

BBSReadRexx.adoc

COLLABORATORS

	<i>TITLE :</i> BBSReadRexx.adoc		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		April 17, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	BBSReadRexx.adoc	1
1.1	BBSREAD	1
1.2	BBSREAD/--background--	2
1.3	BBSREAD/AMIGA2DATE	3
1.4	BBSREAD/APPENDPASSCONFLIST	3
1.5	BBSREAD/BUFFREEALLOCATED	4
1.6	BBSREAD/BUFHITRATE	5
1.7	BBSREAD/BUFMODE	6
1.8	BBSREAD/CONFIGARC	6
1.9	BBSREAD/CONFIGBBS	8
1.10	BBSREAD/CONFIGCONF	10
1.11	BBSREAD/CONFIGFAREA	12
1.12	BBSREAD/CONFIGGLOBAL	14
1.13	BBSREAD/DATE2AMIGA	16
1.14	BBSREAD/EXTERNALBBSCONFIG	16
1.15	BBSREAD/FINDDUPBRMSG	17
1.16	BBSREAD/GETARCDATA	18
1.17	BBSREAD/GETBBSDATA	19
1.18	BBSREAD/GETBBSLIST	21
1.19	BBSREAD/GETCOMMANDINFO	22
1.20	BBSREAD/GETCONFDATA	23
1.21	BBSREAD/GETCONFLIST	25
1.22	BBSREAD/GETFAREADATA	26
1.23	BBSREAD/GETFAREALIST	27
1.24	BBSREAD/GETGLOBALDATA	28
1.25	BBSREAD/PACKDATAFILE	29
1.26	BBSREAD/READARGS	30
1.27	BBSREAD/READBREVENT	32
1.28	BBSREAD/READBRFILE	34
1.29	BBSREAD/READBRMESSAGE	36

1.30	BBSREAD/READBRUSER	41
1.31	BBSREAD/READPASSCONFLIST	43
1.32	BBSREAD/SEARCHBRFILE	44
1.33	BBSREAD/SEARCHBRUSER	45
1.34	BBSREAD/SORTBBSLIST	47
1.35	BBSREAD/SORTCONFLIST	47
1.36	BBSREAD/SORTFAREALIST	48
1.37	BBSREAD/UNIQUEMSGFILE	48
1.38	BBSREAD/UPDATEBREVENT	50
1.39	BBSREAD/UPDATEBRMESSAGE	51
1.40	BBSREAD/VERSION	52
1.41	BBSREAD/WRITEBREVENT	53
1.42	BBSREAD/WRITEBRFILE	56
1.43	BBSREAD/WRITEBRIEFMSG	57
1.44	BBSREAD/WRITEBRMESSAGE	58
1.45	BBSREAD/WRITEBRUSER	62
1.46	BBSREAD/WRITEPASSCONFLIST	63

Chapter 1

BBSReadRexx.adoc

1.1 BBSREAD

```
--background--  
  
AMIGA2DATE  
  
APPENDPASSCONFLIST  
  
BUFFREEALLOCATED  
  
BUFHITRATE  
  
BUFMODE  
  
CONFIGARC  
  
CONFIGBBS  
  
CONFIGCONF  
  
CONFIGFAREA  
  
CONFIGGLOBAL  
  
DATE2AMIGA  
  
EXTERNALBBSCONFIG  
  
FINDDUPBRMSG  
  
GETARCDATA  
  
GETBBSDATA  
  
GETBBSLIST  
  
GETCOMMANDINFO  
  
GETCONFDATA
```

GETCONFLIST
GETFAREADATA
GETFAREALIST
GETGLOBALDATA
PACKDATAFILE
READARGS
READBREVENT
READBRFILE
READBRMESSAGE
READBRUSER
READPASSCONFLIST
SEARCHBRFILE
SEARCHBRUSER
SORTBBSLIST
SORTCONFLIST
SORTFAREALIST
UNIQUEMSGFILE
UPDATEBREVENT
UPDATEBRMESSAGE
VERSION
WRITEBREVENT
WRITEBRFILE
WRITEBRIEFMSG
WRITEBRMESSAGE
WRITEBRUSER
WRITEPASSCONFLIST

1.2 BBSREAD/--background--

NOTES

All functions sets the BBSREAD.LASTERROR variable on error. It contains a description of why the function returned a error code.

All funtions returns a error level of 30 on failure. The reason for the failure is described in BBSREAD.LASTERROR.

1.3 BBSREAD/AMIGA2DATE

NAME

AMIGA2DATE -- Fill in a ClockData stem based on a system time stamp.

SYNOPSIS

AMIGA2DATE SECONDS/N/A,STEM/A

FUNCTION

Returns a ClockData stem with the date and time calculated from the SECONDS parameter. The ClockData information is returned in stem variables. The stem variables returned are:

<stem>.SEC
<stem>.MIN
<stem>.HOUR
<stem>.MDAY
<stem>.MONTH
<stem>.YEAR
<stem>.WDAY

INPUTS

SECONDS - Time stamp to convert. Seconds since 1 january 1978.

STEM - Base name of the stem variables the ClockData information is returned in.

RESULT

EXAMPLE

NOTES

BUGS

SEE ALSO

1.4 BBSREAD/APPENDPASSCONFLIST

NAME

APPENDPASSCONFLIST -- Append list to passive conference list.

SYNOPSIS

APPENDPASSCONFLIST BBSNAME/A,STEM/A

FUNCTION

Appends a list of conference names and descriptions to the passive conference list of a bbs. There will be no checking for duplicates.

The list of conferences are passed in stem variables. The list is expected to be in the following stem variables:

```
<stem>.COUNT - See
                WRITEPASSCONFLIST
                for stem descriptions.
<stem>.<l <= n <= count>.NAME
<stem>.<l <= n <= count>.DESCR
<stem>.<l <= n <= count>.BBSCONFNR
<stem>.<l <= n <= count>.FLAGS
```

INPUTS

BBSNAME - Name of bbs.

STEM - Base name for the stem variables used to pass the conference list.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 30 - Failure.

EXAMPLE

```
address "BBSREAD"
```

```
CDF_MAIL = '00000002'x /* Private mail conference. */
```

```
LIST.1.NAME = 'QuickTools/Beta'
LIST.1.DESCR = 'QuickTools beta testing conference'
LIST.2.NAME = 'Generelt'
LIST.3.NAME = 'Post'
LIST.3.BBSCONFNR = 0
LIST.3.FLAGS = CDF_MAIL
LIST.COUNT = 3
```

```
APPENDPASSCONFLIST bbsname '"Ultima"' stem LIST
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

NOTES**BUGS****SEE ALSO**

1.5 BBSREAD/BUFFFREEALLOCATED

NAME

BUFFFREEALLOCATED - Frees most memory allocated for buffers.

SYNOPSIS
BUFFREEALLOCATED

FUNCTION
This command frees most memory allocated for buffers.

INPUTS

RESULT

EXAMPLE

NOTES

BUGS

SEE ALSO

1.6 BBSREAD/BUFHITRATE

NAME
BUFHITRATE - Return buffer hitrate.

SYNOPSIS
BUFHITRATE STEM,CLEAR/S

FUNCTION
Returns the global hitrate per cent for the buffer system in the RESULT variable. If the STEM parameter is given, is the read and write hitrate be returned in stem variables. The data returned in stem variables are:
<stem>.READ - The read hitrate per cent.
<stem>.WRITE - The write hitrate per cent.

INPUTS
STEM - Base name of the stem variables to put data in. Must be uppercase.
CLEAR - Clear buffer hit/miss counters after getting hitrate.

RESULT
Returns: 0 - Success. The RESULT variable is set to the global hit rate per cent.
30 - Failure.

EXAMPLE
options results
address BBSREAD

BUFHITRATE RATES

```
say 'Global hitrate:' result
say 'Read hitrate: ' RATES.READ
say 'Write hitrate: ' RATES.WRITE
```

exit

NOTES

BUGS

SEE ALSO

1.7 BBSREAD/BUFMODE

NAME

BUFMODE -- Set buffer modes.

SYNOPSIS

BUFMODE MODE/A

FUNCTION

Set buffer mode. In copyback mode are changed buffers only written back to to file when they are flushed from memory. Each 'BUFMODE COPYBACK' *must* be coupled with a 'BUFMODE ENDCOPYBACK'. The copyback mode should be used while adding large ammount of data to the library. (Eg. parsing a grabfile and adding the messages to the database.) BUFMODE commands can be nested.

INPUTS

MODE - Mode commands:

- o COPYBACK - Turn on copyback mode.
- o ENDCOPYBACK - Turn off copyback mode.

RESULT

Returns: 0 - Success.
10 - Unknown mode command.
30 - Failure.

EXAMPLE

address BBSREAD

```
BUFMODE COPYBACK      /* Turn on copyback mode */
```

```
BUFMODE ENDCOPYBACK  /* Turn off copyback mode */
```

NOTES

BUGS

SEE ALSO

1.8 BBSREAD/CONFIGARC

NAME

CONFIGARC - Configure archive types.

SYNOPSIS

```
CONFIGARC ARCHIVE/A,PATTERN=PAT/K,UNARCHIVECOMMAND=UNARC/K,
  ARCHIVECOMMAND=ARC/K,DELETE/S,USE/S,LASTSAVED/S
```

FUNCTION

Command to configure archive types for use with bbsread.library. The archivers are expected to use the standard archiver organization of the arguments.

INPUTS

ARCHIVER - Name of archive type to configure.
 PATTERN - Standard AmigaDOS matchpattern to recognize archives of this type. If the pattern begins with '\$', is the pattern matched with the contents of the archive instead of the filename. Eg. \$????2d6c68 finds LhA-archives.
 UNARCHIVECOMMAND - Command to depack an archive of this type.
 ARCHIVECOMMAND - Command to pack an archive of this type.
 DELETE - Delete archive type. The deletion will fail if one bbstype or bbs uses this archiver as an eventarchiver.
 USE - Don't save the changes done to the configuration. Is ignored when deleting or creating new archive types.
 LASTSAVED - Will retrieve the configuration last saved before applying any changes.

RESULT

Returns: 0 - Success.
 30 - Failure.

EXAMPLE

address BBSREAD

