

# Package ‘PBSadmb’

May 13, 2013

**Version** 0.65

**Date** 2013-05-06

**Title** ADMB for R Using Scripts or GUI

**Author** Jon T. Schnute, Rowan Haigh, Alex Couture-Beil

**Maintainer** Jon T. Schnute <schnutej-dfo@shaw.ca>

**Depends** R (>= 2.15.0), methods, tcltk, PBSmodelling (>= 2.65)

**Description** R Support for ADMB (Automatic Differentiation Model Builder)

**License** GPL (>=2)

**URL** <http://code.google.com/p/pbs-admb/>

## R topics documented:

admb . . . . .	18
appendLog . . . . .	18
atget . . . . .	19
checkADopts . . . . .	20
cleanAD . . . . .	21
compAD . . . . .	21
convAD . . . . .	22
convOS . . . . .	23
copyFiles . . . . .	24
editAD . . . . .	25
editADfile . . . . .	25
installADMB . . . . .	26
linkAD . . . . .	26
makeAD . . . . .	27
makeADopts . . . . .	28
plotMC . . . . .	29
readADopts . . . . .	29
readRep . . . . .	30
runAD . . . . .	31
runMC . . . . .	32
setADMBPath . . . . .	33
showADargs . . . . .	33
startLog . . . . .	34
writeADopts . . . . .	35

**Index****36**

admb

*Start the PBS ADMB GUI***Description**

Start up the PBS GUI for running ADMB.

**Usage**

```
admb(prefix="", wdf="admbWin.txt", optfile="ADopts.txt")
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
wdf	string name of the <i>window description file</i> that creates the GUI.
optfile	string name of options file (usually in user's working directory).

**Author(s)**

Rowan Haigh, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[makeADopts](#)

appendLog

*Append Data to Log File***Description**

Append summary information or output to a previously created log file.

**Usage**

```
appendLog(prefix, lines)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
lines	data to append to 'prefix'.log).

**Value**

No explicit value returned. Appends data into a log file 'prefix'.log.

**Note**

A wrapper function that can be called from a GUI exists as `.win.appendLog`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[startLog](#), [editADfile](#)

---

atget

*Get/Print Objects From or Put Objects Into Temporary Work Environment*


---

**Description**

These functions are wrappers to the PBSmodelling accessor functions that get/print objects from or put objects into a temporary work environment, in this case `.PBSadmbEnv`. Working objects include `PBSadmb`, which acts as a storage object for some of the functions, and `.PBSadmb`, which controls the options for the user's project.

**Usage**

```
atget(...)
atcall(...)
atprint(...)
atput(...)
alisp(...)
```

**Arguments**

... For `atget` through to `atput`, the only free argument is:  
`x` – name (with or without quotes) of an object to retrieve or store in the temporary environment; cannot be represented by a variable.  
Fixed arguments: `penv = parent.frame()`, `tenv = .PBSadmbEnv`  
See [tget](#) for additional information.  
For `alisp`, there is only one fixed argument:  
`pos = .PBSadmbEnv`  
All other arguments are available – see [lisp](#)

**Details**

These accessor functions were developed as a response to the CRAN repository policy statement: “Packages should not modify the global environment (user's workspace).”

**Value**

Objects are retrieved from or sent to the temporary working environment to/from the place where the function(s) are called. Additionally, `atcall` invisibly returns the object without transferring, which is useful when the object is a function that the user may wish to call, for example, `atcall(myfunc)()`, or as arguments in other functions.

**Note**

Additional wrapper functions to access functions in `.PBSadmbEnv` are named with the prefix `.win` (none at the moment).

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**References**

CRAN Repository Policy: <http://cran.r-project.org/web/packages/policies.html>

**See Also**

[tget](#) and [lisp](#) in [PBSmodelling](#)

---

checkADopts

*Check ADMI Options for Link Integrity*


---

**Description**

Check that .ADopts has all required components and that links point to actual files on the hard drive.

**Usage**

```
checkADopts(opts=getOptions(atcall(.PBSadmb)),
  check=c("admbpath","gccpath","editor"), warn=TRUE, popup=FALSE)
```

**Arguments**

opts	ADMB options values.
check	components of .ADopts to check.
warn	logical: if TRUE, print the results of the check to the R console.
popup	logical: if TRUE, display program location problems in a popup GUI.

