mplReadModelGLPK`.

---

mplPostsolveGLPK      *Postsolve Model*

---

### Description

Low level interface function to the GLPK function `glp_mpl_postsolve`. Consult the GLPK documentation for more detailed information.

### Usage

```
mplPostsolveGLPK(wk, lp, sol)
```

### Arguments

<code>wk</code>	An object of class " <code>glpkPtr</code> " as returned by <code>mplAllocWspGLPK</code> . This is basically a pointer to a GLPK translocator workspace.
<code>lp</code>	A pointer to a GLPK problem object.
<code>sol</code>	Type of solution to be copied to the translator workspace, for possible values, see <code>glpkConstants</code> , section 'LP/MIP problem object'.

### Details

Interface to the C function `mplPostsolve` which calls the GLPK function `glp_mpl_postsolve`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`mplAllocWspGLPK`, `mplBuildProbGLPK`, `mplFreeWspGLPK`, `mplGenerateGLPK`, `mplReadDataGLPK` and `mplReadModelGLPK`.

---

`mplReadDataGLPK`*Read and Translate Data Section*

---

### Description

Low level interface function to the GLPK function `glp_mpl_read_data`. Consult the GLPK documentation for more detailed information.

### Usage

```
mplReadDataGLPK(wk, fname)
```

### Arguments

<code>wk</code>	An object of class " <code>glpkPtr</code> " as returned by <code>mplAllocWkspGLPK</code> . This is basically a pointer to a GLPK translocator workspace.
<code>fname</code>	The name of the data file to be read in.

### Details

Interface to the C function `mplReadData` which calls the GLPK function `glp_mpl_read_data`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`mplAllocWkspGLPK`, `mplBuildProbGLPK`, `mplFreeWkspGLPK`, `mplGenerateGLPK`, `mplPostsolveGLPK` and `mplReadModelGLPK`.



---

`mplReadModelGLPK`      *Read and Translate Model Section*

---

### Description

Low level interface function to the GLPK function `glp_mpl_read_model`. Consult the GLPK documentation for more detailed information.

### Usage

```
mplReadModelGLPK(wk, fname, skip)
```

### Arguments

<code>wk</code>	An object of class " <code>glpPtr</code> " as returned by <code>mplAllocWkspGLPK</code> . This is basically a pointer to a GLPK translocator workspace.
<code>fname</code>	The name of the model file to be read in.
<code>skip</code>	Flag, how to treat the data section.

### Details

Interface to the C function `mplReadModel` which calls the GLPK function `glp_mpl_read_model`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`mplAllocWkspGLPK`, `mplBuildProbGLPK`, `mplFreeWkspGLPK`, `mplGenerateGLPK`, `mplPostsolveGLPK` and `mplReadDataGLPK`.

---

printIptGLPK

*Write Interior-Point Solution in Printable Format*

---

### Description

Low level interface function to the GLPK function `glp_print_ipr`. Consult the GLPK documentation for more detailed information.

### Usage

```
printIptGLPK(lp, fname)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be written out.

### Details

Interface to the C function `printIpt` which calls the GLPK function `glp_print_ipr`.

### Value

Returns zero on success, otherwise it returns non-zero and prints an error message.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[printSolGLPK](#), [readSolGLPK](#), [writeSolGLPK](#), [readIptGLPK](#), [writeIptGLPK](#), [printMIPGLPK](#), [readMIPGLPK](#) and [writeMIPGLPK](#).

---

printMIPGLPK

*Write Interior-Point Solution in Printable Format*

---

### Description

Low level interface function to the GLPK function `glp_print_mip`. Consult the GLPK documentation for more detailed information.

### Usage

```
printMIPGLPK(lp, fname)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
fname	The name of the text file to be written out.

### Details

Interface to the C function `printMIP` which calls the GLPK function `glp_print_mip`.

### Value

Returns zero on success, otherwise it returns non-zero and prints an error message.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`printSolGLPK`, `readSolGLPK`, `writeSolGLPK`, `printIptGLPK`, `readIptGLPK`, `writeIptGLPK`, `readMIPGLPK` and `writeMIPGLPK`.

---

printRangesGLPK

*Print Sensitivity Analysis Report*

---

### Description

Low level interface function to the GLPK function `glp_print_ranges`. Consult the GLPK documentation for more detailed information.

### Usage

```
printRangesGLPK(lp, numrc = 0, rowcol = NULL, fname = "sar.txt")
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>numrc</code>	Length of the row/column list (argument <code>rowcol</code> ).
<code>rowcol</code>	Ordinal numbers of rows and columns to be analyzed.
<code>fname</code>	A filename.

### Details

Interface to the C function `printRanges` which calls the GLPK function `glp_print_ranges`.

### Value

Zero on success, otherwise non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

printSolGLPK

*Write Basic Solution in Printable Format*

---

### Description

Low level interface function to the GLPK function `glp_print_sol`. Consult the GLPK documentation for more detailed information.

### Usage

```
printSolGLPK(lp, fname)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
fname	The name of the text file to be written out.

### Details

Interface to the C function `printSol` which calls the GLPK function `glp_print_sol`.

### Value

Returns zero on success, otherwise it returns non-zero and prints an error message.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[readSolGLPK](#), [writeSolGLPK](#), [printIptGLPK](#), [readIptGLPK](#), [writeIptGLPK](#), [printMIPGLPK](#), [readMIPGLPK](#) and [writeMIPGLPK](#).

---

`readIptGLPK`*Read Interior-Point Solution From Text File*

---

**Description**

Low level interface function to the GLPK function `glp_read_ip`. Consult the GLPK documentation for more detailed information.

**Usage**

```
readIptGLPK(lp, fname)
```

**Arguments**

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be read in.

**Details**

Interface to the C function `readIpt` which calls the GLPK function `glp_read_ip`.

**Value**

Returns zero on success, otherwise it returns non-zero.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[printSolGLPK](#), [readSolGLPK](#), [writeSolGLPK](#), [printIptGLPK](#), [writeIptGLPK](#), [printMIPGLPK](#), [readMIPGLPK](#) and [writeMIPGLPK](#).

---

`readLPGLPK`*Read Problem Data in CPLEX LP Format*

---

**Description**

Low level interface function to the GLPK function `glp_read_lp`. Consult the GLPK documentation for more detailed information.

**Usage**

```
readLPGLPK(lp, fname)
```

**Arguments**

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be read in.

**Details**

Interface to the C function `readLP` which calls the GLPK function `glp_read_lp`.

**Value**

Returns zero on success, otherwise it returns non-zero and prints an error message.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[readMPSGLPK](#), [readProbGLPK](#), [writeMPSGLPK](#), [writeLPGLPK](#) and [writeProbGLPK](#).

---

`readMIPGLPK`*Read MIP Solution From Text File*

---

### Description

Low level interface function to the GLPK function `glp_read_mip`. Consult the GLPK documentation for more detailed information.

### Usage

```
readMIPGLPK(lp, fname)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be read in.

### Details

Interface to the C function `readMIP` which calls the GLPK function `glp_read_mip`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`printSolGLPK`, `readSolGLPK`, `writeSolGLPK`, `printIptGLPK`, `readIptGLPK`, `writeIptGLPK`, `printMIPGLPK` and `writeMIPGLPK`.



---

`readMPSGLPK`*Read Problem Data in MPS Format*

---

### Description

Low level interface function to the GLPK function `glp_read_mps`. Consult the GLPK documentation for more detailed information.

### Usage

```
readMPSGLPK(lp, fmt, fname)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fmt</code>	MPS format. See <code>glpkConstants</code> , section 'MPS file formats'.
<code>fname</code>	The name of the text file to be read in.

### Details

Interface to the C function `readMPS` which calls the GLPK function `glp_read_mps`.

### Value

Returns zero on success, otherwise it returns non-zero and prints an error message.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`readLPGLPK`, `readProbGLPK`, `writeMPSGLPK`, `writeLPGLPK`, `writeProbGLPK` and `glpkConstants`.

---

`readProbGLPK`*Read Problem Data in GLPK F ormat*

---

**Description**

Low level interface function to the GLPK function `glp_read_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
readProbGLPK(lp, fname)
```

**Arguments**

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be read in.

**Details**

Interface to the C function `readProb` which calls the GLPK function `glp_read_prob`.

**Value**

Returns zero on success, otherwise it returns non-zero and prints an error message.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[readMPSGLPK](#), [readLPGLPK](#), [writeMPSGLPK](#), [writeLPGLPK](#) and [writeProbGLPK](#).

---

`readSolGLPK`*Read Basic Solution From Text File*

---

### Description

Low level interface function to the GLPK function `glp_read_sol`. Consult the GLPK documentation for more detailed information.

### Usage

```
readSolGLPK(lp, fname)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be read in.