```
/* Configure Arj archive type */
```

```
CONFIGARC archive 'Arj' pat '#?.arj' arc '"Arj a"',
  unarc '"UnArj e -y"'
if(rc ~= 0) then
do
  say BBSREAD.LASTERORR
  exit
end
```

```
/* Delete Arc archive type */
```

```
CONFIGARC archive 'Arc' DELETE
if(rc ~= 0) then
do
  say BBSREAD.LASTERORR
  exit
end
```

NOTES

If the archive/unarchive commands aren't in the dos search path must

the full path to the commands also be given.

Using the LASTSAVED switch will also load the last saved global configuration. If you have used the USE switch with

```
CONFIGGLOBAL
```

is the global configuration saved if you don't use the USE switch with CONFIGARC.

BUGS

SEE ALSO

1.9 BBSREAD/CONFIGBBS

NAME

CONFIGBBS -- Configure bbs.

SYNOPSIS

```
CONFIGBBS BBSNAME/A, NEWNAME/K, DELETEBBS/S, GRABNAME=GRAB/K,
  BBSTYPE=TYPE/K, EMAILADDR=EMAIL/K, USERNAME=USER/K,
  USERSTREET=STREET/K, USERADDRESS=ADDR/K, USERCOUNTRY=COUNTRY/K,
  USERPHONE=PHONE/K, SCRIPTFLAGS=SF/K, SIGNATURE=SIG/K,
  SETFLAGS=SET/K, CLEARFLAGS=CLEAR/K, KEEPMSG/K/N, KEPTIME/K/N,
  CHARSET/N/K, XPKMETHOD=XPK/K, LINELENGTH=LL/K/N, ALIAS/K,
  DNLOADPATH=DNLOAD/K, TAGFILE=TAG/K, EVENTARCHIVER=ARC/K,
  REPLYPACKET=REP/K, UPLOADPATH=UPLOAD/K, REPLYSTRING=REPSTR/K,
  QUOTETYPE=QT/N/K, QUOTECHARS=QC/K, BBSENTERSCRIPT=BBSSENT/K,
  BBSLEAVESCRIPT=BBSLEAVE/K, CONFENTERSCRIPT=CONFENT/K,
  CONFLEAVESCRIPT=CONFLEAVE/K, TOP/S, BOTTOM/S, UP/S, DOWN/S
```

FUNCTION

Command to configure bbses. BBSes can be created, changed, deleted, and individually rearranged.

INPUTS

BBSNAME	- Name of bbs to configure. If this bbs doesn't exist, is a bbs with this name created.
NEWNAME	- Change name of bbs to this.
DELETEBBS	- Delete bbs.
GRABNAME	- Name of grabfile (without extension).
BBSTYPE	- Type of bbs.
EMAILADDR	- The email address the user has on the bbs.
USERNAME	- The name the user is registered as.
USERSTREET	- User street address. Overrides global setting.
USERADDRESS	- User address. Overrides global setting.
USERCOUNTRY	- User country. Overrides global setting.
USERPHONE	- User phone number. Overrides global setting.
SCRIPTFLAGS	- Flags for script. See rexx/examples/GlobalDefs.br for definitions. Flags must be given as a hex string.
SIGNATURE	- Signature to use on this bbs. Overrides global settings.
SETFLAGS	- BBS flags to set. See rexx/examples/GlobalDefs.br for definitions. Flags must be given as a hex string.

CLEARFLAGS - BBS flags to clear.
 KEEPMSG - Messages to keep in conferences.
 KEEPTIME - Time (in seconds) to keep messages in conferences.
 CHARSET - Charset for grab from this BBS uses. Overrides bbs type if ~= BRCS_ANY.
 XPKMETHOD - Xpk method to use when packing this bbs. Overrides global settings.
 LINELENGTH - Max linelength in messages. Overrides bbs type if > 0.
 ALIAS - Alias used in conferences marked with CDF_ALIAS.
 DNLOADPATH - Dnloadpath for this bbs. Overrides global settings.
 TAGFILE - Tagfile to use on this BBS. Overrides global settings.
 EVENTARCHIVER - Archiver to use when packing the events from this bbs. Overrides bbs type.
 REPLYPACKET - File name of eventual reply packet. The file name is expected to be relative to the upload directory defined for this BBS. (Should only be set by parsers)
 UPLOADPATH - Uploadpath for this bbs. Overrides global settings.
 REPLYSTRING - Reply string to use when a message is replied and moved. Overrides global settings.
 QUOTETYPE - Preferred quote type for this bbs. See rexx/examples/GlobalDefs.br for definitions. Overrides bbs type setting.
 QUOTECHARS - Quote chars in custom quote type. (Max length is 3.) Overrides global settings.
 BBSENTERSCRIPT - Arexx script run when a bbs is entered.
 BBSLEAVESCRIPT - Arexx script run when a bbs is left.
 CONFENTERSCRIPT - Arexx script run when a conf is entered.
 CONFLEAVESCRIPT - Arexx script run when a conf is left.
 Above script settings overrides global settings.
 TOP - Move bbs to the top of the list.
 BOTTOM - Move bbs to the bottom of the list.
 UP - Move bbs one step up on the list.
 DOWN - Move bbs one step down on the list.

Strings which overrides global and type settings does not override if they are empty.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 30 - Failure.

EXAMPLE

```
address "BBSREAD"
```

```
/* Move bbs "Ultima" to the top of the list */
```

```

CONFIGBBS bbsname ' "Ultima" ' TOP
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end

```

```

/* Set username to "Eivind Nordseth" */

CONFIGBBS bbsname ' "Ultima" ' USERNAME ' "Eivind Nordseth" '
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end

/* Set the BDF_NO_TAGS flag */

BDF_NO_TAGS = '00000100'x
CONFIGBBS bbsname ' "Ultima" ' SET c2x(BDF_NO_TAGS)
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end

exit

```

NOTES

The result of combining TOP, BOTTOM, UP and DOWN in the same call is **NOT** guaranteed to be the same on future versions.

TOP, BOTTOM, UP, DOWN and DELETEBBS can only be used when the bbs given in BBSNAME exists.

To remove string settings which overrides global and type settings set the string to ''.

When creating new bbses you must fill in the BBSTYPE and GRABNAME parameters.

BUGS

SEE ALSO

<rex/examples/GlobalDefs.br>

1.10 BBSREAD/CONFIGCONF

NAME

CONFIGCONF -- Configure conference.

SYNOPSIS

```

CONFIGCONF BBSNAME/A, CONFNAME/A, NEWNAME/K, DELETECONF/S,
  SETFLAGS=SET/K, CLEARFLAGS=CLEAR/K, SIGNATURE=SIG/K, KEEPMSG/K/N,
  KEEPTIME/K/N, ALIAS/K, BBSCONFNR=NR/K/N, XPKMETHOD=XPK/K,
  CHARSET/N/K, LINELENGTH=LL/K/N, TAGFILE=TAG/K, EMAILADDR=EMAIL/K,
  REPLYSTRING=REPSTR/K, QUOTETYPE=QT/N/K, QUOTECHARS=QC/K,
  ENTERSCRIPT=ENT/K, LEAVESCRIPT=LEAVE/K, CONFNETTYPE=NET/K/N,
  TOP/S, BOTTOM/S, UP/S, DOWN/S

```

FUNCTION

Command to configure conferences. Conferences can be created, changed, deleted and individually rearranged.

See

GETCONFDATA
stem definitions for more details on parameters.

INPUTS

BBSNAME - Name of the bbs the conference to configure is on.
CONFNAME - Name of conference to configure. If this conference doesn't exist, is a conference with this name created.
NEWNAME - Change name of conference to this.
DELETECONF - Delete conference.
SETFLAGS - Conference flags to set. See rexx/examples/GlobalDefs.br for definitions. Flags must be given as a hex string.
CLEARFLAGS - Conference flags to clear. Flags must be given as a hex string.
SIGNATURE - Signature to use in this conference. Overrides bbs settings.
KEEPMSG - Messages to keep in this conference.
KEEPTIME - Time (in seconds) to keep messages in this conference.
ALIAS - Alias used in conference if marked with CDF_ALIAS.
BBSCONFNR - The internal number the conference has on the bbs.
XPKMETHOD - What xpk method to use when packing this conference. Overrides bbs settings.
CHARSET - Charset the grab from this bbs uses in this conference. Overrides bbs settings if ~= BRCS_ANY. See rexx/examples/GlobalDefs.br for definitions.
LINELENGTH - Max linelength in messages. Overrides bbs settings if > 0.
TAGFILE - Tagfile to use in this conference. Overrides bbs settings.
EMAILADDR - The Email address the user has in this conference. For use on bbses where the user is member of more than one net.
REPLYSTRING - Reply string to use when a message is replied and moved. Overrides bbs settings.
QUOTETYPE - Preferred quote type for this conf. See rexx/examples/GlobalDefs.br for definitions. Overrides bbs setting.
QUOTECHARS - Quote chars in custom quote type. (Max length is 3.) Overrides bbs settings.
ENTERSCRIPT - Arexx script run when a conf is entered.
LEAVESCRIPT - Arexx script run when a conf is left. Above script settings overrides bbs settings.
CONFNETTYPE - Which type of network this conference is connected to.

TOP - Move conference to the top of the list.
BOTTOM - Move conference to the bottom of the list.
UP - Move conference one step up on the list.
DOWN - Move conference one step down on the list.

RESULT

Returns: 0 - Success.
5 - BBS name was not found.
7 - Conference name was not found.
30 - Failure.

```
EXAMPLE
address "BBSREAD"

/* Move conf "ChitChat" to the bottom of the list */

CONFIGCONF bbsname ' "Ultima" ' confname ' "ChitChat" ' BOTTOM
if(rc ~= 0) then
do
    say BBSREAD.LASTERROR
    exit
end

/* Set signature to "- Leo -" */

CONFIGCONF ' "Ultima" ' "ChitChat" ' SIGNATURE '- Leo -'
if(rc ~= 0) then
do
    say BBSREAD.LASTERROR
    exit
end

/* Clear the CDF_NO_SIGNATURE flag */

CDF_NO_SIGNATURE = '00004000'x
CONFIGCONF ' "Ultima" ' "ChitChat" ' CLEAR c2x(CDF_NO_SIGNATURE)
if(rc ~= 0) then
do
    say BBSREAD.LASTERROR
    exit
end

exit

NOTES
The result of combining TOP, BOTTOM, UP and DOWN in the same call
is *NOT* guaranteed to be the same on future versions.