**Value**

Boolean value where TRUE indicates all programs were located in the specified directories and FALSE if at least one program cannot be found. The returned Boolean scalar has two attributes:

warn - named list of test results, and  
message - named vector of test results.

**Note**

A wrapper function that can be called from a GUI exists as .win.checkADopts.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Fisheries and Oceans Canada, Nanaimo BC

**See Also**

[makeADopts](#), [readADopts](#)

---

cleanAD

*Clean ADMB-Generated Files from the Working Directory*


---

**Description**

Detects files in the working directory with the specified `prefix` and removes them all save those with the suffix `.tpl`, `.dat`, and `.pin`.

**Usage**

```
cleanAD(prefix)
```

**Arguments**

`prefix`                      string name prefix of the ADMB project (e.g., "vonb").

**Details**

Aside from potential garbage files with the specified `prefix`, other files associated with ADMB are detected. Also files `*.tmp` and `*.bak` are displayed. Calling `cleanAD` invokes the hidden function `.cleanUp`, which creates a GUI menu of the potential garbage files. The user can select whichever files s/he wishes for disposal.

**Value**

Returns nothing. Invokes a GUI menu of potential garbage files.

**Note**

A wrapper function that can be called from a GUI exists as `.win.cleanAD`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[makeAD](#), [runAD](#), [readRep](#)

---

compAD

*Compile C Code*


---

**Description**

Compile C++ code in '`prefix`'.cpp to create a binary object file '`prefix`'.o.

**Usage**

```
compAD(prefix, ranef=FALSE, safe=TRUE, dll=FALSE, debug=FALSE, logfile=TRUE, add=TRUE, verbose=TRUE)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
raneff	logical: use the random effects model, otherwise use the normal model (currently does not influence the compile stage, but the argument is preserved here for future development).
safe	logical: if TRUE, use safe mode with bounds checking on all array objects, otherwise use optimized mode for fastest execution.
dll	create dll (rather than executable)
debug	compile with debug symbols
logfile	logical: if TRUE, create a log file of the messages from the shell call.
add	logical: if TRUE, append shell call messages to an existing log file.
verbose	logical: if TRUE, report the shell call and its messages to the R console.

**Details**

This function uses the C++ compiler declared in `.ADopts`. If `logfile=TRUE`, any errors will appear in `'prefix'.log`. If `verbose=TRUE`, they will appear in the R console.

**Value**

Invisibly returns the shell call and its messages.

**Note**

A wrapper function that can be called from a GUI exists as `.win.compAD`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[convAD](#), [linkAD](#), [makeAD](#)

---

convAD

---

*Convert TPL Code to CPP Code*


---

**Description**

Convert code in `'prefix'.tpl` to C++ code in `'prefix'.cpp`.

**Usage**

```
convAD(prefix, raneff=FALSE, safe=TRUE, dll=FALSE, debug=FALSE, logfile=TRUE, add=TRUE, verbose=TRUE)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
raneff	logical: if TRUE, use the random effects model executable tp12rem.exe, otherwise use the normal model executable tp12cpp.exe.
safe	logical: if TRUE, use safe mode with bounds checking on all array objects, otherwise use optimized mode for fastest execution.
dll	create dll (rather than executable)
debug	compile with debug symbols
logfile	logical: if TRUE, create a log file of the messages from the shell call.
add	logical: if TRUE, append shell call messages to an existing log file.
verbose	logical: if TRUE, report the shell call and its messages to the R console.

**Details**

This function invokes the ADMB command tp12cpp.exe or tp12rem.exe, if raneff is FALSE or TRUE respectively. If logfile=TRUE, any errors will appear in 'prefix'.log. If verbose=TRUE, they will appear in R console.

**Value**

Invisibly returns the shell call and its messages.

**Note**

A wrapper function that can be called from a GUI exists as .win.convAD.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[compAD](#), [linkAD](#), [makeAD](#)

---

convOS

---

*Convert Text Files to Default OS Format*


---

**Description**

Convert text files to the default format of the operating system.