### Details

Interface to the C function `readSol` which calls the GLPK function `glp_read_sol`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`printSolGLPK`, `writeSolGLPK`, `printIptGLPK`, `readIptGLPK`, `writeIptGLPK`, `printMIPGLPK`, `readMIPGLPK` and `writeMIPGLPK`.

---

return_codeGLPK	<i>Translates a GLPK Return Code into a Human Readable String</i>
-----------------	---

---

**Description**

Translates a GLPK return code into a human readable string.

**Usage**

```
return_codeGLPK(code)
```

**Arguments**

code	Return code from GLPK.
------	------------------------

**Value**

A character string associated with the GLPK return code.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘return codes’.

---

scaleProbGLPK	<i>Scale Problem Data</i>
---------------	---------------------------

---

**Description**

Low level interface function to the GLPK function `glp_scale_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
scaleProbGLPK(lp, opt)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
opt	Scaling option, see <code>glpkConstants</code> , section 'LP/MIP problem object' for possible values.

**Details**

Interface to the C function `scaleProb` which calls the GLPK function `glp_scale_prob`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#)

---

setBfcpGLPK

*Change Basis Factorization Control Parameters*

---

**Description**

Sets/Changes the values of corresponding members of in the structure `glp_bfcp`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setBfcpGLPK(lp, parm, val)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
parm	A vector containing integer values or symbolic names of the control parameters to be changed (see <code>glpkConstants</code> , section 'Control Parameters').
val	A vector containing the new values for the corresponding control parameters.

**Details**

The Arguments `parm` and `val` must have the same length. The value `val[i]` belongs to the parameter `parm[i]`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#)

---

setColBndGLPK

*Set/Change Column Bounds*

---

**Description**

Low level interface function to the GLPK function `glp_set_col_bnds`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setColBndGLPK(lp, j, type, lb, ub)
```

**Arguments**

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .
<code>type</code>	Column type. For possible values, see <code>glpkConstants</code> , section 'LP/MIP problem object'.
<code>lb</code>	Lower bound.
<code>ub</code>	Upper bound.

**Details**

Interface to the C function setColBnd which calls the GLPK function glp\_set\_col\_bnds.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#)

---

setColKindGLPK	<i>Set Column Kind</i>
----------------	------------------------

---

**Description**

Low level interface function to the GLPK function glp\_set\_col\_kind. Consult the GLPK documentation for more detailed information.

**Usage**

```
setColKindGLPK(lp, j, kind)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
j	Column number j.
kind	Kind of column number j, for possible values see <a href="#">glpkConstants</a> , section 'LP/MIP problem object'.

**Details**

Interface to the C function setColKind which calls the GLPK function glp\_set\_col\_kind.

**Value**

NULL

**Author(s)**

Gabriel Gelius-Dietrich &lt;geliudie@uni-duesseldorf.de&gt;

Maintainer: Mayo Roettger &lt;mayo.roettger@hhu.de&gt;

**References**Based on the package **glpk** by Lopaka Lee.The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.**See Also**[glpkConstants](#)

---

`setColNameGLPK`*Set/Change Column Name*

---

**Description**

Low level interface function to the GLPK function `glp_set_col_name`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setColNameGLPK(lp, j, cname = NULL)
```

**Arguments**

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .
<code>cname</code>	Column name.

**Details**

Interface to the C function `setColName` which calls the GLPK function `glp_set_col_name`.

**Value**

NULL



**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setColsBndsGLPK      *Set/Change Column Bounds*

---

**Description**

This is an advanced version of [setColBndGLPK](#). Here, *j* can be an integer vector, *lb* and *ub* can be numeric vectors.

**Usage**

```
setColsBndsGLPK(lp, j, lb, ub, type = NULL)
```

**Arguments**

<i>lp</i>	An object of class " <a href="#">glpkPtr</a> " as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>j</i>	Vector of column numbers.
<i>lb</i>	Vector of lower bounds.
<i>ub</i>	Vector of upper bounds.
<i>type</i>	Vector of variable types (default: NULL). For possible values, see <a href="#">glpkConstants</a> , section 'LP/MIP problem object'.

**Details**

Interface to the C function `setColsBnds` which calls the GLPK function `glp_set_col_bnds`.

If *type* is set to NULL, the type of the variables will be estimated. If *lb*[*i*] equals *ub*[*i*], variable *j*[*i*] is fixed, otherwise double bounded.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#)

---

setColsBndsObjCoefsGLPK

*Set/Change Column Bounds and Objective Coefficients and/or Constant Term*

---

**Description**

This is a combined version of [setColsBndsGLPK](#) and [setObjCoefsGLPK](#). Here, *j* can be an integer vector, *lb*, *ub* and *obj\_coef* can be numeric vectors.

**Usage**

```
setColsBndsObjCoefsGLPK(lp, j, lb, ub, obj_coef, type = NULL)
```

**Arguments**

<i>lp</i>	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>j</i>	Vector of column numbers.
<i>lb</i>	Vector of lower bounds.
<i>ub</i>	Vector of upper bounds.
<i>obj_coef</i>	Vector of objective coefficients.
<i>type</i>	Vector of variable types (default: NULL). For possible values, see <a href="#">glpkConstants</a> , section 'LP/MIP problem object'.

**Details**

Interface to the C function `setColsBndsObjCoefs` which calls the GLPK functions `glp_set_col_bnds` and `glp_set_obj_coef`.

If *type* is set to NULL, the type of the variables will be estimated. If *lb*[*i*] equals *ub*[*i*], variable *j*[*i*] is fixed, otherwise double bounded.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#)

---

setColsKindGLPK

*Set Column Kind for a Set of Columns*

---

**Description**

This is an advanced version of [setColKindGLPK](#). Here, j can be an integer vector.

**Usage**

```
setColsKindGLPK(lp, j, kind)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
j	An integer vector of column indices.
kind	An integer vector of column kinds, for possible values see <a href="#">glpkConstants</a> , section 'LP/MIP problem object'.

**Details**

Interface to the C function setColsKind which calls the GLPK function glp\_set\_col\_kind.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

## See Also

[glpkConstants](#)

---

setColsNamesGLPK	<i>Set/Change Column Names</i>
------------------	--------------------------------

---

## Description

This is an advanced version of [setColNameGLPK](#). Here, *j* can be an integer vector, *cnames* can be a character vector.

## Usage

```
setColsNamesGLPK(lp, j, cnames = NULL)
```

## Arguments

<i>lp</i>	An object of class " <a href="#">glpkPtr</a> " as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>j</i>	Vector of column numbers.
<i>cnames</i>	Vector of column names of the same length as <i>j</i> or NULL.

## Details

Interface to the C function `setColsNames` which calls the GLPK function `glp_set_col_name`.

If *cnames* is set to NULL, all column names for column indices in *j* will be removed. If *cname*[*k*] is the empty string "", column name *j*[*k*] will be removed.

## Value

NULL

## Author(s)

Gabriel Gelius-Dietrich <[geliudie@uni-duesseldorf.de](mailto:geliudie@uni-duesseldorf.de)>

Maintainer: Mayo Roettger <[mayo.roettger@hhu.de](mailto:mayo.roettger@hhu.de)>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setColStatGLPK	<i>Set column status</i>
----------------	--------------------------

---

### Description

Low level interface function to the GLPK function `glp_set_col_stat`. Consult the GLPK documentation for more detailed information.

### Usage

```
setColStatGLPK(lp, j, stat)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Column number <code>j</code> .
<code>stat</code>	A status parameter, see <code>glpkConstants</code> , section 'LP/MIP problem object' for possible values.

### Details

Interface to the C function `setColStat` which calls the GLPK function `glp_set_col_stat`.

### Value

NULL

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#)

---

setDefaultIptParamGLPK *Sets the Default Control Parameters for the Interior-point Method.*

---

**Description**

Initializes a new structure `glp_iptcp`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setDefaultIptParamGLPK()
```

**Details**

Interface to the C function `setDefaultIptParam` which calls the GLPK function `glp_init_iptcp`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘Control Parameters’.