TOP, BOTTOM, UP, DOWN and DELETECONF can only be used when the
conference given in CONFNAME exists.

To remove string settings which overrides bbs settings set the string
to '' .

BUGS

SEE ALSO
<rexexamples/GlobalDefs.br>
```

1.11 BBSREAD/CONFIGFAREA

NAME

CONFIGFAREA -- Configure fileareas.

SYNOPSIS

CONFIGFAREA BBSNAME/A,FAREANAME/A,NEWNAME/K,DELETEFAREA/S, TOP/S,
BOTTOM/S,UP/S,DOWN/S

FUNCTION

Funtion to configure file areas. Areas can be created, changed, deleted and individually rearrangend.

INPUTS

BBSNAME - Name of bbs.

FAREANAME - File area to configure. If this file area is not in the file area list, is it created.

NEWNAME - Change name of file area.

DELETEFAREA - Delete this file area.

TOP - Move file area to the top of the list.

BOTTOM - Move file area to the bottom of the list.

UP - Move file area one step up on the list.

DOWN - Move file area one step down on the list.

RESULT

Returns: 0 - Success.

5 - BBS name was not found.

6 - File area was not found.

30 - Failure.

EXAMPLE

address "BBSREAD"

/* Creates file area "Test" */

```
configfarea ' "Ultima" ' "Test"'
```

```
if(rc ~= 0) then
```

```
do
```

```
    say BBSREAD.LASTERROR
```

```
    exit
```

```
end
```

/* Renames file area "Test" to "Test2" */

```
configfarea ' "Ultima" ' "Test" ' NEWNAME ' "Test2"'
```

```
if(rc ~= 0) then
```

```
do
```

```
    say BBSREAD.LASTERROR
```

```
    exit
```

```
end
```

/* Move file area "Test2" to the top of the list */

```
configfarea ' "Ultima" ' "Test2" ' TOP
```

```
if(rc ~= 0) then
```

```
do
```

```
    say BBSREAD.LASTERROR
```

```

    exit
end

```

NOTES

The result of combining TOP, BOTTOM, UP and DOWN in the same call is *NOT* guaranteed to be the same on future versions.

TOP, BOTTOM, UP, DOWN and DELETEDFAREA can only be used when the filearea given in FAREANAME exists.

BUGS

SEE ALSO

1.12 BBSREAD/CONFIGGLOBAL

NAME

CONFIGGLOBAL -- Global configuration.

SYNOPSIS

```

CONFIGGLOBAL DNLOADPATH=DNL/K,UPLOADPATH=UPL/K,BUFFERS=BUF/K/N,
  BUFFERSIZE=BUFSIZE/K/N,SIGNATURE=SIG/K,SETFLAGS=SET/K,
  CLEARFLAGS=CLEAR/K,KEEPMSG/N/K,KEEPTIME/N/K,XPKMETHOD/K,
  USERPHONE/K,TMPDIR/K,TAGFILE/K,HAZELEVEL1/N/K,HAZELEVEL2/N/K,
  HAZELEVEL3/N/K,PGPCOMMAND/K,PGPSIGNID/K,REPLYSTRING/K,
  QUOTECHARS/K,STARTUPSCRIPT/K,QUITSCRIPT/K,BBSENTERSSCRIPT/K,
  BBSLEAVESSCRIPT/K,CONFENTERSSCRIPT/K,CONFLEAVESSCRIPT/K,
  USE/S,LASTSAVED/S

```

FUNCTION

Command to configure the global configuration.

INPUTS

```

DNLOADPATH - Path to where downloaded files resides.
UPLOADPATH - Path to where files to be uploaded resides.
BUFFERS    - Number of buffers to use in buffer system. Default is
              13 buffers.
BUFFERSIZE - Size of each buffer. Default is 5096 bytes.
SIGNATURE  - Global signature. This signature should be used on
              BBS'es where no signatures is defined.
SETFLAGS   - Global flags to set. See rexx/examples/GlobalDefs.br for
              definitions. Flags must be given as a hex string.
CLEARFLAGS - Global flags to clear. See rexx/examples/GlobalDefs.br
              for definitions. Flags must be given as a hex string.
KEEPMSG    - Messages to keep in conferences.
KEEPTIME   - Time (in seconds) to keep messages conferences.
XPKMETHOD  - Global Xpk method to use.
USERPHONE  - Phone number for user.
TMPDIR     - Path to directory to use for temp files while packing.
TAGFILE    - Path and name of global tag file.
HAZELEVEL1 - Seconds to keep messages marked with haze level 1.
HAZELEVEL2 - Seconds to keep messages marked with haze level 2.
HAZELEVEL3 - Seconds to keep messages marked with haze level 3.
PGPCOMMAND - Command for pgp (with path).
PGPSIGNID  - Id to use when PGP signing messages.

```

REPLYSTRING - Reply string to use when a message is replied and moved.
QUOTECHARS - Quote chars in custom quote type. (Max length is 3.)
STARTUPSCRIPT - Arexx script run when Thor is started.
QUITSCRIPT - Arexx script run when Thor is quited.
BBSENTERSSCRIPT - Arexx script run when a bbs is entered.
BBSLEAVESSCRIPT - Arexx script run when a bbs is left.
CONFENTERSSCRIPT - Arexx script run when a conf is entered.
CONFLEAVESSCRIPT - Arexx script run when a conf is left.
USE - Don't save the changes done to the configuration.
LASTSAVED - Retrives the configuration last saved before
applying any changes.

RESULT

Returns: 0 - Success.
30 - Failure.

EXAMPLE

address BBSREAD

/* Set global signature */

```
CONFIGGLOBAL signature '"- Leo -> Alphatesting BBSRead V3"'
if(rc ~= 0) then
do
    say BBSREAD.LASTERROR
    exit
end
```

/* Set number of buffers to use to 15, each 5096 bytes */

```
CONFIGGLOBAL Buffers 15 BufferSize 5096
if(rc ~= 0) then
do
    say BBSREAD.LASTERROR
    exit
end
```

/* Set the GCF_IGNORE_KEEPMSG flag */

```
GCF_IGNORE_KEEPMSG = '00000001'x
CONFIGGLOBAL SET c2x(GCF_IGNORE_KEEPMSG)
if(rc ~= 0) then
do
    say BBSREAD.LASTERROR
    exit
end
```

NOTES

Using the LASTSAVED switch will also load the last saved archiver configuration. If you have used the USE switch with
CONFIGARC
, is
the archiver configuration saved if you don't use the USE switch with CONFIGGLOBAL.

BUGS

SEE ALSO

<rexexamples/GlobalDefs.br>

1.13 BBSREAD/DATE2AMIGA

NAME

DATE2AMIGA -- Calculate seconds from 1.january 1978.

SYNOPSIS

DATE2AMIGA STEM/A

FUNCTION

Calculates the number of seconds from 1.january 1978 to the date specified in the ClockData stem. The ClockData is passed as stem variables. The ClockData is expected to be in the following stem variables:

<stem>.SEC

<stem>.MIN

<stem>.HOURL

<stem>.MDAY

<stem>.MONTH

<stem>.YEAR

<stem>.WDAY

INPUTS

STEM - Base name of the stem variables udes to pass the ClockData information.

RESULT

The result variable is set to the number of seconds.

EXAMPLE

NOTES

This function does no sanity checking of the data in the ClockData information.

BUGS

SEE ALSO

1.14 BBSREAD/EXTERNALBBSCONFIG

NAME

EXTERNALBBSCONFIG -- External configuration for bbs'es.

SYNOPSIS

EXTERNALBBSCONFIG BBSNAME/A,CONFNAME,PUBSCREEN/K

FUNCTION

This command calls the command for the external configuration defined in the bbstype of this bbs. If no external configuration command is defined, returns this command success.

INPUTS

BBSNAME - Name of bbs.
CONFNAME - Name of active conference. Don't use if no conferences are active.
PUBSCREEN - Name of public screen the external configuration should use for windows.

RESULT

Returns: 0 - Success.
5 - BBS name was not found.
7 - Conference name was not found.
30 - Failure.

EXAMPLE**NOTES****BUGS****SEE ALSO**

1.15 BBSREAD/FINDDUPBRMSG

NAME

FINDDUPBRMSG - Find duplicate messages in database.

SYNOPSIS

FINDDUPBRMSG BBSNAME/A,DELETEDUPINCONF/S,UNMARKCROSSPOSTS/S,
SHOWPROGRESS/S

FUNCTION

Searches databse for duplicate messages.

INPUTS

BBSNAME - Name of bbs.
DELETEDUPINCONF - Mark duplicate messages in same conference as deleted. The newest message is marked as deleted if duplicates are found.
UNMARKCROSSPOSTS - Search trough unread messages and mark crossposts as read. If a instance of the crossposted message is found read in the database, is all instances of it marked as read. Otherwise is only the first instance of the crossposted message kept unread. (The conferences will be scanned in the same order as your conference list.) This parameter is only usefull on bbs'es where messages are identified with unique message identifiers. (BRMSG_MsgID)

RESULT**EXAMPLE**

```

address "BBSREAD"

/* Mark duplicate messages in same conference as deleted */

FINDDUPBRMSG bbsname ' "News"' DELETEDUPINCONF
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

NOTES
If the message has a messageid is this messageid regarded as unique.

