The emme2 Package

February 16, 2008

Version 0.8

Date 2007-08-13

Title Read and Write to an EMME/2 databank

Author Ben Stabler <benstabler@yahoo.com>

Maintainer Ben Stabler <benstabler@yahoo.com>

Depends R (>= 1.6.0), graphics

Description This package includes functions to read and write to an EMME/2 databank

License GPL

R topics documented:

`emme2` *Read and Write to an EMME/2 databank*

Description

This package includes functions to read and write to an EMME/2 databank.

Usage

```
read.file0(bank)
read.file1(bank, file0)
read.matdir(bank, file0, mmat)
read.ms(bank, file0)
read.mo(numname, bank, file0, mcent, mat.dir)
read.md(numname, bank, file0, mcent, mat.dir)
```

```
read.mf(numname, bank, file0, mcent, mat.dir)
write.mf(data, numname, bank, file0, mcent, mmat, mat.dir, newname=NULL, newde
read.link.data(bank, scen.num, file0, mscen, mlink, mnode)
read.nodes(bank, scen.num, file0, mscen, mlink, mnode)
plotLinks(tofrom, nodes, title, ...)
ftnode(node.data, outgoing.links, jnode, mlink)
formatMf(data, file1)
get.emme2.time(timestamp)
```

Arguments

bank	String of the EMME/2 databank file name
file0	Databank metadata data frame
file1	Databank global and scenario parameters
mmat	Maximum number of matrices
numname	mf name as a string or mf number to read in or write to
newname	new name of the matrix to write out
newdesc	new description of the matrix to write out
mcent	Maximum number of centroids
mat.dir	matrix directory object
data	either a vector or matrix of data to write to bank
scen.num	scenario number to read from (in EMME/2 order - not named number)
mscen	Maximum number of scenarios
mlink	Maximum number of links
mnode	maximum number of nodes
link.data	EMME/2 link data.frame
nodes	EMME/2 nodes data.frame
tofrom	EMME/2 link data in from to node format
title	title for plot generated by plotLinks
node.data	EMME/2 nodes data.frame
outgoing.links	EMME/2 internal file 9 vector from read.link.data
jnode	EMME/2 internal file 11 vector from read.link.data
...	graphical parameters can be given as arguments to plot
timestamp	Sys.time()

Details

The EMME/2 databank stores dummy placeholder values for all the cells for all the matrices in a databank. Thus, if a matrix consists of 80x80 values and the databank has a maximum number of centroids of 100, then the databank is storing the 80x80 values in row-major order starting in the upper left corner and padding the remaining 20 "columns" with default values and the remaining 20

"rows" with default values. This is important since `read.mf` returns the full matrix - the matrix with the padding default values - and `write.mf` writes the full matrix - the data matrix plus the padded values. It is important then to call `formatMf` before `write.mf` in order to format the matrix that is to be written to the databank.

For details about the EMME/2 internal file structure refer to Appendix C of the EMME/2 User's Manual.

Steve Hansen <⟨Hansens@metro.dst.or.us⟩> helped with `read.mf` and `write.mf`

Brian Gregor <⟨Brian.J.GREGOR@odot.state.or.us⟩> helped with `read.link.data`

Value

read.file0	data.frame	EMME/2 internal file offsets
read.file1	list	EMME/2 global and scenario parameters
read.matdir	data.frame	EMME/2 matrix directory
read.ms	vector	EMME/2 all ms values
read.mo	vector	EMME/2 mo values
read.md	vector	EMME/2 md values
read.mf	matrix	EMME/2 mf values
write.mf	NA	Nothing returned
read.link.data	list	EMME/2 link data
read.nodes	data.frame	EMME/2 node data
plotLinks	NA	Plots EMME/2 network
ftnode	named numeric	EMME/2 link data in from to node format
formatMf	matrix	EMME/2 matrix with padded default values
get.emme2.time	integer	EMME/2 timestamp

Author(s)

Ben Stabler <⟨benstabler@yahoo.com⟩>

Examples

```
## Not run:
# Function call to create databank offset file0
file0 <- read.file0("emme2/emme2ban")

#Function call to create file1 info (global parameters)
file1 <- read.file1("emme2/emme2ban", file0)

#Function call to read matrix directory
mat.dir <- read.matdir("emme2/emme2ban", file0, file1$global["mmat"])

#Function call to read all ms from databank
```

```
ms <- read.ms("emme2/emme2ban", file0)

#Function call to read mo2
mo2 <- read.mo(2, "emme2/emme2ban", file0, file1$global["mcent"], mat.dir)

#Function call to read md2
md2 <- read.md(2, "emme2/emme2ban", file0, file1$global["mcent"], mat.dir)

#Function call to read mf2
mf2 <- read.mf(2, "emme2/emme2ban", file0, file1$global["mcent"], mat.dir)

#Function call to read mf "opskim"
mf2 <- read.mf("opskim", "emme2/emme2ban", file0, file1$global["mcent"], mat.dir)
mf2 <- mf2[zonesUsed,zonesUsed] #To crop the padded default values

#Function call to write mf2
x <- matrix(rnorm(mf2), nrow(mf2), ncol(mf2)) #Random generate length(mf) numbers to wr
x <- formatMf(x, file1) #Append the padded default values to the matrix
write.mf(x, 2, "emme2/emme2ban", file0, file1$global["mcent"], file1$global["mmat"], ma

#Function call to read link data
link.data <- read.link.data("emme2/emme2ban", 1, file0, file1$global["mscen"], file1$gl

#Function call to create from to node link table
tofrom <- ftnode(link.data[[1]], link.data[[2]], link.data[[3]], file1$global["mlink"])

#Function call to create node table
nodes <- read.nodes("emme2/emme2ban", 1, file0, file1$global["mscen"], file1$global["ml

#Function call to plot network
plotLinks(tofrom, nodes, "Network")

#Function call to format a mf to write to the databank
mf2 <- formatMf(mf2, file1)

## End(Not run)
```

Index