---

setDefaultMIPParamGLPK *Sets the Default Control Parameters for the MIP Method*

---

**Description**

Initializes a new structure `glp_iocp`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setDefaultMIPParamGLPK()
```

**Details**

Interface to the C function `setDefaultMIPParam` which calls the GLPK function `glp_init_iocp`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘Control Parameters’.

---

`setDefaultSmpParmGLPK` *Sets the Default Control Parameters for the Simplex Methods.*

---

**Description**

Initializes a new structure `glp_smcp`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setDefaultSmpParmGLPK()
```

**Details**

Interface to the C function `setDefaultSmpParam` which calls the GLPK function `glp_init_smcp`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

## See Also

[glpkConstants](#), section ‘Control Parameters’.

---

setInteriorParmGLPK    *Sets/Changes Control Parameters or the Interior-point Method.*

---

## Description

Sets/Changes the values of corresponding members of in the structure `glp_icopt`. Consult the GLPK documentation for more detailed information.

## Usage

```
setInteriorParmGLPK(parm, val)
```

## Arguments

<code>parm</code>	A vector containing integer values or symbolic names of the control parameters to be changed (see <a href="#">glpkConstants</a> , section ‘Control Parameters’) and ‘interior-point solver control parameters’).
<code>val</code>	A vector containing the new values for the corresponding control parameters.

## Details

The Arguments `parm` and `val` must have the same length. The value `val[i]` belongs to the parameter `parm[i]`.

## Value

NULL

## Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.



**See Also**[glpkConstants](#)

---

`setMatColGLPK`*Set (Replace) Column of the Constraint Matrix*

---

**Description**

Low level interface function to the GLPK function `glp_set_mat_col`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setMatColGLPK(lp, j, len, ind, val)
```

**Arguments**

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Replace the <code>j</code> -th column of the constraint matrix of the specified problem object.
<code>len</code>	Number of new column elements.
<code>ind</code>	Row indices of the new column elements.
<code>val</code>	Numerical values of the new column elements.

**Details**

Interface to the C function `setMatCol` which calls the GLPK function `glp_set_mat_col`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

`setMatRowGLPK`*Set (Replace) Row of the Constraint Matrix*

---

### Description

Low level interface function to the GLPK function `glp_set_mat_row`. Consult the GLPK documentation for more detailed information.

### Usage

```
setMatRowGLPK(lp, i, len, ind, val)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Replace the <i>i</i> -th row of the constraint matrix of the specified problem object.
<code>len</code>	Number of new row elements.
<code>ind</code>	Column indices of the new row elements.
<code>val</code>	Numerical values of the new row elements.

### Details

Interface to the C function `setMatRow` which calls the GLPK function `glp_set_mat_row`.

### Value

NULL

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

`setMIPParamGLPK`*Sets/Changes Control Parameters or the MIP Methods*

---

**Description**

Sets/Changes the values of corresponding members of in the structure `glp_iocp`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setMIPParamGLPK(parm, val)
```

**Arguments**

<code>parm</code>	A vector containing integer values or symbolic names of the control parameters to be changed (see <a href="#">glpkConstants</a> , section ‘Control Parameters’ and ‘integer optimizer control parameters’).
<code>val</code>	A vector containing the new values for the corresponding control parameters.

**Details**

The Arguments `parm` and `val` must have the same length. The value `val[i]` belongs to the parameter `parm[i]`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#)

---

setObjCoefGLPK	<i>Set/Change Objective Coefficient or Constant Term</i>
----------------	--

---

### Description

Low level interface function to the GLPK function `glp_set_obj_coef`. Consult the GLPK documentation for more detailed information.

### Usage

```
setObjCoefGLPK(lp, j, obj_coef)
```

### Arguments

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.
obj_coef	Objective coefficient or constant term.

### Details

Interface to the C function `setObjCoef` which calls the GLPK function `glp_set_obj_coef`.

### Value

NULL

### Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>  
Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setObjCoefsGLPK	<i>Set/Change Objective Coefficients and/or Constant Term</i>
-----------------	---

---

### Description

This is an advanced version of [setColBndGLPK](#). Here, `j` can be an integer vector, `obj_coef` can be a numeric vector.

### Usage

```
setObjCoefsGLPK(lp, j, obj_coef)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<code>j</code>	Vector of column numbers.
<code>obj_coef</code>	Vector of objective coefficients.

### Details

Interface to the C function `setObjCoefs` which calls the GLPK function `glp_set_obj_coef`.

### Value

NULL

### Author(s)

Gabriel Gelius-Dietrich <geliudie@uni-duesseldorf.de>  
Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

`setObjDirGLPK`*Set/Change Optimization Direction Flag*

---

**Description**

Low level interface function to the GLPK function `glp_set_obj_dir`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setObjDirGLPK(lp, lpdire)
```

**Arguments**

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>lpdire</code>	Optimization direction flag, which can be <code>GLP_MIN</code> (default) or <code>GLP_MAX</code> .

**Details**

Interface to the C function `setObjDir` which calls the GLPK function `glp_set_obj_dir`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section 'LP/MIP problem object'.

---

setObjNameGLPK	<i>Set/Change Objective Function Name</i>
----------------	---

---

### Description

Low level interface function to the GLPK function `glp_set_obj_name`. Consult the GLPK documentation for more detailed information.

### Usage

```
setObjNameGLPK(lp, oname = NULL)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>oname</code>	Objective Function name.

### Details

Interface to the C function `setObjName` which calls the GLPK function `glp_set_obj_name`.

### Value

NULL

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setProbNameGLPK	<i>Set/Change Problem Name</i>
-----------------	--------------------------------

---

### Description

Low level interface function to the GLPK function `glp_set_prob_name`. Consult the GLPK documentation for more detailed information.

### Usage

```
setProbNameGLPK(lp, pname = NULL)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>pname</code>	Problem name.

### Details

Interface to the C function `setProbName` which calls the GLPK function `glp_set_prob_name`.

### Value

NULL

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.



---

setRhsZeroGLPK	<i>Set/Change all Row Bounds to Zero</i>
----------------	--

---

**Description**

This is an advanced version of [setRowsBndsGLPK](#).

**Usage**

```
setRhsZeroGLPK(lp)
```

**Arguments**

lp                   An object of class "glpkPtr" as returned by [initProbGLPK](#). This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `setRowsBnds` which calls the GLPK function `glp_set_col_bnds`. All row bounds are fixed at zero.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setRiiGLPK	<i>Set row scale factor</i>
------------	-----------------------------

---

**Description**

Low level interface function to the GLPK function `glp_set_rii`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setRiiGLPK(lp, i, rii)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
i	Row number i.
rii	Scale factor \$r_{ii}\$.

**Details**

Interface to the C function `setRii` which calls the GLPK function `glp_set_rii`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setRowBndGLPK	<i>Set/Change Row Bounds</i>
---------------	------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_set_row_bnds`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setRowBndGLPK(lp, i, type, lb, ub)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
i	Row number i.
type	Row type. For possible values, see <code>glpkConstants</code> , section 'LP/MIP problem object'.
lb	Lower bound.
ub	Upper bound.

**Details**

Interface to the C function setRowBnd which calls the GLPK function `glp_set_row_bnds`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#)

---

setRowNameGLPK	<i>Set/Change Row Name</i>
----------------	----------------------------

---

**Description**

Low level interface function to the GLPK function `glp_set_row_name`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setRowNameGLPK(lp, i, rname = NULL)
```

**Arguments**

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .
<code>rname</code>	Row name.

**Details**

Interface to the C function setRowName which calls the GLPK function `glp_set_row_name`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setRowsBndsGLPK      *Set/Change Row Bounds*

---

**Description**

This is an advanced version of [setRowBndGLPK](#). Here, *i* can be an integer vector, *lb* and *ub* can be numeric vectors.

**Usage**

```
setRowsBndsGLPK(lp, i, lb, ub, type = NULL)
```

**Arguments**

<i>lp</i>	An object of class " <a href="#">glpkPtr</a> " as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>i</i>	Vector of row numbers.
<i>lb</i>	Vector of lower bounds.
<i>ub</i>	Vector of upper bounds.
<i>type</i>	Vector of variable types (default: NULL). For possible values, see <a href="#">glpkConstants</a> , section 'LP/MIP problem object'.

**Details**

Interface to the C function `setRowsBnds` which calls the GLPK function `glp_set_row_bnds`.