BUGS

SEE ALSO

```

1.16 BBSREAD/GETARCDATA

NAME
 GETARCRDATA -- Get data about archive types.

SYNOPSIS
 GETARCDATA STEM/A,ARCHIVE

FUNCTION
 Command to return data about archive types. The data is returned in stem variables. What to return depends on the ARCHIVE parameter. If this is omitted, is all data about archive types returned in the following variables:

```

<stem>.COUNT      - Number of archive types.
<stem>.n           - Name of archive type 'n'.
<stem>.n.PATTERN   - MatchPattern for archive type 'n'.
<stem>.n.UNARCCMD - UnArc command for archive type 'n'.
<stem>.n.ARCCMD   - Arc command for archive type 'n'.

```

If the ARCHIVE parameter is used, is the data about this archive type returned in the following variables:

```

<stem>.NAME       - Name of archive type.
<stem>.PATTERN   - MatchPattern for archive type.
<stem>.UNARCCMD - UnArc command for archive type.
<stem>.ARCCMD   - Arc command for archive type.

```

INPUTS
 STEM - Base name of the stem variables to put data in. Must be uppercase.
 ARCHIVE - Name of archive type to get data about.

RESULT
 Returns: 0 - Success.
 10 - Unknown archive type.
 30 - Failure.

EXAMPLE

```

address BBSREAD

GETARCDATA ARCHIVERS
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end

do i=1 to ARCHIVERS.COUNT
  say 'Name:      ' ARCHIVERS.i
  say ' Pattern: ' ARCHIVERS.i.PATTERN
  say ' UnArcCmd:' ARCHIVERS.i.UNARCCMD
  say ' ArcCmd:  ' ARCHIVERS.i.ARCCMD
end

NOTES

BUGS

SEE ALSO

```

1.17 BBSREAD/GETBBSDATA

NAME

GETBBSDATA -- Get data about a bbs.

SYNOPSIS

GETBBSDATA BBSNAME/A,STEM/A

FUNCTION

Function to get data about a bbs. The data is put into stem variables. The data returned are:

<stem>.NAME	- Name of bbs.
<stem>.GRABNAME	- Name of grabfile (without extension).
<stem>.BBSTYPE	- The type of the bbs.
<stem>.USERNAME	- The name the user is registered as.
<stem>.SIGNATURE	- Signature to use on this bbs. Overrides global settings.
<stem>.LINELENGTH	- Max linelength in messages. Overrides bbs type if > 0.
<stem>.CHARSET	- Charset the grab from this bbs uses. Overrides bbs type if ~= BRCS_ANY. See rexx/examples/GlobalDefs.br for definitions.
<stem>.SCRFLAGS	- Flags for script. See rexx/examples/GlobalDefs.br for definitions.
<stem>.FLAGS	- Flags for this bbs. See rexx/examples/GlobalDefs.br for definitions. If more than one flag is set are these flags or'ed together.
<stem>.KEEPMSG	- Messages to keep in each conference.
<stem>.KEEPTIME	- Time (in seconds) to keep messages in conferences.
<stem>.EMAILADDR	- The address the user has on the bbs.

<stem>.XPKMETHOD - What xpk method to use when packing this
 bbs. Overrides global settings.
 <stem>.USERSTREET - User street. Overrides global settings.
 <stem>.USERADDRESS - User address. Overrides global settings.
 <stem>.USERCOUNTRY - User country. Overrides global settings.
 <stem>.USERPHONE - User phone number. Overrides global settings.
 <stem>.ALIAS - Alias used on this bbs. Only for conferences
 marked with CDF_ALIAS.
 <stem>.DNLOADPATH - Dnloadpath for this bbs. Overrides global
 settings.
 <stem>.TAGFILE - Tagfile to use on this bbs. Overrides global
 settings.
 <stem>.EVENTARCHIVER - Archiver to use when packing the events.
 Overrides settings in bbs type.
 <stem>.REPLYPACKET - File name of eventual reply packet. The file
 name is expected to be relative to the upload
 directory defined for this bbs.
 <stem>.UPLOADPATH - Uploadpath for this bbs. Overrides global
 settings.
 <stem>.REPLYSTRING - Reply string to use when a message is replied
 and moved.
 <stem>.QUOTETYPE - Preferred quote type for this bbs. See
 rexx/examples/GlobalDefs.br for definitions.
 Overrides bbs type setting.
 <stem>.QUOTECHARS - Quote chars in custom quote type.
 Overrides global settings.
 <stem>.BBSENTERSSCRIPT - Arexx script run when a bbs is entered.
 <stem>.BBSLEAVESSCRIPT - Arexx script run when a bbs is left.
 <stem>.CONFENTERSSCRIPT - Arexx script run when a conf is entered.
 <stem>.CONFLEAVESSCRIPT - Arexx script run when a conf is left.
 Script settings overrides global settings.
 <stem>.BBSPATH - Path to the data directory for this bbs.
 Ends with a '/' or ':'.
 <stem>.FIRSTEVENT - Number of first event for this bbs.
 <stem>.LASTEVENT - Number of last event for this bbs.
 <stem>.NUMEVENTS - Number of active events for this bbs.
 <stem>.FIRSTUSER - Number of first user in userfile.
 <stem>.LASTUSER - Number of last user in userfile.
 <stem>.NUMUSERS - Number of active users in userfile.
 <stem>.FIRSTKILL - Number of first kill in killfile.
 <stem>.LASTKILL - Number of last kill in killfile.
 <stem>.NUMKILLS - Number of active kills in killfile.
 <stem>.SUMMARKED - Number of marked messages on this bbs.
 <stem>.SUMM2USER - Number of marked messages to the user .

Strings which overrides global and type settings does not override if they are empty.

INPUTS

BBSNAME - Name of bbs to get data about.

STEM - Base name of the stem variables to put data in. Must be uppercase.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 30 - Failure.


```
EXAMPLE
address "BBSREAD"

GETBBSDATA bbsname ' "Ultima"' stem BBSDATA
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

say 'Name:           ' BBSDATA.NAME
say 'GrabName:      ' BBSDATA.GRABNAME
say 'BBSType:       ' BBSDATA.BBSTYPE
say 'UserName:      ' BBSDATA.USERNAME
say 'FirstEvent:    ' BBSDATA.FIRSTEVENT
say 'LastEvent:     ' BBSDATA.LASTEVENT
say 'Signature:     ' BBSDATA.SIGNATURE
say 'SumMarked:     ' BBSDATA.SUMMARKED
say 'LineLength:    ' BBSDATA.LINELENGTH
say 'CharSet:       ' BBSDATA.CHARSET
say 'Flags:         ' c2x(BBSDATA.FLAGS)

exit
```

NOTES

BUGS

SEE ALSO

<rex/examples/GlobalDefs.br>

1.18 BBSREAD/GETBBSLIST

NAME

GETBBSLIST -- Obtain a list of bbses.

SYNOPSIS

GETBBSLIST STEM/A

FUNCTION

Function to obtain a list of bbses. The list is put in stemvariables. Each bbs name is in a stem variable <stem>.<num>. The number of bbses will be put in <stem>.COUNT. The numbering starts at 1.

INPUTS

STEM - Base name of the stem variables to put list in. Must be uppercase.

RESULT

Returns: 0 - Success.
30 - Failure.

EXAMPLE

```
address "BBSREAD"
```

```

GETBBSLIST stem BBSLIST
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

do i=1 to BBSLIST.COUNT
  say BBSLIST.i
end

exit

```

NOTES

BUGS

SEE ALSO

1.19 BBSREAD/GETCOMMANDINFO

NAME

GETCOMMANDINFO -- Return information on available commands.

SYNOPSIS

GETCOMMANDINFO COMMAND=CMD,STEM/A

FUNCTION

Command to return information on available commands. If the command parameter is given is the template for the command returned in the STEM variable. If the command is not given are all commands and templates be returned. The data are be put into the following variables:

```

<stem>.COUNT      - Number of commands.
<stem>.n           - Name of command 'n'.
<stem>.n.TEMPLATE - Template for command 'n'.

```

INPUTS

COMMAND - Command to get template for.
STEM - Base name of the stem variables to put data in. Must be uppercase.

RESULT

```

Returns: 0 - Success.
         10 - Unknown command.
         30 - Failure.

```

EXAMPLE

```
address BBSREAD
```

```

GETCOMMANDINFO command
                  CONFIGBBS
                  stem TEMP
if(rc ~= 0) then

```

```

do
  say BBSREAD.LASTEROR
  exit
end

say 'Template for
      CONFIGBBS
      command:' TEMP

GETCOMMANDINFO stem INFO
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

say "The number of ARexx commands in BBSREAD:" INFO.COUNT

do i=1 to INFO.COUNT
  say 'Command :' INFO.i
  say 'Template:' INFO.i.TEMPLATE
end

exit

NOTES

BUGS

SEE ALSO

```

1.20 BBSREAD/GETCONFDATA

NAME

GETCONFDATA -- Get data about a conference.

SYNOPSIS

GETCONFDATA BBSNAME/A, CONFNAME/A, STEM/A

FUNCTION

Function to get the data about a conference. The data is put into stem variables. The data returned are:

<stem>.NAME	- Name of conference.
<stem>.KEEPMSG	- Messages to keep in conference.
<stem>.KEEPTIME	- Time (in seconds) to keep messages in conference.
<stem>.SIGNATURE	- Signature for conference. Overrides bbs settings.
<stem>.ALIAS	- Alias used in conference. Will only be used in conferences marked with CDF_ALIAS. Overrides bbs settings.
<stem>.BBSCONFNR	- The internal number the conference has on the bbs.

<stem>.LINELENGTH - Max linelength in messages. Overrides bbs settings if > 0.
 <stem>.CHARSET - Charset the grab from this bbs uses in this conference. Overrides bbs settings if ~= BRCS_ANY. See rexx/examples/GlobalDefs.br for definitions.
 <stem>.XPKMETHOD - What xpk method to use when packing this conference. Overrides bbs settings.
 <stem>.TAGFILE - Tagfile to use in this conference. Overrides bbs settings.
 <stem>.EMAILADDR - The Email the user has in this conference. For use on bbses where the user is member of more than one net.
 <stem>.REPLYSTRING - Reply string to use when a message is replied and moved.
 <stem>.QUOTETYPE - Preferred quote type for this conf. See rexx/examples/GlobalDefs.br for definitions. Overrides bbs setting.
 <stem>.QUOTECHARS - Quote chars in custom quote type. Overrides bbs settings.
 <stem>.ENTERSCRIPT - Arexx script run when this conf is entered.
 <stem>.LEAVESCRIPT - Arexx script run when this conf is left. Script settings overrides bbs settings.
 <stem>.FLAGS - Flags for this conference. See rexx/examples/GlobalDefs.br for definitions. If more than one flag is set are these flags or'ed together.
 <stem>.FIRSTMSG - First message in conference.
 <stem>.LASTMSG - Last message in conference.
 <stem>.NUMMESSAGES - Number of active messages in conference.
 <stem>.MSGMARKED - Amount of messages marked.
 <stem>.MARKED2USER - Amount of the marked messages which is to the user.
 <stem>.CONFNETTYPE - Which type of network this conference is connected to. This tag is to be used on BBS'es of BBS types which supports several network types. The default value is CDNT_NONET (0). See <rexx/examples/GlobalDefs.br> for definitions. New definitions are added on request. No need to use this tag if the bbs type only supports one network.

INPUTS

BBSNAME - Name of the bbs the conference is on.
 CONFNAME - Name of conference to get data about.
 STEM - Base name of the stem variables to put data in. Must be uppercase.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 7 - Conference name was not found.
 30 - Failure.

EXAMPLE

address "BBSREAD"

GETCONFDATA bbsname ' "Ultima" ' confname ' "ChitChat" ' stem CONFDATA