**Usage**

```
convOS(inam, onam = inam, path = getwd() )
```

**Arguments**

inam	string vector of names specifying files to be converted to the format of the operating system.
onam	string vector of name specifying the output files (the default overwrites the input file).
path	string specifying the path where the input files are located (defaults to current working directory).

**Value**

Text file(s) formatted in accordance with standards of the operating system.

**Note**

This function essentially executes a `readLines` command followed by a call to `writeLines`.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Nanaimo BC

**See Also**

[copyFiles](#), `.addQuotes`

---

copyFiles

*Copy System Files*

---

**Description**

Copy files with specified prefixes and suffixes from one location to another.

**Usage**

```
copyFiles(prefix, suffix=NULL, srcdir=getwd(), dstdir=getwd(), ask=TRUE)
```

**Arguments**

<code>prefix</code>	string scalar/vector of potential file prefixes.
<code>suffix</code>	string scalar/vector of potential file suffixes.
<code>srcdir</code>	source directory from which to copy files.
<code>dstdir</code>	destination directory to copy files to.
<code>ask</code>	logical: if TRUE, popup boxes will prompt the user for every instance that a file will be over-written.

**Details**

This function uses R's `list.files` and `file.copy` functions. The pattern recognition tends not to work when given the wildcard character `*`; however, the user may use this character, and the code will interpret it.

**Value**

Invisibly returns a Boolean vector with names of files that have been copied or not.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Nanaimo, BC

**See Also**

[editAD](#)



editAD

*Edit ADMB Files***Description**

Edit files associated with specified prefix and suffixes.

**Usage**

```
editAD(prefix, suffix=c(".tpl", ".cpp", ".log"))
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
suffix	string scalar/vector specifying one or more suffixes.

**Value**

Invisibly returns Boolean vector with elements TRUE if files exist, FALSE if they do not.

**Note**

A wrapper function that can be called from a GUI exists as `.win.editAD`.

This function explicitly uses the editor chosen for PBSadmb. PBSmodelling has another function `openFile` that uses Windows file associations or an application specified with `setPBSext`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[editADfile](#), [makeADopts](#)

editADfile

*Edit a File***Description**

Edit a file using the text editor specified in `.ADopts`.

**Usage**

```
editADfile(fname)
```

**Arguments**

fname	string name of file in current working directory (or elsewhere if path delimited by / or \).
-------	--

**Value**

Returns Boolean: TRUE if file exists, FALSE if it does not.

**Note**

This function explicitly uses the editor chosen for PBSadmb. PBSmodelling has another function `openFile` that uses Windows file associations or an application specified with `setPBSext`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[editAD](#), [makeADopts](#)

---

installADMB

*Install windows ADMB and MinGW binaries*

---

**Description**

Only applicable for Windows: Downloads and installs the windows ADMB and MinGW binaries.

A user interface is displayed which allows to select to install either ADMB and/or MinGW for either 32bit or 64bit versions of Windows. Default installation directories are suggested for installation; however, a user may choose to install the binaries elsewhere.

A file "pathconfig.txt" is saved in the PBSadmb library (under R), which keeps track of the most recently installed locations.

**Usage**

```
installADMB()
```

---

linkAD

*Link Object Files to Make an Executable*

---

**Description**

Links the binary object file 'prefix'.o to the ADMB libraries and produces the executable file 'prefix'.exe.

**Usage**

```
linkAD(prefix, raneff=FALSE, safe=TRUE, dll=FALSE, debug=FALSE, logfile=TRUE, add=TRUE, verbose=TRUE)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
raneff	logical: use the random effects model, otherwise use the normal model.
safe	logical: if TRUE, use safe mode with bounds checking on all array objects, otherwise use optimized mode for fastest execution.
dll	create dll (rather than executable)
debug	compile with debug symbols
logfile	logical: if TRUE, create a log file of the messages from the shell call.
add	logical: if TRUE, append shell call messages to an existing log file.
verbose	logical: if TRUE, report the shell call and its messages to the R console.