If *type* is set to NULL, the type of the variables will be estimated. If *lb*[*j*] equals *ub*[*j*], variable *i*[*j*] is fixed, otherwise double bounded.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

## See Also

[glpkConstants](#)

---

setRowsNamesGLPK	<i>Set/Change Row Names</i>
------------------	-----------------------------

---

## Description

This is an advanced version of [setRowNameGLPK](#). Here, *i* can be an integer vector, *rnames* can be a character vector.

## Usage

```
setRowsNamesGLPK(lp, i, rnames = NULL)
```

## Arguments

<i>lp</i>	An object of class " <a href="#">glpkPtr</a> " as returned by <a href="#">initProbGLPK</a> . This is basically a pointer to a GLPK problem object.
<i>i</i>	Vector of row numbers.
<i>rnames</i>	Vector of row names of the same length as <i>i</i> or NULL.

## Details

Interface to the C function `setRowsNames` which calls the GLPK function `glp_set_row_name`.

If *rnames* is set to NULL, all row names for row indices in *i* will be removed. If *rname*[*k*] is the empty string "", row name *i*[*k*] will be removed.

## Value

NULL

## Author(s)

Gabriel Gelius-Dietrich <[geliudie@uni-duesseldorf.de](mailto:geliudie@uni-duesseldorf.de)>

Maintainer: Mayo Roettger <[mayo.roettger@hhu.de](mailto:mayo.roettger@hhu.de)>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

setRowStatGLPK	<i>Set row status</i>
----------------	-----------------------

---

### Description

Low level interface function to the GLPK function `glp_set_row_stat`. Consult the GLPK documentation for more detailed information.

### Usage

```
setRowStatGLPK(lp, i, stat)
```

### Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>i</code>	Row number <code>i</code> .
<code>stat</code>	A status parameter, see <code>glpkConstants</code> for possible values.

### Details

Interface to the C function `setRowStat` which calls the GLPK function `glp_set_row_stat`, section 'LP/MIP problem object'.

### Value

NULL

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#)

---

setSimplexParmGLPK      *Sets/Changes Control Parameters or the Simplex Methods.*

---

### Description

Sets/Changes the values of corresponding members of in the structure `glp_smcp`. Consult the GLPK documentation for more detailed information.

### Usage

```
setSimplexParmGLPK(parm, val)
```

### Arguments

<code>parm</code>	A vector containing integer values or symbolic names of the control parameters to be changed (see <a href="#">glpkConstants</a> , section ‘Control Parameters’ and ‘simplex method control parameters’).
<code>val</code>	A vector containing the new values for the corresponding control parameters.

### Details

The Arguments `parm` and `val` must have the same length. The value `val[i]` belongs to the parameter `parm[i]`.

### Value

NULL

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#)

---

setSjjGLPK	<i>Retrieve column scale factor</i>
------------	-------------------------------------

---

**Description**

Low level interface function to the GLPK function `glp_set_sjj`. Consult the GLPK documentation for more detailed information.

**Usage**

```
setSjjGLPK(lp, j, sjj)
```

**Arguments**

lp	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
j	Column number j.
sjj	Scale factor <code>\$s_jj\$</code> .

**Details**

Interface to the C function `setSjj` which calls the GLPK function `glp_set_sjj`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.



---

solveInteriorGLPK      *Solve LP Problem with the Interior-Point Method*

---

### Description

Low level interface function to the GLPK function `glp_interior`. Consult the GLPK documentation for more detailed information.

### Usage

```
solveInteriorGLPK(lp)
```

### Arguments

`lp`                    An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `solveInterior` which calls the GLPK function `glp_interior`.

### Value

A return code.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'return codes' and [return\\_codeGLPK](#).

---

`solveMIPGLPK`*Solve MIP Problem with the Branch-and-Cut Method*

---

### Description

Low level interface function to the GLPK function `glp_intopt`. Consult the GLPK documentation for more detailed information.

### Usage

```
solveMIPGLPK(lp)
```

### Arguments

`lp` An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `solveMIP` which calls the GLPK function `glp_intopt`.

### Value

A return code.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'return codes' and [return\\_codeGLPK](#).

---

solveSimplexExactGLPK *Solve LP Problem in Exact Arithmetic*

---

### Description

Low level interface function to the GLPK function `glp_exact`. Consult the GLPK documentation for more detailed information.

### Usage

```
solveSimplexExactGLPK(lp)
```

### Arguments

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `solveSimplexExact` which calls the GLPK function `glp_exact`.

### Value

A return code.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package `glpk` by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'return codes' and [return\\_codeGLPK](#).

---

solveSimplexGLPK      *Solve LP Problem with the Primal or Dual Simplex Method*

---

### Description

Low level interface function to the GLPK function `glp_simplex`. Consult the GLPK documentation for more detailed information.

### Usage

```
solveSimplexGLPK(lp)
```

### Arguments

`lp`                    An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

### Details

Interface to the C function `solveSimplex` which calls the GLPK function `glp_simplex`.

### Value

A return code.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[glpkConstants](#), section 'return codes' and [return\\_codeGLPK](#).

---

sortMatrixGLPK	<i>Sort Elements of the Constraint Matrix</i>
----------------	---

---

**Description**

Low level interface function to the GLPK function `glp_sort_matrix`. Consult the GLPK documentation for more detailed information.

**Usage**

```
sortMatrixGLPK(lp)
```

**Arguments**

`lp` An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `sortMatrix` which calls the GLPK function `glp_sort_matrix`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

status_codeGLPK	<i>Translates a GLPK Status Value into a Human Readable String</i>
-----------------	--

---

**Description**

Translates a GLPK status code into a human readable string.

**Usage**

```
status_codeGLPK(code)
```

**Arguments**

code                    Status code from GLPK.

**Value**

A character string associated with the GLPK status code.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section ‘LP/MIP problem object’.

---

stdBasisGLPK

*Construct Standard Initial LP Basis*

---

**Description**

Low level interface function to the GLPK function `glp_std_basis`. Consult the GLPK documentation for more detailed information.

**Usage**

```
stdBasisGLPK(lp)
```

**Arguments**

lp                    An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `stdBasis` which calls the GLPK function `glp_std_basis`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

termOutGLPK

*Enable/Disable Terminal Output*

---

**Description**

Low level interface function to the GLPK function `glp_term_out`. Consult the GLPK documentation for more detailed information.

**Usage**

```
termOutGLPK(flag)
```

**Arguments**

`flag`                   GLPK enable/disable flag: `GLP_ON` or `GLP_OFF`.

**Details**

Interface to the C function `termOut` which calls the GLPK function `glp_term_out`.

**Value**

Returns the previous value of the terminal output flag.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[glpkConstants](#), section 'enable/disable flag'.

unscaleProbGLPK      *Problem unscaling*

---

**Description**

Low level interface function to the GLPK function `glp_unscale_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
unscaleProbGLPK(lp)
```

**Arguments**

`lp`      An object of class "glpkPtr" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `unscaleProb` which calls the GLPK function `glp_unscale_prob`.

**Value**

NULL

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

versionGLPK      *Determine GLPK Callable Library Version*

---

**Description**

Low level interface function to the GLPK function `glp_version`. Consult the GLPK documentation for more detailed information.

**Usage**

```
versionGLPK()
```



**Details**

Interface to the C function version which calls the GLPK function `glp_version`.

**Value**

Returns a single character value containing the GLPK version number.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

---

warmUpGLPK

*Warm Up LP Basis*

---

**Description**

Low level interface function to the GLPK function `glp_warm_up`. Consult the GLPK documentation for more detailed information.

**Usage**

```
warmUpGLPK(lp)
```

**Arguments**

`lp` An object of class "`glpkPtr`" as returned by `initProbGLPK`. This is basically a pointer to a GLPK problem object.

**Details**

Interface to the C function `warmUp` which calls the GLPK function `glp_warm_up`.

**Value**

Status of "warming up".

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>

---

writeIptGLPK

*Write Interior-Point Solution to Text File*

---

## Description

Low level interface function to the GLPK function `glp_write_ip`. Consult the GLPK documentation for more detailed information.

## Usage

```
writeIptGLPK(lp, fname)
```

## Arguments

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be written out.

## Details

Interface to the C function `writeIpt` which calls the GLPK function `glp_write_ip`.

## Value

Returns zero on success, otherwise it returns non-zero.

## Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

## References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

## See Also

`printSolGLPK`, `readSolGLPK`, `writeSolGLPK`, `printIptGLPK`, `readIptGLPK`, `printMIPGLPK`, `readMIPGLPK` and `writeMIPGLPK`.

---

`writeLPGLPK`*Write Problem Data in CPLEX LP Format*

---

### Description

Low level interface function to the GLPK function `glp_write_lp`. Consult the GLPK documentation for more detailed information.

### Usage

```
writeLPGLPK(lp, fname)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be written out.