```

if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

say 'Name:          ' CONFDATA.NAME
say 'KeepMsg:      ' CONFDATA.KEEPMSG
say 'KeepTime:     ' CONFDATA.KEEPTIME
say 'Signature:    ' CONFDATA.SIGNATURE
say 'Alias:        ' CONFDATA.ALIAS
say 'BBSConfNr:   ' CONFDATA.BBSCONFNR
say 'LineLength:  ' CONFDATA.LINELENGTH
say 'CharSet:     ' CONFDATA.CHARSET
say 'XPKMethod:   ' CONFDATA.XPKMETHOD
say 'TagFile:     ' CONFDATA.TAGFILE
say 'Flags:       ' c2x(CONFDATA.FLAGS)
say 'FirstMsg:    ' CONFDATA.FIRSTMSG
say 'LastMsg:     ' CONFDATA.LASTMSG
say 'NumMessages:' CONFDATA.NUMMESSAGES
say 'MsgMarked:  ' CONFDATA.MSGMARKED
say 'Marked2User:' CONFDATA.MARKED2USER

```

NOTES

BUGS

SEE ALSO

<rexexamples/GlobalDefs.br>

1.21 BBSREAD/GETCONFLIST

NAME

GETCONFLIST -- Obtain a list of conferences at specified bbs.

SYNOPSIS

GETCONFLIST BBSNAME/A,STEM/A

FUNCTION

Function to obtain a list of conferences at a specified bbs. The list is put in stemvariables. Each conference name will be in a stem variable <stem>.<num>. The number of conferences on the bbs is put in <stem>.COUNT. The numbering starts at 1.

INPUTS

BBSNAME - Name of bbs.

STEM - Base name of the stem variables to put list in. Must be uppercase.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 30 - Failure.

EXAMPLE

```

address "BBSREAD"

GETCONFLIST bbsname ' "Ultima" ' stem CONFLIST
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

do i=1 to CONFLIST.COUNT
  say CONFLIST.i
end

exit

NOTES

BUGS

SEE ALSO

```

1.22 BBSREAD/GETFAREADATA

```

NAME
GETFAREADATA -- Get data about a file area.

SYNOPSIS
GETFAREADATA BBSNAME/A,FAREANAME/A,STEM/A

FUNCTION
Function to get data about a file area. The data is put into stem
variables. The data returned is:
<stem>.FIRSTFILE - Number of the first file in this area.
<stem>.NUMFILES - Number of files in this area

INPUTS
BBSNAME - Name of bbs.
FAREANAME - File area to get data about.
STEM - Base name of the stem variables to put data in. Must be
uppercase.

RESULT
Returns: 0 - Success.
         5 - BBS name was not found.
         6 - File area was not found.
        30 - Failure.

EXAMPLE
address "BBSREAD"

GETFAREADATA bbsname ' "Ultima" ' fareaname ' "UPLOAD" ' stem FAREADATA
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit

```

```
end

say 'FirstFile: ' FAREADATA.FIRSTFILE
say 'NumFiles: ' FAREADATA.NUMFILES

exit

NOTES

BUGS

SEE ALSO
```

1.23 BBSREAD/GETFAREALIST

NAME
GETFAREALIST -- Obtain a list of file areas at specified bbs.

SYNOPSIS
GETFAREALIST BBSNAME/A,STEM/A

FUNCTION
Function to obtain a list of file areas at a specified bbs. The list is put in stemvariables. Each file area name will be in a stem variable <stem>.<num>. The number of file areas on the bbs is put in <stem>.COUNT. The numbering starts at 1.

INPUTS
BBSNAME - Name of bbs.
STEM - Base name of the stem variables to put list in. Must be uppercase.

RESULT
Returns: 0 - Success.
5 - BBS name was not found.
30 - Failure.

EXAMPLE
address "BBSREAD"

```
GETFAREALIST bbsname '"byte bazaar"' stem FAREALIST
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end

do i=1 to FAREALIST.COUNT
  say FAREALIST.i
end

exit
```

NOTES

BUGS

SEE ALSO

1.24 BBSREAD/GETGLOBALDATA

NAME

GETGLOBALDATA -- Get global data.

SYNOPSIS

GETGLOBALDATA STEM/A

FUNCTION

Command to get the global data in the database. The data is put into stem variables. The data returned are:

<stem>.DNLOADPATH - Path to where downloaded files resides.
 <stem>.UPLOADPATH - Path to where files to be uploaded resides.
 <stem>.BUFFERS - Number of buffers to use in buffer system.
 <stem>.BUFFERSIZE - Size of each buffer.
 <stem>.SIGNATURE - Global signature.
 <stem>.KEEPMSG - Messages to keep in conferences.
 <stem>.KEEPTIME - Time (in seconds) to keep messages conferences.
 <stem>.XPKMETHOD - Global Xpk method to use.
 <stem>.TMPDIR - Path to directory to use for temp files while packing.
 <stem>.TAGFILE - Path and name of global tag file.
 <stem>.USERNAME - User data from key-file.
 <stem>.USERSTREET - - " -
 <stem>.USERADDRESS - - " -
 <stem>.USERCOUNTRY - - " -
 <stem>.USERPHONE - Phone number for user.
 <stem>.HAZELEVEL1 - Seconds to keep hazelevel 1 messages.
 <stem>.HAZELEVEL2 - Seconds to keep hazelevel 2 messages.
 <stem>.HAZELEVEL3 - Seconds to keep hazelevel 3 messages.
 <stem>.PGPCOMMAND - Command for pgp (with path)
 <stem>.PGPSIGNID - Id to use when PGP signing messages.
 <stem>.REPLYSTRING - Reply string to use when a message is replied and moved.
 <stem>.QUOTECHARS - Quote chars in custom quote type.
 <stem>.STARTUPSCRIPT - Arexx script run when Thor is started.
 <stem>.QUITSCRIPT - Arexx script run when Thor is quited.
 <stem>.BBSENTERSSCRIPT - Arexx script run when a bbs is entered.
 <stem>.BBSLEAVESSCRIPT - Arexx script run when a bbs is left.
 <stem>.CONFENTERSSCRIPT - Arexx script run when a conf is entered.
 <stem>.CONFLEAVESSCRIPT - Arexx script run when a conf is entered.
 <stem>.FLAGS - Global flags. See
 rexx/examples/GlobalDefs.br for definitions. If more than one
 flag is set are these flags or'ed together.

INPUTS

STEM - Base name of the stem variables to put data in. Must be uppercase.

RESULT

Returns: 0 - Success.

30 - Failure.

```

EXAMPLE
address "BBSREAD"

GETGLOBALDATA stem GLOBALDATA
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

say 'Dnloadpath:' GLOBALDATA.DNLOADPATH
say 'Uploadpath:' GLOBALDATA.UPLOADPATH
say 'KeepMsg: ' GLOBALDATA.KEEPMSG
say 'KeepTime: ' GLOBALDATA.KEEPTIME
say 'Signature: ' GLOBALDATA.SIGNATURE
say 'XPKMethod: ' GLOBALDATA.XPKMETHOD
say 'TagFile: ' GLOBALDATA.TAGFILE
say 'Flags: ' c2x(GLOBALDATA.FLAGS)

```

NOTES

BUGS

SEE ALSO

<rexexamples/GlobalDefs.br>

1.25 BBSREAD/PACKDATAFILE

NAME

PACKDATAFILE -- Removes deleted entries from datafiles.

SYNOPSIS

```

PACKDATAFILE BBSNAME/A,CONFNAME/K,EVENTDATA/S,USERDATA/S,KILLDATA/S,
FILEDATA/S,SAVEPACKEDBRIEF/K,SHOWPROGRESS/S

```

FUNCTION

Command to 'pack' the datafiles. Removes all deleted entries and optimize the database for speed.

INPUTS

BBSNAME - Name of bbs to pack on.
CONFNAME - Name of conference to pack datafiles for.
EVENTDATA - Pack event data.
USERDATA - Pack user data.
KILLDATA - Pack kill data.
FILEDATA - Pack file data.
SAVEPACKEDBRIEF - File to save packed messages to. All messages which are deleted because of KeepTime and KeepMsg will be saved to this file. If the file exists are the messages appended to the file. The messages are saved in BRIEF format.
SHOWPROGRESS - Show progress while packing.

RESULT

```
Returns: 0 - Success.
         5 - BBS name was not found.
         7 - Conference name was not found.
        30 - Failure.
```

EXAMPLE

```
address BBSREAD
```

```
/* Pack conference comp.sys.amiga.misc on
   bbs News. Show progress while packing. */
```

```
PACKDATAFILE 'News' confname 'comp.sys.amiga.misc' SHOWPROGRESS
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

```
/* Pack event data on bbs News. */
```

```
PACKDATAFILE 'News' EVENTDATA
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

NOTES

BUGS

SEE ALSO

1.26 BBSREAD/READARGS

NAME

READARGS -- Parse command line input.

SYNOPSIS

READARGS TEMPLATE/A,STEM/A,CMDLINE/K/A/F

FUNCTION

Parses the argument string according to a template. This is a front end to the `dos.library/ReadAgrs()` function. The arguments are returned in stem variables matching the name of the parameter in the template like this:

```
<stem>.<parameter>
```

'/M' type parameters are returned in numbered stem lists like this:

```
<stem>.<parameter>.COUNT - Number of arguments.
```

```
<stem>.<parameter>.n - 1 <= n <= COUNT
```

The stem variables for string and number arguments are only set if they actually occur in CMDLINE. This makes it possible for the arexx

script to check if a string or number parameter was used in CMDLINE. This makes it also possible to set defaults.

INPUTS

TEMPLATE - Template string. See dos.library/ReadArgs() how to set up one.

STEM - Stem variables base name for returned arguments.

CMDLINE - Command line to parse.

RESULT

Returns: 0 - Success.
5 - ReadArgs failed.
30 - Failure.

EXAMPLE