**Details**

This function uses the C++ compiler declared in `.ADopts`. If `logfile=TRUE`, any errors will appear in `'prefix'.log`. If `verbose=TRUE`, they will appear in the R console.

**Value**

Invisibly returns the shell call and its messages.

**Note**

A wrapper function that can be called from a GUI exists as `.win.linkAD`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[convAD](#), [compAD](#), [makeAD](#)

---

makeAD

---

*Make an Executable Binary File from a C File*


---

**Description**

Essentially a wrapper function that calls in sequence: `convAD`, `compAD`, and `linkAD`.

**Usage**

```
makeAD(prefix, ranef=FALSE, safe=TRUE, dll=FALSE, debug=FALSE, logfile=TRUE, add=TRUE, verbose=TRUE)
```

**Arguments**

<code>prefix</code>	string name prefix of the ADMB project (e.g., "vonb").
<code>ranef</code>	logical: use the random effects model, otherwise use the normal model.
<code>safe</code>	logical: if TRUE, use safe mode with bounds checking on all array objects, otherwise use optimized mode for fastest execution.
<code>dll</code>	create dll (rather than executable)
<code>debug</code>	compile with debug symbols
<code>logfile</code>	logical: if TRUE, create a log file of the messages from the shell call.
<code>add</code>	logical: if TRUE, append shell call messages to an existing log file.
<code>verbose</code>	logical: if TRUE, report the shell call and its messages to the R console.

**Details**

This function uses the C++ compiler declared in `.ADopts`. If `logfile=TRUE`, any errors will appear in `'prefix'.log`. If `verbose=TRUE`, they will appear in the R console.

**Value**

Returns nothing. The three functions called by `makeAD` each return the shell call and its messages.

**Note**

A wrapper function that can be called from a GUI exists as `.win.makeAD`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[convAD](#), [compAD](#), [linkAD](#), [cleanAD](#)

---

makeADopts

*Creates the ADMB Options List*


---

**Description**

Creates a global list object detailing the pathways to the ADMB directory, the GCC bin, and the user's preferred text editor.

**Usage**

```
makeADopts(admbpath, gccpath, editor)
```

**Arguments**

admbpath	explicit path to the user's ADMB directory.
gccpath	explicit path to the user's GCC bin (C-compiler) directory.
editor	explicit path and program to use for editing text.

**Value**

Creates a global, hidden list object called `.ADopts`.

**Note**

A wrapper function that can be called from a GUI exists as `.win.makeADopts`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[makeADopts](#), [writeADopts](#)

plotMC

*Plot Results of MCMC Simulation***Description**

Plot results of an ADMB MCMC simulation using various plot methods.

**Usage**

```
plotMC(prefix, act="pairs", pthin=1, useCols=NULL)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
act	string scalar: action describing plot type (current choices: "pairs", "eggs", "acf", "trace", and "dens").
pthin	numeric scalar indicating interval at which to collect records from the .mc.dat file for plotting.
useCols	logical vector indicating which columns of .mc.dat to plot.

**Note**

A wrapper function that can be called from a GUI exists as `.win.plotMC`. Use the PBSadmb GUI to explore these plots easily.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[runMC](#), [showADargs](#)

readADopts

*Reads an ADMB Options List into Memory From a File***Description**

Reads ADMB options into a global, hidden list object called `.ADopts` from an ASCII text file using `PBSmodelling::readList`.

**Usage**

```
readADopts(optfile="ADopts.txt")
```

**Arguments**

optfile	string name of an ASCII text file containing ADMB options information.
---------	--

**Value**

No values returned. Reads the ADMB options into the list object `.ADopts`.

**Note**

A wrapper function that can be called from a GUI exists as `.win.readADopts`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[makeADopts](#), [writeADopts](#)

---

readRep

*Read an ADMB Report into R Memory*


---

**Description**

Import ADMB-generated report files into R's memory using the names of the report files to name the R-objects.

**Usage**

```
readRep(prefix, suffix=c(".cor", ".rep", ".std", ".mc.dat"), global=FALSE)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
suffix	string scalar/vector specifying one or more suffixes.
global	logical: if TRUE, save the imported reports as objects to global environment using the same names as the report files.