### Details

Interface to the C function `writeLP` which calls the GLPK function `glp_write_lp`.

### Value

Returns zero on success, otherwise it returns non-zero and prints an error message.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[readMPSGLPK](#), [readLPGLPK](#), [readProbGLPK](#), [writeMPSGLPK](#) and [writeProbGLPK](#).

---

`writeMIPGLPK`*Write MIP Solution to Text File*

---

### Description

Low level interface function to the GLPK function `glp_write_mip`. Consult the GLPK documentation for more detailed information.

### Usage

```
writeMIPGLPK(lp, fname)
```

### Arguments

<code>lp</code>	An object of class " <code>glpPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be written out.

### Details

Interface to the C function `writeMIP` which calls the GLPK function `glp_write_mip`.

### Value

Returns zero on success, otherwise it returns non-zero.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

`printSolGLPK`, `readSolGLPK`, `writeSolGLPK`, `printIptGLPK`, `readIptGLPK`, `writeIptGLPK`, `printMIPGLPK` and `readMIPGLPK`.

---

`writeMPSGLPK`*Write Problem Data in MPS Format*

---

### Description

Low level interface function to the GLPK function `glp_write_mps`. Consult the GLPK documentation for more detailed information.

### Usage

```
writeMPSGLPK(lp, fmt, fname)
```

### Arguments

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fmt</code>	MPS format. See <code>glpkConstants</code> , section 'MPS file formats'.
<code>fname</code>	The name of the text file to be written out.

### Details

Interface to the C function `writeMPS` which calls the GLPK function `glp_write_mps`.

### Value

Returns zero on success, otherwise it returns non-zero and prints an error message.

### Author(s)

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

### References

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

### See Also

[readMPSGLPK](#), [readLPGLPK](#), [readProbGLPK](#), [writeLPGLPK](#), [writeProbGLPK](#) and [glpkConstants](#).

---

`writeProbGLPK`*Write Problem Data in GLPK Format*

---

**Description**

Low level interface function to the GLPK function `glp_write_prob`. Consult the GLPK documentation for more detailed information.

**Usage**

```
writeProbGLPK(lp, fname)
```

**Arguments**

<code>lp</code>	An object of class "glpkPtr" as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be written out.

**Details**

Interface to the C function `writeProb` which calls the GLPK function `glp_write_prob`.

**Value**

Returns zero on success, otherwise it returns non-zero and prints an error message.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

[readMPSGLPK](#), [readLPGLPK](#), [readProbGLPK](#), [writeLPGLPK](#) and [writeMPSGLPK](#).

---

`writeSolGLPK`*Write Basic Solution to Text File*

---

**Description**

Low level interface function to the GLPK function `glp_write_sol`. Consult the GLPK documentation for more detailed information.

**Usage**

```
writeSolGLPK(lp, fname)
```

**Arguments**

<code>lp</code>	An object of class " <code>glpkPtr</code> " as returned by <code>initProbGLPK</code> . This is basically a pointer to a GLPK problem object.
<code>fname</code>	The name of the text file to be written out.

**Details**

Interface to the C function `writeSol` which calls the GLPK function `glp_write_sol`.

**Value**

Returns zero on success, otherwise it returns non-zero.

**Author(s)**

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

Maintainer: Mayo Roettger <mayo.roettger@hhu.de>

**References**

Based on the package **glpk** by Lopaka Lee.

The GNU GLPK home page at <http://www.gnu.org/software/glpk/glpk.html>.

**See Also**

`printSolGLPK`, `readSolGLPK`, `printIptGLPK`, `readIptGLPK`, `writeIptGLPK`, `printMIPGLPK`, `readMIPGLPK` and `writeMIPGLPK`.

# Index

## \* optimize

- addColsGLPK, 6
- addRowsGLPK, 7
- advBasisGLPK, 8
- bfExistsGLPK, 9
- bfUpdatedGLPK, 10
- checkDupGLPK, 11
- copyProbGLPK, 12
- cpxBasisGLPK, 13
- createIndexGLPK, 13
- delColsGLPK, 14
- deleteIndexGLPK, 15
- delProbGLPK, 16
- delRowsGLPK, 16
- eraseProbGLPK, 17
- factorizeGLPK, 18
- findColGLPK, 19
- findRowGLPK, 20
- getBfcpGLPK, 21
- getBheadGLPK, 22
- getCbindGLPK, 23
- getColDualGLPK, 24
- getColDualIptGLPK, 25
- getColKindGLPK, 26
- getColLowBndGLPK, 27
- getColNameGLPK, 28
- getColPrimGLPK, 29
- getColPrimIptGLPK, 30
- getColsDualGLPK, 31
- getColsDualIptGLPK, 31
- getColsKindGLPK, 32
- getColsLowBndsGLPK, 33
- getColsPrimGLPK, 34
- getColsPrimIptGLPK, 34
- getColsStatGLPK, 35
- getColStatGLPK, 36
- getColsUppBndsGLPK, 37
- getColTypeGLPK, 38
- getColUppBndGLPK, 39
- getDualStatGLPK, 40
- getInteriorParmGLPK, 41
- getMatColGLPK, 42
- getMatRowGLPK, 43
- getMIPParmGLPK, 44
- getNumBinGLPK, 45
- getNumColsGLPK, 45
- getNumIntGLPK, 46
- getNumNnzGLPK, 47
- getNumRowsGLPK, 48
- getObjCoefGLPK, 48
- getObjCoefsGLPK, 49
- getObjDirGLPK, 50
- getObjNameGLPK, 51
- getObjValGLPK, 52
- getObjValIptGLPK, 52
- getPrimStatGLPK, 53
- getProbNameGLPK, 54
- getRbindGLPK, 55
- getRiiGLPK, 56
- getRowDualGLPK, 57
- getRowDualIptGLPK, 58
- getRowLowBndGLPK, 59
- getRowNameGLPK, 60
- getRowPrimGLPK, 61
- getRowPrimIptGLPK, 62
- getRowsDualGLPK, 63
- getRowsDualIptGLPK, 63
- getRowsLowBndsGLPK, 64
- getRowsPrimGLPK, 65
- getRowsPrimIptGLPK, 66
- getRowsStatGLPK, 66
- getRowStatGLPK, 67
- getRowsTypesGLPK, 68
- getRowsUppBndsGLPK, 69
- getRowTypeGLPK, 70
- getRowUppBndGLPK, 71
- getSimplexParmGLPK, 72
- getSjjGLPK, 73