```
address "BBSREAD"
```

```
drop ARGV. /* Important */
```

```
ARGV.CONFNAME = 'Post' /* Defaults */
```

```
ARGV.EVENTDATA = 1
```

```
arguments = '"Ultima Thule" CONFNAME "Småprat" NUMBER 7769 ',  
            || ' EVENTDATA MULTIPLE Mul1 MULTIPLE Mul2'
```

```
temp = 'BBSNAME/A,CONFNAME/K,EVENTDATA/S/T,USERDATA/S,',  
       || 'NUMBER/N,MULTIPLE/K/M'
```

```
READARGS temp ARGV CMDLINE arguments
```

```
if(rc ~= 0) then
```

```
do
```

```
    say BBSREAD.LASTERROR
```

```
    exit
```

```
end
```

```
/* BBSNAME is a '/A' parameter, and will always be set if  
   READARGS returns success */
```

```
say 'BBSNAME:' ARGV.BBSNAME
```

```
if(symbol("ARGV.CONFNAME") = "VAR") then say 'CONFNAME:' ARGV.CONFNAME
```

```
say 'EVENTDATA:' ARGV.EVENTDATA
```

```
say 'USERDATA:' ARGV.USERDATA
```

```
if(symbol("ARGV.NUMBER") = "VAR") then say 'NUMBER:' ARGV.NUMBER
```

```
if(symbol("ARGV.MULTIPLE.COUNT") = "VAR") then
```

```
do
```

```
    say 'MULTIPLE.COUNT:' ARGV.MULTIPLE.COUNT
```

```
    do n = 1 to ARGV.MULTIPLE.COUNT
```

```
        say ' MULTIPLE.' || n || ':' ARGV.MULTIPLE.n
```

```
    end
```

```
end
```

NOTES

BUGS

SEE ALSO

1.27 BBSREAD/READBREVENT

NAME

READBREVENT -- Read the entry for an event from the database.

SYNOPSIS

READBREVENT BBSNAME/A, EVENTNR/N/A, TAGSSTEM/K, DATASTEM/K

FUNCTION

Reads an event entry for a bbs from the database. The event entry tags and data are returned in stem variables. The tag stem variables returned are:

<tagstem>.TONAME - See
 WRITEBREVENT
 for description of tagstem
 variables.

<tagstem>.TOADDR
 <tagstem>.SUBJECT
 <tagstem>.CONFERENCE
 <tagstem>.REFNR
 <tagstem>.REFORIGINALNR
 <tagstem>.REFID
 <tagstem>.MSGFILE
 <tagstem>.PRIVATE
 <tagstem>.LOCALFILE
 <tagstem>.DIRECTORY
 <tagstem>.FILENAME
 <tagstem>.DOWNLOADNOTIFY
 <tagstem>.FILEDESCR
 <tagstem>.DETAILEDFILEDESCR
 <tagstem>.FROMMESSAGENR
 <tagstem>.TOMESSAGENR
 <tagstem>.COMMANDSTRING
 <tagstem>.BOOLEAN
 <tagstem>.DATE
 <tagstem>.PGPSIGNID
 <tagstem>.PGPENCRYPTID
 <tagstem>.REFCONFERENCE
 <tagstem>.URGENT
 <tagstem>.IMPORTANT
 <tagstem>.CONFIDENTIAL
 <tagstem>.RETURNRECIEPT
 <tagstem>.ENCODE8BIT

<datastem>.EVENTTYPE - The type of this event.

<datastem>.EVENTDATE - The date the event was added to the database.
 In seconds since 1.January 1978.

<datastem>.FLAGS - The flags for this event. Currently defined:
 EDF_DELETED = '00000001'x - Event is deleted.

```
EDF_PACKED      = '00000002'x - Event is packed.
EDF_DONE        = '00000004'x - Event is done.
EDF_ERROR       = '00000008'x - Error performing event.
EDF_UNRECOVERABLE = '00000010'x - Event can not be undeleted.
EDF_FREEZE      = '00000020'x - Event is frozen. Will not be
                        done as long as this flag is
                        set.
```

If more than one flag is set are these flags or'ed together.

INPUTS

```
BBSNAME - Name of bbs.
EVENTNR - Number of the event to read.
TAGSSTEM - Base name of the stem variables the tag data should be put
           in. If this tag is omitted or the event is marked as deleted is
           no tag data returned.
DATASTEM - Base name of the stem variables the event entry data
           should be put in. If this tag is omitted is no event entry data
           returned.
```

RESULT

```
Returns:  0 - Success.
          5 - BBS name was not found.
          30 - Failure.
```

EXAMPLE

```
address BBSREAD
```

```
EDB_DELETED      = 0      /* Event is deleted */
EVE_UPLOAD       = 5      /* Upload file */
```

```
drop EVENTDATA.
```

```
READBREVENT bbsname ' "Ultima" ' eventnr 2 datastem EVENTDATA
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

```
if EVENTDATA.EVENTTYPE = EVE_UPLOAD then
do
  say 'Upload event'
  say 'Event date:' EVENTDATA.EVENTDATE
  say 'Event flags:' c2x(EVENTDATA.FLAGS)

  if ~bittst(EVENTDATA.FLAGS,EDB_DELETED) then
  do
    say 'Event tags:'

    drop EVENTTAGS.
```

```
READBREVENT bbsname ' "Ultima" ' eventnr 2 tagstem EVENTTAGS
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

```

say ' Filename:' EVENTTAGS.FILENAME
say ' FileDescr:' EVENTTAGS.FILEDESCR

if(symbol("EVENTTAGS.TONAME") = "VAR") then,
  say ' ToName:', EVENTTAGS.TONAME
if(symbol("EVENTTAGS.CONFERENCE") = "VAR") then,
  say ' Conference:' EVENTTAGS.CONFERENCE
if(symbol("EVENTTAGS.DETAILEDFILEDESCR") = "VAR") then,
  say ' DetailedFileDescr:' EVENTTAGS.DETAILEDFILEDESCR
if(symbol("EVENTTAGS.LOCALFILE") = "VAR") then,
  say ' LocalFile:' EVENTTAGS.LOCALFILE
end
else say 'Event is deleted'
end
else say 'Event is not a upload event'

```

NOTES

BUGS

SEE ALSO

1.28 BBSREAD/READBRFILE

NAME

READBRFILE -- Read an entry from the file database.

SYNOPSIS

READBRFILE BBSNAME/A,FAREANAME/A,FILENR/N/A,TAGSSTEM/K,DATASTEM/K

FUNCTION

Reads a file entry from a file area in the database for a bbs. The file entry tags and data are returned in stem variables. The tag stem variables returned are:

<tagsstem>.NAME - See
WRITEBRFILE
for description of tagsstem
variables.

<tagsstem>.DATE

<tagsstem>.SIZE

<tagsstem>.DOWNLOADS

<tagsstem>.DESCRIPTION.COUNT

<tagsstem>.DESCRIPTION.<1-count>

<datastem>.FILEDATE - The date the file was added to the database.
Same format as <tagsstem>.DATE.

<datastem>.FLAGS - The flags for this file.

Currently defined flags: FDF_DELETED = '00000001'x

If more than one flag is set are these flags or'ed together.

INPUTS

BBSNAME - Name of bbs.
 FAREANAME - File area to read the file entry from.
 FILENR - Number of the file to read from the database.
 TAGSSTEM - Base name of the stem variables the tag data should be put in. If this tag is omitted or the file is marked as deleted are no tag data returned.
 DATASTEM - Base name of the stemvariables the file entry data should be put in. If this tag is omitted are no file entry data returned.

RESULT

Returns: 0 - Success. The result variable is set to the number of the next file in the same file area as this. *Must* be used when traversing the files of a file area.
 5 - BBS name was not found.
 6 - File area was not found.
 30 - Failure.

EXAMPLE

/* Shows the first file in the UPLOAD file area */

```
options results
address "BBSREAD"
FDF_DELETED = '00000001'x /* File deleted flag */
```

/* Needs the number of the first file in the file area */

```
                GETFAREADATA
                bbsname 'Ultima' fareaname 'UPLOAD' stem FAREADATA
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end

filenr = FAREADATA.FIRSTFILE

drop FILE. /* Important */

READBRFILE 'Ultima' 'UPLOAD' filenr tagsstem FILE datastem DATA
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end

filenr = result /* The next file in the area */

/* Check if file is marked as deleted */

if (bitand(DATA.FLAGS,FDF_DELETED) ~= FDF_DELETED) then
do
  /* Check which tags are used for this file */
```

```

if(symbol("FILE.DATE") = "VAR") then fdate = FILE.DATE
else fdate = DATA.FILEDATE

        AMIGA2DATE
        fdate CLOCKDATA /* Convert date */
fdatestr = right(CLOCKDATA.YEAR, 2) || right('0' ||
        CLOCKDATA.MONTH, 2) || right('0' || CLOCKDATA.MDAY, 2)

if(symbol("FILE.SIZE") = "VAR") then fsize = FILE.SIZE
else fsize = "Unkn"

if(symbol("FILE.DOWNLOADS") = "VAR") then fdnls = FILE.DOWNLOADS
else fdnls = "Unkn"

if(symbol("FILE.DESCRPTION.COUNT") = "VAR") then,
    descr = FILE.DESCRPTION.1
else descr = "NONE"

/* Show the file information in the file database */

say left(FILE.NAME,16) || " " || fdatestr || " ",
    || right(fsize,7) || right(fdnls,4) || " " descr