**Details**

If the report object is one of `c(".cor", ".std", ".mc.dat")`, the report object is a data frame, otherwise it is a string vector. Multiple report objects are returned as a list of objects. A single report object is returned as the object itself.

This function attempts to detect the file format from a number of possibilities. For example, if the file has the special format recognized by `PBSmodelling`, then the function returns a list with named components. The example `vonb` included with this package shows how to write the template to get consistent variable names between ADMB and R. See the User's Guide for complete details.

**Value**

Invisibly returns the list of report objects. If only one report is imported, a single report object is returned.

**Note**

A wrapper function that can be called from a GUI exists as `.win.readRep`.

**Author(s)**

Rowan Haigh, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[editADfile](#), `.win.viewRep`

---

`runAD`*Run an Executable Binary File*

---

## Description

Run the executable binary file 'prefix'.exe that was created by makeAD.

## Usage

```
runAD(prefix, argvec="", logfile=TRUE, add=TRUE, verbose=TRUE)
```

## Arguments

<code>prefix</code>	string name prefix of the ADMB project (e.g., "vonb").
<code>argvec</code>	string scalar/vector of arguments appropriate for the executable 'prefix'.exe.
<code>logfile</code>	logical: if TRUE, create a log file of the messages from the shell call.
<code>add</code>	logical: if TRUE, append shell call messages to an existing log file.
<code>verbose</code>	logical: if TRUE, report the shell call and its messages to the R console.

## Details

This function typically reads the two files 'prefix'.dat and 'prefix'.pin, although in some cases one or both of these files may not be necessary.

If `logfile=TRUE`, output (including error messages, if any) will appear in 'prefix'.log. If `verbose=TRUE`, it will appear in the R console.

## Value

Invisibly returns the results of the shell call.

## Note

A wrapper function that can be called from a GUI exists as `.win.runAD`.

## Author(s)

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

## See Also

[runMC](#), [makeAD](#), [cleanAD](#)

runMC

*Run an Executable Binary File in MCMC Mode***Description**

Run the executable binary file 'prefix'.exe, created by makeAD, to generate MCMC simulations.

**Usage**

```
runMC(prefix, nsims=2000, nthin=20, outsuff=".mc.dat",
      logfile=FALSE, add=TRUE, verbose=TRUE)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
nsims	numeric scalar indicating number of MCMC simulations to perform.
nthin	numeric scalar indicating the sampling rate or thinning of the nsims MCMC simulations to report.
outsuff	string name suffix of the MCMC output data file.
logfile	logical: if TRUE, create a log file of the messages from the shell call.
add	logical: if TRUE, append shell call messages to an existing log file.
verbose	logical: if TRUE, report the shell call and its messages to the R console.

**Details**

This function runs 'prefix'.exe twice, first with the arguments -mcmc 'nsims' -mcsave 'nthin' and second with the argument -mceval. By default, output goes to the file 'prefix'.mc.dat, although a user can specify a different output suffix.

To see this function in action, use the PBSadmb GUI with the example vonb or simpleMC.

**Value**

Invisibly returns the results of the shell call.

**Note**

A wrapper function that can be called from a GUI exists as .win.runMC.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[runAD](#), [makeAD](#), [cleanAD](#)



---

setADMBPath	<i>Creates the ADMB Options List</i>
-------------	--------------------------------------

---

**Description**

Creates a global list object detailing the pathways to the ADMB directory, the GCC bin, and the user's preferred text editor.

**Usage**

```
setADMBPath(admbpath, gccpath, editor)
```

**Arguments**

admbpath	explicit path to the user's ADMB directory.
gccpath	explicit path to the user's GCC bin (C-compiler) directory.
editor	explicit path and program to use for editing text.

**Value**

Creates a global, hidden list object called `.ADopts`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[makeADopts](#), [writeADopts](#)

---

showADargs	<i>Show All Arguments for an ADMB Executable</i>
------------	--

---

**Description**

Show all arguments available for an ADMB executable in the default text editor.

**Usage**

```
showADargs(prefix, ed=TRUE)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
ed	logical: if TRUE, write the ADMB arguments to a file and view them with the text editor, else display the arguments on the R console.

**Value**

Invisibly returns the argument list.