- getSolStatGLPK, 74
- getSolStatIptGLPK, 75
- getUnbndRayGLPK, 76
- glpkAPI-package, 5
- glpkConstants, 77
- glpkPtr-class, 83
- initProbGLPK, 84
- loadMatrixGLPK, 85
- mipColsValGLPK, 86
- mipColValGLPK, 87
- mipObjValGLPK, 88
- mipRowsValGLPK, 88
- mipRowValGLPK, 89
- mipStatusGLPK, 90
- mplAllocWkspGLPK, 91
- mplBuildProbGLPK, 92
- mplFreeWkspGLPK, 93
- mplGenerateGLPK, 94
- mplPostsolveGLPK, 95
- mplReadDataGLPK, 96
- mplReadModelGLPK, 97
- printIptGLPK, 98
- printMIPGLPK, 99
- printRangesGLPK, 100
- printSolGLPK, 101
- readIptGLPK, 102
- readLPGLPK, 103
- readMIPGLPK, 104
- readMPSGLPK, 105
- readProbGLPK, 106
- readSolGLPK, 107
- return\_codeGLPK, 108
- scaleProbGLPK, 108
- setBfcpGLPK, 109
- setColBndGLPK, 110
- setColKindGLPK, 111
- setColNameGLPK, 112
- setColsBndsGLPK, 113
- setColsBndsObjCoefsGLPK, 114
- setColsKindGLPK, 115
- setColsNamesGLPK, 116
- setColStatGLPK, 117
- setDefaultIptParmGLPK, 118
- setDefaultMIPParmGLPK, 118
- setDefaultSmpParmGLPK, 119
- setInteriorParmGLPK, 120
- setMatColGLPK, 121
- setMatRowGLPK, 122
- setMIPParmGLPK, 123
- setObjCoefGLPK, 124
- setObjCoefsGLPK, 125
- setObjDirGLPK, 126
- setObjNameGLPK, 127
- setProbNameGLPK, 128
- setRhsZeroGLPK, 129
- setRiiGLPK, 129
- setRowBndGLPK, 130
- setRowNameGLPK, 131
- setRowsBndsGLPK, 132
- setRowsNamesGLPK, 133
- setRowStatGLPK, 134
- setSimplexParmGLPK, 135
- setSjjGLPK, 136
- solveInteriorGLPK, 137
- solveMIPGLPK, 138
- solveSimplexExactGLPK, 139
- solveSimplexGLPK, 140
- sortMatrixGLPK, 141
- status\_codeGLPK, 141
- stdBasisGLPK, 142
- termOutGLPK, 143
- unscaleProbGLPK, 144
- versionGLPK, 144
- warmUpGLPK, 145
- writeIptGLPK, 146
- writeLPGLPK, 147
- writeMIPGLPK, 148
- writeMPSGLPK, 149
- writeProbGLPK, 150
- writeSolGLPK, 151
- \* **package**
  - glpkAPI-package, 5
- addColsGLPK, 6
- addRowsGLPK, 7
- advBasisGLPK, 8
- bfExistsGLPK, 9
- bfUpdatedGLPK, 10
- BINARIZE (glpkConstants), 77
- BR\_TECH (glpkConstants), 77
- BT\_TECH (glpkConstants), 77
- CB\_FUNC (glpkConstants), 77
- CB\_SIZE (glpkConstants), 77
- checkDupGLPK, 11
- CLQ\_CUTS (glpkConstants), 77

- constantsGLPK (glpkConstants), 77
- copyProbGLPK, 12
- COV\_CUTS (glpkConstants), 77
- cpxBasisGLPK, 13
- createIndexGLPK, 13, 19, 20
- delColsGLPK, 14
- deleteIndexGLPK, 15
- delProbGLPK, 16
- delRowsGLPK, 16
- EPS\_TOL (glpkConstants), 77
- eraseProbGLPK, 17
- factorizeGLPK, 18
- findColGLPK, 19
- findRowGLPK, 20
- FP\_HEUR (glpkConstants), 77
- getBfcpGLPK, 21
- getBheadGLPK, 22
- getCbindGLPK, 23
- getColDualGLPK, 24, 31
- getColDualIptGLPK, 25, 31
- getColKindGLPK, 26, 32
- getColLowBndGLPK, 27, 33
- getColNameGLPK, 28
- getColPrimGLPK, 29, 34
- getColPrimIptGLPK, 30
- getColsDualGLPK, 31
- getColsDualIptGLPK, 31
- getColsKindGLPK, 32
- getColsLowBndsGLPK, 33
- getColsPrimGLPK, 34
- getColsPrimIptGLPK, 34
- getColsStatGLPK, 35
- getColStatGLPK, 35, 36
- getColsUppBndsGLPK, 37
- getColTypeGLPK, 38
- getColUppBndGLPK, 37, 39
- getDualStatGLPK, 40
- getInteriorParmGLPK, 41
- getMatColGLPK, 42
- getMatRowGLPK, 43
- getMIPParmGLPK, 44
- getNumBinGLPK, 45
- getNumColsGLPK, 45
- getNumIntGLPK, 46
- getNumNnzGLPK, 47
- getNumRowsGLPK, 48
- getObjCoefGLPK, 48, 49
- getObjCoefsGLPK, 49
- getObjDirGLPK, 50
- getObjNameGLPK, 51
- getObjValGLPK, 52
- getObjValIptGLPK, 52
- getPrimStatGLPK, 53
- getProbNameGLPK, 54
- getRbindGLPK, 55
- getRiiGLPK, 56
- getRowDualGLPK, 57, 63
- getRowDualIptGLPK, 58, 63
- getRowLowBndGLPK, 59, 64
- getRowNameGLPK, 60
- getRowPrimGLPK, 61, 65
- getRowPrimIptGLPK, 62, 66
- getRowsDualGLPK, 63
- getRowsDualIptGLPK, 63
- getRowsLowBndsGLPK, 64
- getRowsPrimGLPK, 65
- getRowsPrimIptGLPK, 66
- getRowsStatGLPK, 66
- getRowStatGLPK, 66, 67
- getRowsTypesGLPK, 68
- getRowsUppBndsGLPK, 69
- getRowTypeGLPK, 68, 70
- getRowUppBndGLPK, 69, 71
- getSimplexParmGLPK, 72
- getSjjGLPK, 73
- getSolStatGLPK, 74
- getSolStatIptGLPK, 75
- getUnbndRayGLPK, 76
- glp\_add\_cols (addColsGLPK), 6
- glp\_add\_rows (addRowsGLPK), 7
- glp\_adv\_basis (advBasisGLPK), 8
- GLP\_BF\_BG (glpkConstants), 77
- GLP\_BF\_BTf (glpkConstants), 77
- glp\_bf\_exists (bfExistsGLPK), 9
- GLP\_BF\_FT (glpkConstants), 77
- GLP\_BF\_GR (glpkConstants), 77
- GLP\_BF\_LUF (glpkConstants), 77
- glp\_bf\_updated (bfUpdatedGLPK), 10
- GLP\_BR\_DTH (glpkConstants), 77
- GLP\_BR\_FFV (glpkConstants), 77
- GLP\_BR\_LFV (glpkConstants), 77
- GLP\_BR\_MFV (glpkConstants), 77
- GLP\_BR\_PCH (glpkConstants), 77