/* Show eventual multiline file description */

if(descr ~= "NONE") then
do
    if(FILE.DESCRPTION.COUNT > 1) then
    do
        do n=2 to FILE.DESCRPTION.COUNT
            say left("",37) || FILE.DESCRPTION.n
        end
    end
end
end
else say 'File is deleted'

say 'Next File: ' filenr
exit

NOTES

BUGS

SEE ALSO

```

1.29 BBSREAD/READBRMESSAGE

NAME
 READBRMESSAGE -- Read the entry for a message from the database.

SYNOPSIS

READBRMESSAGE BBSNAME/A, CONFNAME/A, MSGNR/N/A, HEADSTEM/K, TEXTSTEM/K,
 DATASTEM/K

FUNCTION

Reads a message entry for a conference from the database.

The multiple parts in multipart messages should be presented in the order they are set up in the PART tag. The TEXT tag should always be preseted first.

The message entry headtags, texttags and data are returned in stem variables. The tag stem variables returned are:

<headstem>.FROMNAME - See
 WRITEBRMESSAGE
 for description of headstem
 variables.

<headstem>.FROMADDR
 <headstem>.TONAME
 <headstem>.TOADDR
 <headstem>.MSGID
 <headstem>.ORIGINALNR
 <headstem>.REFID
 <headstem>.REFNR
 <headstem>.CREATIONDATE
 <headstem>.CREATIONDATETEXT
 <headstem>.SUBJECT

<textstem>.REPLYCONF - See
 WRITEBRMESSAGE
 for description of textstem
 variables.

<textstem>.REPLYNAME
 <textstem>.REPLYADDR
 <textstem>.COMMENT
 <textstem>.TEXT
 <textstem>.PART

<datastem>.MSGDATE - Date when message was added. In seconds since
 1.january 1978.
 <datastem>.REFERENCE - The messagenumber this message refers to.
 <datastem>.FIRSTREF - First message which refers to this message.
 <datastem>.LASTREF - Last message which refers to this message.
 <datastem>.PREVREF - Previous message which refers to same message
 as this message.
 <datastem>.NEXTREF - Next message which refers to same message as
 this message.
 <datastem>.HAZELEVEL - Which haze level this message is marked with.
 <datastem>.FLAGS - The flags for this message.
 rexexamples/GlobalDefs.br for definitions. If more than one flag
 is set are these flags or'ed together.

INPUTS

BBSNAME - Name of bbs.
 CONFNAME - Name of conference.
 MSGNR - Number of the message to read.
 HEADSTEM - Base name of the stem variables the head tag data should

be put in. If this tag is omitted or the message is marked as deleted, is no head tag data returned.

TEXTSTEM - Base name of the stem variables the text tag data should be put in. If this tag is omitted or the message is marked as deleted, is no text tag data returned.

DATASTEM - Base name of the stem variables the message entry data should be put in. If this tag is omitted, is no message entry data returned.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 7 - Conference name was not found.
 30 - Failure.

EXAMPLE

address BBSREAD

MDB_DELETED = 5 /* Message is deleted. */

Globals = 'HEADTAGS. TEXTTAGS. Globals'

/* Read only message data first */

```
READBRMESSAGE bbsname "Ultima" confname "Generelt",
  msgnr 5 datastem MSGDATA
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

/* Test if message is deleted */

```
if ~bittst(MSGDATA.FLAGS,MDB_DELETED) then
do
  drop HEADTAGS. /* Important */
  drop TEXTTAGS. /* Important */
```

/* Read message header and text */

```
READBRMESSAGE bbsname "Ultima" confname "Generelt",
  msgnr 5 headstem HEADTAGS textstem TEXTTAGS
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

```
  call ShowMessagePart(' ', 'HEADTAGS', 'TEXTTAGS')
end
else say 'Message is deleted'

exit
```

ShowMessagePart:

```
/* Since message parts can be recursive should a recursive function
   be used to show a message. */
```

```
interpret 'procedure expose' Globals
parse arg pad, hstem, tstem
```

```
say '0a'x || pad'Message:'
```

```
from = value(hstem'.FROMNAME')
```

```
/* FROMADDR is optional */
```

```
if symbol(hstem'.FROMADDR') = "VAR" then,
  from = from || ' <' || value(hstem'.FROMADDR' || '>'
```

```
say pad'From:' from
```

```
if symbol(hstem'.TONAME') = "VAR" then
do
```

```
  to = value(hstem'.TONAME')
  if symbol(hstem'.TOADDR') = "VAR" then,
    to = to || ' <' || value(hstem'.TOADDR' || '>'
```

```
    say pad'To:' to
```

```
end
```

```
say pad'Subject:' value(hstem'.SUBJECT') '0a'x
```

```
/* Show message text */
```

```
cnt = value(tstem'.TEXT.COUNT')
```

```
do n=1 to cnt
  say pad || value(tstem'.TEXT.'n)
end
```

```
/* Show comment */
```

```
if(symbol(tstem'.COMMENT.COUNT') = "VAR") then
do
```

```
  say
```

```
  cnt = value(tstem'.COMMENT.COUNT')
```

```
  do n=1 to cnt
    say pad'Comment:' value(tstem'.COMMENT.'n)
  end
```

```
end
```

```
/* Check for message parts */
```

```
if(symbol(tstem'.PART.COUNT') = "VAR") then
do
```

```
parts = value(tstem'.PART.COUNT')

do n=1 to parts

  /* Is it a binary part? */

  if(symbol(tstem'.PART.'n'.BINARY') = "VAR") then
  do
    say '0a'x || pad'Binary:' value(tstem'.PART.'n'.BINARY')
    say pad'BinaryDescription:',
    value(tstem'.PART.'n'.BINARY.DESC')

    cnt = value(tstem'.PART.'n'.BINARY.COMMENT.COUNT')

    do m=1 to cnt
      say pad'BinaryComment:',
      value(tstem'.PART.'n'.BINARY.COMMENT.'m)
    end
  end

  /* Is it a text part? */

  else if(symbol(tstem'.PART.'n'.TEXT.COUNT') = "VAR") then
  do
    say

    cnt = value(tstem'.PART.'n'.TEXT.COUNT')

    do m=1 to cnt
      say pad'Text:' value(tstem'.PART.'n'.TEXT.'m)
    end

    cnt = value(tstem'.PART.'n'.TEXT.COMMENT.COUNT')

    do m=1 to cnt
      say pad'Text:' value(tstem'.PART.'n'.TEXT.COMMENT.'m)
    end
  end
end
else
end

/* It is a message part. */

do
  usestem = tstem'.PART.'n'.MSG'

  call ShowMessagePart(pad' ', usestem, usestem)
end
end
end

return 0
```

NOTES

BUGS

SEE ALSO

1.30 BBSREAD/READBRUSER

NAME

READBRUSER -- Read an entry from the user database.

SYNOPSIS

READBRUSER BBSNAME/A,USERNR/N/A,TAGSSTEM/K,DATASTEM/K

FUNCTION

Reads an user entry from a user database. The user entry tags and data are returned in stem variables. The tag stem variables returned are:

<tagstem>.NAME - See
WRITEBRUSER
for description of tagstem
variables.

<tagstem>.ADDRESS

<tagstem>.ALIAS

<tagstem>.COMMENT.COUNT

<tagstem>.COMMENT.<l-count>

<tagstem>.ENCODE8BITMSG

<tagstem>.PGPKEYID

<datastem>.USERDATE - The date the user was added to the database.
Seconds since 1.january 1978.

<datastem>.FLAGS - The flags for this user.

Currently defined flags:

UDF_DELETED = '00000001'x User is deleted

UDF_UNRECOVERABLE = '00000002'x User can not be undeleted

If more than one flag is set are these flags or'ed together.

INPUTS

BBSNAME - Name of bbs.

USERNR - Number of the user to read from database.

TAGSSTEM - Base name of the stem variables the tag data should be put in. If this tag is omitted or the user is marked as deleted, is no tag data returned.

DATASTEM - Base name of the stemvariables the user entry data should be put in. If this tag is omitted, is no user entry data returned.

RESULT

Returns: 0 - Success.

5 - BBS name was not found.

30 - Failure.

EXAMPLE

address BBSREAD

```
UDB_DELETED = 0 /* User is deleted */

/* Reads user 1 from the userdatabase on Ultima bbs */

drop USERDATA.

READBRUSER bbsname "Ultima" usernr 1 datastem USERDATA
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

say 'User date :' USERDATA.USERDATE
say 'User flags:' c2x(USERDATA.FLAGS)

if ~bittst(USERDATA.FLAGS,UDB_DELETED) then
do
  say 'User tags:'

  drop USERTAGS.

  READBRUSER bbsname "Ultima" usernr 1 tagsstem USERTAGS
  if(rc ~= 0) then
  do
    say BBSREAD.LASTEROR
    exit
  end

  say ' Name:' USERTAGS.NAME

  if(symbol("USERTAGS.ADDRESS") = "VAR") then,
    say ' Address:' USERTAGS.ADDRESS

  if(symbol("USERTAGS.ALIAS") = "VAR") then,
    say ' Alias:' USERTAGS.ALIAS

  if(symbol("USERTAGS.COMMENT.COUNT") = "VAR") then
  do
    say ' Comment:' USERTAGS.COMMENT.1
    do n = 2 to USERTAGS.COMMENT.COUNT
      say '          ' USERTAGS.COMMENT.n
    end
  end
end

end
else say 'User is deleted'

NOTES

BUGS

SEE ALSO
```

1.31 BBSREAD/READPASSCONFLIST

NAME

READPASSCONFLIST -- Read passive conference list.

SYNOPSIS

READPASSCONFLIST BBSNAME/A,STEM/A

FUNCTION

Gives you a list of all conferences in the passive conference list datafiles. The passive conference list is a list of *all* available conferences at the bbs. It should be used when the user want to send a join event. The list of conference names are returned in the following stem variables:

<stem>.COUNT - Number of conferences in the list.
(0 if the list is empty)

<stem>.<1 <= n <= count>.NAME - The conference names.

<stem>.<1 <= n <= count>.DESCR - Conference description. This stem variable is not be set if no description exists.

<stem>.<1 <= n <= count>.BBSCONFNR - The internal number this conference has on the bbs.

<stem>.<1 <= n <= count>.FLAGS - Flags for conference. Same as defined for GETCONFDATA.FLAGS.

INPUTS

BBSNAME - Name of bbs.

STEM - Base name for the stem variables used to return the conference list.

RESULT

Returns: 0 - Success.
5 - BBS name was not found.
30 - Failure.

EXAMPLE