**Note**

A wrapper function that can be called from a GUI exists as `.win.showADargs`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[editADfile](#), [runAD](#)

---

startLog	<i>Start a Log File</i>
----------	-------------------------

---

**Description**

Start a log file by removing any previous version and appending header information.

**Usage**

```
startLog(prefix)
```

**Arguments**

prefix	string name prefix of the ADMB project (e.g., "vonb").
--------	--

**Value**

No explicit value returned. Writes header lines into a log file `'prefix'.log`.

**Note**

A wrapper function that can be called from a GUI exists as `.win.startLog`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[appendLog](#), [editADfile](#)

---

`writeADopts`*Writes the ADMB Options List from Memory to a File*

---

**Description**

Writes the global ADMB options list to a file in 'PBS' format (see `PBSmodelling::writeList`).

**Usage**

```
writeADopts(optfile="ADopts.txt")
```

**Arguments**

`optfile`                string name of the intended output file.

**Value**

Returns `opts` invisibly. Writes the options list object to an ASCII file.

**Note**

A wrapper function that can be called from a GUI exists as `.win.writeADopts`.

**Author(s)**

Jon T. Schnute, Pacific Biological Station, Nanaimo BC, Canada

**See Also**

[makeADopts](#), [readADopts](#)

# Index

## \*Topic **IO**

admb, [18](#)  
copyFiles, [24](#)

## \*Topic **character**

convOS, [23](#)

## \*Topic **data**

checkADopts, [20](#)  
makeADopts, [28](#)  
readADopts, [29](#)  
setADMBPath, [33](#)  
writeADopts, [35](#)

## \*Topic **environment**

atget, [19](#)

## \*Topic **file**

appendLog, [18](#)  
convOS, [23](#)  
editAD, [25](#)  
editADfile, [25](#)  
installADMB, [26](#)  
readRep, [30](#)  
showADargs, [33](#)  
startLog, [34](#)

## \*Topic **hplot**

plotMC, [29](#)

## \*Topic **interface**

compAD, [21](#)  
convAD, [22](#)  
linkAD, [26](#)  
makeAD, [27](#)  
runAD, [31](#)  
runMC, [32](#)

## \*Topic **list**

checkADopts, [20](#)  
makeADopts, [28](#)  
readADopts, [29](#)  
setADMBPath, [33](#)  
writeADopts, [35](#)

## \*Topic **manip**

atget, [19](#)  
cleanAD, [21](#)  
readRep, [30](#)

## \*Topic **programming**

compAD, [21](#)  
convAD, [22](#)  
linkAD, [26](#)

makeAD, [27](#)

runAD, [31](#)

runMC, [32](#)

## \*Topic **utilities**

copyFiles, [24](#)

admb, [18](#)

alisp (atget), [19](#)

appendLog, [18](#), [34](#)

atcall (atget), [19](#)

atget, [19](#)

atprint (atget), [19](#)

atput (atget), [19](#)

checkADopts, [20](#)

cleanAD, [21](#), [28](#), [31](#), [32](#)

compAD, [21](#), [23](#), [27](#), [28](#)

convAD, [22](#), [22](#), [27](#), [28](#)

convOS, [23](#)

copyFiles, [24](#), [24](#)

editAD, [24](#), [25](#), [26](#)

editADfile, [19](#), [25](#), [25](#), [30](#), [34](#)

installADMB, [26](#)

linkAD, [22](#), [23](#), [26](#), [28](#)

lisp, [19](#), [20](#)

makeAD, [21–23](#), [27](#), [27](#), [31](#), [32](#)

makeADopts, [18](#), [20](#), [25](#), [26](#), [28](#), [28](#), [30](#), [33](#), [35](#)

plotMC, [29](#)

readADopts, [20](#), [29](#), [35](#)

readRep, [21](#), [30](#)

runAD, [21](#), [31](#), [32](#), [34](#)

runMC, [29](#), [31](#), [32](#)

setADMBPath, [33](#)

showADargs, [29](#), [33](#)

startLog, [19](#), [34](#)

tget, [19](#), [20](#)

writeADopts, [28](#), [30](#), [33](#), [35](#)