- GLP\_BS (glpkConstants), 77
- GLP\_BT\_BFS (glpkConstants), 77
- GLP\_BT\_BLB (glpkConstants), 77
- GLP\_BT\_BPH (glpkConstants), 77
- GLP\_BT\_DFS (glpkConstants), 77
- GLP\_BV (glpkConstants), 77
- glp\_check\_dup (checkDupGLPK), 11
- glp\_copy\_prob (copyProbGLPK), 12
- glp\_cpx\_basis (cpxBasisGLPK), 13
- glp\_create\_index (createIndexGLPK), 13
- glp\_create\_prob (initProbGLPK), 84
- GLP\_CV (glpkConstants), 77
- GLP\_DB (glpkConstants), 77
- glp\_del\_cols (delColsGLPK), 14
- glp\_del\_rows (delRowsGLPK), 16
- glp\_delete\_index (deleteIndexGLPK), 15
- glp\_delete\_prob (delProbGLPK), 16
- GLP\_DN\_BRNCH (glpkConstants), 77
- GLP\_DUAL (glpkConstants), 77
- GLP\_DUALP (glpkConstants), 77
- GLP\_EBADB (glpkConstants), 77
- GLP\_EBOUND (glpkConstants), 77
- GLP\_ECOND (glpkConstants), 77
- GLP\_EDATA (glpkConstants), 77
- GLP\_EFAIL (glpkConstants), 77
- GLP\_EINSTAB (glpkConstants), 77
- GLP\_EITLIM (glpkConstants), 77
- GLP\_EMIPGAP (glpkConstants), 77
- GLP\_ENOCVG (glpkConstants), 77
- GLP\_ENODFS (glpkConstants), 77
- GLP\_ENOFEAS (glpkConstants), 77
- GLP\_ENOPFS (glpkConstants), 77
- GLP\_EOBJLL (glpkConstants), 77
- GLP\_EOBJJUL (glpkConstants), 77
- GLP\_ERANGE (glpkConstants), 77
- glp\_erase\_prob (eraseProbGLPK), 17
- GLP\_EROOT (glpkConstants), 77
- GLP\_ESING (glpkConstants), 77
- GLP\_ESTOP (glpkConstants), 77
- GLP\_ETMLIM (glpkConstants), 77
- glp\_exact (solveSimplexExactGLPK), 139
- glp\_factorize (factorizeGLPK), 18
- GLP\_FEAS (glpkConstants), 77
- glp\_find\_col (findColGLPK), 19
- glp\_find\_row (findRowGLPK), 20
- GLP\_FR (glpkConstants), 77
- GLP\_FX (glpkConstants), 77
- glp\_get\_bfcp (getBfcpGLPK), 21
- glp\_get\_bhead (getBheadGLPK), 22
- glp\_get\_col\_bind (getCbindGLPK), 23
- glp\_get\_col\_dual (getColDualGLPK), 24
- glp\_get\_col\_kind (getColKindGLPK), 26
- glp\_get\_col\_lb (getColLowBndGLPK), 27
- glp\_get\_col\_name (getColNameGLPK), 28
- glp\_get\_col\_prim (getColPrimGLPK), 29
- glp\_get\_col\_stat (getColStatGLPK), 36
- glp\_get\_col\_type (getColTypeGLPK), 38
- glp\_get\_col\_ub (getColUppBndGLPK), 39
- glp\_get\_dual\_stat (getDualStatGLPK), 40
- glp\_get\_mat\_col (getMatColGLPK), 42
- glp\_get\_mat\_row (getMatRowGLPK), 43
- glp\_get\_num\_bin (getNumBinGLPK), 45
- glp\_get\_num\_cols (getNumColsGLPK), 45
- glp\_get\_num\_int (getNumIntGLPK), 46
- glp\_get\_num\_nz (getNumNnzGLPK), 47
- glp\_get\_num\_rows (getNumRowsGLPK), 48
- glp\_get\_obj\_coef (getObjCoefGLPK), 48
- glp\_get\_obj\_dir (getObjDirGLPK), 50
- glp\_get\_obj\_name (getObjNameGLPK), 51
- glp\_get\_obj\_val (getObjValGLPK), 52
- glp\_get\_prim\_stat (getPrimStatGLPK), 53
- glp\_get\_prob\_name (getProbNameGLPK), 54
- glp\_get\_rii (getRiiGLPK), 56
- glp\_get\_row\_bind (getRowBindGLPK), 55
- glp\_get\_row\_dual (getRowDualGLPK), 57
- glp\_get\_row\_lb (getRowLowBndGLPK), 59
- glp\_get\_row\_name (getRowNameGLPK), 60
- glp\_get\_row\_prim (getRowPrimGLPK), 61
- glp\_get\_row\_stat (getRowStatGLPK), 67
- glp\_get\_row\_type (getRowTypeGLPK), 70
- glp\_get\_row\_ub (getRowUppBndGLPK), 71
- glp\_get\_sjj (getSjjGLPK), 73
- glp\_get\_status (getSolStatGLPK), 74
- glp\_get\_unbnd\_ray (getUnbndRayGLPK), 76
- GLP\_IBINGO (glpkConstants), 77
- GLP\_IBRANCH (glpkConstants), 77
- GLP\_ICUTGEN (glpkConstants), 77
- GLP\_IHEUR (glpkConstants), 77
- GLP\_INFEAS (glpkConstants), 77
- glp\_init\_iocp (setDefaultMIPParamGLPK), 118
- glp\_init\_iptcp (setDefaultIptParamGLPK), 118
- glp\_interior (solveInteriorGLPK), 137
- glp\_intopt (solveMIPGLPK), 138
- GLP\_IPREPRO (glpkConstants), 77

- GLP\_IPT (glpkConstants), 77
- glp\_apt\_col\_dual (getColDualIptGLPK), 25
- glp\_apt\_col\_prim (getColPrimIptGLPK), 30
- glp\_apt\_obj\_val (getObjValIptGLPK), 52
- glp\_apt\_row\_dual (getRowDualIptGLPK), 58
- glp\_apt\_row\_prim (getRowPrimIptGLPK), 62
- glp\_apt\_status (getSolStatIptGLPK), 75
- GLP\_IROWGEN (glpkConstants), 77
- GLP\_ISELECT (glpkConstants), 77
- GLP\_IV (glpkConstants), 77
- GLP\_KKT\_CS (glpkConstants), 77
- GLP\_KKT\_DB (glpkConstants), 77
- GLP\_KKT\_DE (glpkConstants), 77
- GLP\_KKT\_PB (glpkConstants), 77
- GLP\_KKT\_PE (glpkConstants), 77
- GLP\_LO (glpkConstants), 77
- glp\_load\_matrix (loadMatrixGLPK), 85
- GLP\_MAX (glpkConstants), 77
- GLP\_MIN (glpkConstants), 77
- GLP\_MIP (glpkConstants), 77
- glp\_mip\_col\_val (mipColValGLPK), 87
- glp\_mip\_obj\_val (mipObjValGLPK), 88
- glp\_mip\_row\_val (mipRowValGLPK), 89
- glp\_mip\_status (mipStatusGLPK), 90
- glp\_mpl\_alloc\_wksp (mplAllocWkspGLPK), 91
- glp\_mpl\_build\_prob (mplBuildProbGLPK), 92
- glp\_mpl\_free\_wksp (mplFreeWkspGLPK), 93
- glp\_mpl\_generate (mplGenerateGLPK), 94
- glp\_mpl\_postsolve (mplPostsolveGLPK), 95
- glp\_mpl\_read\_data (mplReadDataGLPK), 96
- glp\_mpl\_read\_model (mplReadModelGLPK), 97
- GLP\_MPS\_DECK (glpkConstants), 77
- GLP\_MPS\_FILE (glpkConstants), 77
- GLP\_MSG\_ALL (glpkConstants), 77
- GLP\_MSG\_DBG (glpkConstants), 77
- GLP\_MSG\_ERR (glpkConstants), 77
- GLP\_MSG\_OFF (glpkConstants), 77
- GLP\_MSG\_ON (glpkConstants), 77
- GLP\_NF (glpkConstants), 77
- GLP\_NL (glpkConstants), 77
- GLP\_NO\_BRNCH (glpkConstants), 77
- GLP\_NOFEAS (glpkConstants), 77
- GLP\_NS (glpkConstants), 77
- GLP\_NU (glpkConstants), 77
- GLP\_OFF (glpkConstants), 77
- GLP\_ON (glpkConstants), 77
- GLP\_OPT (glpkConstants), 77
- GLP\_ORD\_AMD (glpkConstants), 77
- GLP\_ORD\_NONE (glpkConstants), 77
- GLP\_ORD\_QMD (glpkConstants), 77
- GLP\_ORD\_SYMMAMD (glpkConstants), 77
- GLP\_PP\_ALL (glpkConstants), 77
- GLP\_PP\_NONE (glpkConstants), 77
- GLP\_PP\_ROOT (glpkConstants), 77
- GLP\_PRIMAL (glpkConstants), 77
- glp\_print\_apt (printIptGLPK), 98
- glp\_print\_mip (printMIPGLPK), 99
- glp\_print\_ranges (printRangesGLPK), 100
- glp\_print\_sol (printSolGLPK), 101
- GLP\_PT\_PSE (glpkConstants), 77
- GLP\_PT\_STD (glpkConstants), 77
- glp\_read\_apt (readIptGLPK), 102
- glp\_read\_lp (readLPGLPK), 103
- glp\_read\_mip (readMIPGLPK), 104
- glp\_read\_mps (readMPSGLPK), 105
- glp\_read\_prob (readProbGLPK), 106
- glp\_read\_sol (readSolGLPK), 107
- GLP\_RF\_CLQ (glpkConstants), 77
- GLP\_RF\_COV (glpkConstants), 77
- GLP\_RF\_CUT (glpkConstants), 77
- GLP\_RF\_GMI (glpkConstants), 77
- GLP\_RF\_LAZY (glpkConstants), 77
- GLP\_RF\_MIR (glpkConstants), 77
- GLP\_RF\_REG (glpkConstants), 77
- GLP\_RT\_HAR (glpkConstants), 77
- GLP\_RT\_STD (glpkConstants), 77
- glp\_scale\_prob (scaleProbGLPK), 108
- glp\_set\_bfcp (setBfcpGLPK), 109
- glp\_set\_col\_bnds (setColBndGLPK), 110
- glp\_set\_col\_kind (setColKindGLPK), 111
- glp\_set\_col\_name (setColNameGLPK), 112
- glp\_set\_col\_stat (setColStatGLPK), 117
- glp\_set\_mat\_col (setMatColGLPK), 121
- glp\_set\_mat\_row (setMatRowGLPK), 122
- glp\_set\_obj\_coef (setObjCoefGLPK), 124
- glp\_set\_obj\_dir (setObjDirGLPK), 126
- glp\_set\_obj\_name (setObjNameGLPK), 127
- glp\_set\_prob\_name (setProbNameGLPK), 128
- glp\_set\_rii (setRiiGLPK), 129
- glp\_set\_row\_bnds (setRowBndGLPK), 130
- glp\_set\_row\_name (setRowNameGLPK), 131
- glp\_set\_row\_stat (setRowStatGLPK), 134
- glp\_set\_sjj (setSjjGLPK), 136

- GLP\_SF\_2N (glpkConstants), 77
- GLP\_SF\_AUTO (glpkConstants), 77
- GLP\_SF\_EQ (glpkConstants), 77
- GLP\_SF\_GM (glpkConstants), 77
- GLP\_SF\_SKIP (glpkConstants), 77
- glp\_simplex (solveSimplexGLPK), 140
- GLP\_SOL (glpkConstants), 77
- glp\_sort\_matrix (sortMatrixGLPK), 141
- glp\_std\_basis (stdBasisGLPK), 142
- glp\_term\_out (termOutGLPK), 143
- GLP\_UNBND (glpkConstants), 77
- GLP\_UNDEF (glpkConstants), 77
- glp\_unscale\_prob (unscaleProbGLPK), 144
- GLP\_UP (glpkConstants), 77
- GLP\_UP\_BRNCH (glpkConstants), 77
- glp\_version (versionGLPK), 144
- glp\_warm\_up (warmUpGLPK), 145
- glp\_write\_ipt (writeIptGLPK), 146
- glp\_write\_lp (writeLPGLPK), 147
- glp\_write\_mip (writeMIPGLPK), 148
- glp\_write\_mps (writeMPSGLPK), 149
- glp\_write\_prob (writeProbGLPK), 150
- glp\_write\_sol (writeSolGLPK), 151
- glpk\_Constants (glpkConstants), 77
- glpkAPI (glpkAPI-package), 5
- glpkAPI-package, 5
- glpkConstants, 12, 19, 21, 36, 38, 40, 41, 44, 51, 54, 68–70, 72, 74, 75, 77, 95, 105, 108–121, 123, 126, 130–135, 137–140, 142, 143, 149
- glpkPointer (glpkPtr-class), 83
- glpkPointer, glpkPtr-method (glpkPtr-class), 83
- glpkPtr, 7–10, 12–40, 42, 43, 45–71, 73–76, 85–107, 109–117, 121, 122, 124–134, 136–142, 144–151
- glpkPtr (glpkPtr-class), 83
- glpkPtr-class, 83
- glpkPtrType (glpkPtr-class), 83
- glpkPtrType, glpkPtr-method (glpkPtr-class), 83
- glpkPtrType<- (glpkPtr-class), 83
- glpkPtrType<- , glpkPtr-method (glpkPtr-class), 83
- GMI\_CUTS (glpkConstants), 77
- initProbGLPK, 7–10, 12–40, 42, 43, 45–71, 73–76, 84, 84, 85–90, 98–107, 109–117, 121, 122, 124–134, 136–142, 144–151
- isGLPKpointer (glpkPtr-class), 83
- isGLPKpointer, glpkPtr-method (glpkPtr-class), 83
- isNULLpointerGLPK (glpkPtr-class), 83
- isNULLpointerGLPK, glpkPtr-method (glpkPtr-class), 83
- isTRWKSpointer (glpkPtr-class), 83
- isTRWKSpointer, glpkPtr-method (glpkPtr-class), 83
- IT\_LIM (glpkConstants), 77
- loadMatrixGLPK, 85
- LU\_SIZE (glpkConstants), 77
- MAX\_GRO (glpkConstants), 77
- METH (glpkConstants), 77
- MIP\_GAP (glpkConstants), 77
- mipColsValGLPK, 86
- mipColValGLPK, 86, 87
- mipObjValGLPK, 88
- mipRowsValGLPK, 88
- mipRowValGLPK, 88, 89
- mipStatusGLPK, 90
- MIR\_CUTS (glpkConstants), 77
- mplAllocWkspGLPK, 84, 91, 92–97
- mplBuildProbGLPK, 92, 93–97
- mplFreeWkspGLPK, 92, 93, 94–97
- mplGenerateGLPK, 92, 93, 94, 95–97
- mplPostsolveGLPK, 92–94, 95, 96, 97
- mplReadDataGLPK, 92–95, 96, 97
- mplReadModelGLPK, 92–96, 97
- MSG\_LEV (glpkConstants), 77
- NFS\_MAX (glpkConstants), 77
- NRS\_MAX (glpkConstants), 77
- OBJ\_LL (glpkConstants), 77
- OBJ\_UL (glpkConstants), 77
- ORD\_ALG (glpkConstants), 77
- OUT\_DLY (glpkConstants), 77
- OUT\_FRQ (glpkConstants), 77
- PIV\_LIM (glpkConstants), 77
- PIV\_TOL (glpkConstants), 77
- PP\_TECH (glpkConstants), 77
- PRESOLVE (glpkConstants), 77
- PRICING (glpkConstants), 77

- printIptGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
- printMIPGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
- printRangesGLPK, [100](#)
- printSolGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
  
- R\_TEST (glpkConstants), [77](#)
- readIptGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
- readLPGLPK, [103](#), [105](#), [106](#), [147](#), [149](#), [150](#)
- readMIPGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
- readMPSGLPK, [103](#), [105](#), [106](#), [147](#), [149](#), [150](#)
- readProbGLPK, [103](#), [105](#), [106](#), [147](#), [149](#), [150](#)
- readSolGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
- return\_codeGLPK, [83](#), [108](#), [137–140](#)
- RS\_SIZE (glpkConstants), [77](#)
  
- scaleProbGLPK, [108](#)
- setBfcpGLPK, [109](#)
- setColBndGLPK, [110](#), [113](#), [125](#)
- setColKindGLPK, [111](#), [115](#)
- setColNameGLPK, [112](#), [116](#)
- setColsBndsGLPK, [113](#), [114](#)
- setColsBndsObjCoefsGLPK, [114](#)
- setColsKindGLPK, [115](#)
- setColsNamesGLPK, [116](#)
- setColStatGLPK, [117](#)
- setDefaultIptParmGLPK, [118](#)
- setDefaultMIPParmGLPK, [118](#)
- setDefaultSmpParmGLPK, [119](#)
- setInteriorParmGLPK, [120](#)
- setMatColGLPK, [121](#)
- setMatRowGLPK, [122](#)
- setMIPParmGLPK, [123](#)
- setObjCoefGLPK, [124](#)
- setObjCoefsGLPK, [114](#), [125](#)
- setObjDirGLPK, [126](#)
- setObjNameGLPK, [127](#)
- setProbNameGLPK, [128](#)
- setRhsZeroGLPK, [129](#)
- setRiiGLPK, [129](#)
- setRowBndGLPK, [130](#), [132](#)
- setRowNameGLPK, [131](#), [133](#)
- setRowsBndsGLPK, [129](#), [132](#)
- setRowsNamesGLPK, [133](#)
  
- setRowStatGLPK, [134](#)
- setSimplexParmGLPK, [135](#)
- setSjjGLPK, [136](#)
- solveInteriorGLPK, [137](#)
- solveMIPGLPK, [138](#)
- solveSimplexExactGLPK, [139](#)
- solveSimplexGLPK, [140](#)
- sortMatrixGLPK, [141](#)
- status\_codeGLPK, [83](#), [141](#)
- stdBasisGLPK, [142](#)
- SUHL (glpkConstants), [77](#)
  
- termOutGLPK, [143](#)
- TM\_LIM (glpkConstants), [77](#)
- TOL\_BND (glpkConstants), [77](#)
- TOL\_DJ (glpkConstants), [77](#)
- TOL\_INT (glpkConstants), [77](#)
- TOL\_OBJ (glpkConstants), [77](#)
- TOL\_PIV (glpkConstants), [77](#)
- TYPE (glpkConstants), [77](#)
  
- unscaleProbGLPK, [144](#)
- UPD\_TOL (glpkConstants), [77](#)
  
- versionGLPK, [144](#)
  
- warmUpGLPK, [145](#)
- writeIptGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
- writeLPGLPK, [103](#), [105](#), [106](#), [147](#), [149](#), [150](#)
- writeMIPGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)
- writeMPSGLPK, [103](#), [105](#), [106](#), [147](#), [149](#), [150](#)
- writeProbGLPK, [103](#), [105](#), [106](#), [147](#), [149](#), [150](#)
- writeSolGLPK, [98](#), [99](#), [101](#), [102](#), [104](#), [107](#), [146](#), [148](#), [151](#)