```
address "BBSREAD"
```

```
READPASSCONFLIST bbsname "Ultima" stem LIST
```

```
if(rc ~= 0) then
```

```
do
```

```
    say BBSREAD.LASTEROR
```

```
    exit
```

```
end
```

```
if(LIST.COUNT = 0) then
```

```
do
```

```
    say 'No passive conference list'
```

```
    exit
```

```
end
```

```
do n = 1 to LIST.COUNT
```

```
    if(symbol("LIST.n.DESCR") = "VAR") then desc = LIST.n.DESCR
```

```
    else desc = ''
```

```
say left(LIST.n.NAME, 15) right(LIST.n.BBSCONFNR, 4) c2x(LIST.n.FLAGS) desc
end
```

NOTES

BUGS

SEE ALSO

1.32 BBSREAD/SEARCHBRFILE

NAME

SEARCHBRFILE -- Search file database.

SYNOPSIS

SEARCHBRFILE BBSNAME/A, STEM/A, FAREANAME/K, SEARCH/K, NAME/S, KEYWORD/S,
NEWERTHAN/K/N

FUNCTION

Command to search the file database for a file matching the given arguments. It's possible to search a file area or all file areas on a bbs. The search result are returned in stem variables. The stem variables returned are:

```
<stem>.FILEAREA.COUNT - Number of file areas.
<stem>.FILEAREA.<l <= fareanr <= count> Name of file area.
<stem>.FILE.<fareanr>.COUNT - Number of files found in
    <stem>.FILE.<fareanr>.
<stem>.FILE.<fareanr>.<l - count> - File number of the found file.
```

When there are found no files are only the <stem>.FILEAREA.COUNT variable set. (0)

INPUTS

BBSNAME - Name of bbs.

STEM - Base name of the stem variables the search result should be put in.

FAREANAME - File area to search, if this parameter is omitted are all file areas on the bbs searched.

SEARCH - String to search for. AmigaDos wildcards are allowed.

NAME - Search for a file with a name matching SEARCH parameter.

KEYWORD - Search hole file entry for a match of SEARCH. This switch has higher priority than the NAME switch.

NEWERTHAN - Find files newer than a given date. The format of the date is expected to be: yymmdd. Example: 931231.

RESULT

```
Returns: 0 - Success. The result variable is set to the total number
           of found files.
          5 - BBS name was not found.
          6 - File area was not found.
          30 - Failure.
```

EXAMPLE

options results


```

address "BBSREAD"

SEARCHBRFILE bbsname ' "Ultima"' stem SRESULT search ' "arexx"',
  keyword newerthan 930601

if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

if(result > 0) then
do
  do f=1 to SRESULT.FILEAREA.COUNT
    say '0a'x || '(' || SRESULT.FILEAREA.f || ')'

    do i=1 to SRESULT.FILE.f.COUNT
      say 'File nr:' SRESULT.FILE.f.i
    end
  end
end
else say 'No files found'

exit

NOTES

BUGS

SEE ALSO

```

1.33 BBSREAD/SEARCHBRUSER

NAME
SEARCHBRUSER -- Search user database.

SYNOPSIS
SEARCHBRUSER BBSNAME/A,STEM/A,SEARCH/K,NAME/S,ADDRESS/S,ALIAS/S,
COMMENT/S,SUGGESTUSERSSTEM/K

FUNCTION
Command to search the user database for a user matching the given arguments. The search order is aliases, names, addresses and comment. The search result are returned in stem variables. The stem variables returned are:

<stem>.COUNT - Number of users found.
<stem>.<n>.USERNR - The number of the n'th found user.
<stem>.<n>.FOUNDINTAG - In which tag the search string was found.
Possible values are:

FUIT_NAME	= 0
FUIT_ADDRESS	= 1
FUIT_ALIAS	= 2
FUIT_COMMENT	= 3

INPUTS

BBSNAME - Name of bbs.

STEM - Base name of the stem variables the search result should be put in.

SEARCH - String to search for. AmigaDos wildcards are allowed.

NAME - Search for a user with a name matching the SEARCH parameter exact.

ADDRESS - Search for a user with an address matching the SEARCH parameter exact.

ALIAS - Search for a user with an alias matching the SEARCH parameter exact.

COMMENT - Search for a user with a comment matching the SEARCH parameter. If wildcards are used must the search pattern match a line in the comment.

SUGGESTUSERSSTEM - If the SEARCH parameter doesn't match any user names in the database can a list of suggestions for user names be returned in this stem base name. No suggestion list will be returned if the switch NAME isn't used, if the SEARCH parameter contains wildcards, if there are no users to suggest or if there was a failure. The stem variables returned are:

<stem>.COUNT - Number of users in suggestion list.

<stem>.<n> - Name of n'th user in list.

<stem>.<n>.USERNR - Number n'th user in list.

RESULT

Returns: 0 - Success. The result variable will be set to the total number of found users.

5 - BBS name was not found.

30 - Failure.

EXAMPLE

options results

address "BBSREAD"

```
SEARCHBRUSER bbsname '"Ultima"' stem SRESULT,
  search '"Eivind Nordseth"' NAME suggestusersstem SUGSTEM
```

```
if(rc ~= 0) then
```

```
do
```

```
  say BBSREAD.LASTERROR
```

```
  exit
```

```
end
```

```
if(result > 0) then /* Found Users */
```

```
do
```

```
  do n = 1 to SRESULT.COUNT
```

```
    say 'User nr:' SRESULT.n.USERNR ' In tag:',
```

```
      SRESULT.n.FOUNDINTAG
```

```
    end
end
else
do
    say 'No users found'

    if (symbol("SUGSTEM.COUNT") = "VAR") then
do
    say 'Suggestion list:'

    do n=1 to SUGSTEM.COUNT
        say'  Name:' SUGSTEM.n 'Number:' SUGSTEM.n.USERNR
    end
end
end
end

NOTES

BUGS

SEE ALSO
```

1.34 BBSREAD/SORTBBSLIST

```
NAME
SORTBBSLIST -- Sort bbs list.

SYNOPSIS
SORTBBSLIST

FUNCTION
Funtion to sort the bbs list.

INPUTS

RESULT
Returns: 0 - Success.
        30 - Failure.

EXAMPLE

NOTES

BUGS

SEE ALSO
```

1.35 BBSREAD/SORTCONFLIST

```
NAME
SORTCONFLIST -- Sort the conference list for a bbs.
```

SYNOPSIS
SORTCONFLIST BBSNAME/A

FUNCTION
Function to sort the conference list of a bbs.

INPUTS
BBSNAME - Name of bbs.

RESULT
Returns: 0 - Success.
5 - BBS name was not found.
30 - Failure.

EXAMPLE

NOTES

BUGS

SEE ALSO

1.36 BBSREAD/SORTFAREALIST

NAME
SORTFAREALIST -- Sort file area list for a bbs.

SYNOPSIS
SORTFAREALIST BBSNAME/A

FUNCTION
Function to sort the file areas of a bbs.

INPUTS
BBSNAME - Name of bbs.

RESULT
Returns: 0 - Success.
5 - BBS name was not found.
30 - Failure.

EXAMPLE

NOTES

BUGS

SEE ALSO

1.37 BBSREAD/UNIQUEMSGFILE

NAME

UNIQUEMSGFILE -- Create a unique message file (for events)

SYNOPSIS

UNIQUEMSGFILE BBSNAME/A,STEM/A,EXTENSION/K,USETAG/K/N,FROMFILE/K

FUNCTION

Creates a unique message file to be used with FILENAME and DETAILEDFILEDESCR when creating events. The file is created in the directory for the given bbs. If the type of the bbs has INITMSGFILE set, is the command used to initialize the file. The file information is returned in stem variables. The tag stem variables returned are:

<stem>.NAME - Complete path and name of created file.
<stem>.FILEPART - The file part of the name returned. Must be used in FILENAME and DETAILEDFILEDESCR.

INPUTS

BBSNAME - Name of bbs.

STEM - Base name of the stem variables the file information is returned in.

EXTENSION - Extension to use for the file. If this parameter isn't given is "msg" used as extension.

USETAG - Tag to use the message file in. Defined are:

UT_MSGFILE = 1

UT_DETAILEDFILEDESCR = 2

Default is MSGFILE.

FROMFILE - The contents of this file is added to the created file.

RESULT

Returns: 0 - Success.
5 - BBS name was not found.
30 - Failure.

EXAMPLE

address BBSREAD

```
UNIQUEMSGFILE bbsname ' "Ultima" ' stem STEM
```

```
if(rc ~= 0) then
```

```
do
```

```
    say BBSREAD.LASTERROR
```

```
    exit
```

```
end
```

```
say 'Complete filename:' STEM.NAME
```

```
say 'Filepart:         ' STEM.FILEPART
```

```
/* The file must be deleted if it isn't used in an event */
```

```
address command 'Delete >nil: '||' '||STEM.NAME||' '||' '||'
```

NOTES

If you for some reason don't use the returned filename in any event, should you delete the file.

BUGS

SEE ALSO

1.38 BBSREAD/UPDATEBREVENT

NAME

UPDATEBREVENT -- Updates the flags of an event.

SYNOPSIS

UPDATEBREVENT BBSNAME/A, EVENTNR/A/N, SETDELETED/S, CLEARDELETED/S,
SETPACKED/S, CLEARPACKED/S, SETERROR/S, CLEARERROR/S, SETDONE/S,
CLEARDONE/S, SETFREEZE/S, CLEARFREEZE/S

FUNCTION

Lets you update the flags of an event. This command sets the BDF_EVENT_CHANGED flag for the bbs. The priority of the flags are: (highest to lowest) EDF_DELETED, EDF_DONE, EDF_ERROR, EDF_FREEZE, EDF_PACKED.

INPUTS

BBSNAME - Name of the bbs.

EVENTNR - Number of the event to update flags for.

SETDELETED - Set the EDF_DELETED flag. Clears EDF_PACKED and EDF_ERROR flags.

CLEARDELETED - Clear the EDF_DELETED flag.

SETPACKED - Set the EDF_PACKED flag. This flag is for use against BBS'es where events are sent in packages. Set this flag when the event is packed into the package. Then use this flag to delete the events done when the package is successfully sent. This makes it possible to repack a package easy without loosing any events.

CLEARPACKED - Clear EDF_PACKED flag.

SETERROR - Set EDF_ERROR flag. Set this flag if the event couldn't be done. While this flag is set should event packers ignore this event. Clears the EDF_PACKED flag.

CLEARERROR - Clear EDF_ERROR flag.

SETDONE - Set EDF_DONE flag. This flag should be set when an event is successfully executed. Events marked with this flag is removed from the datafile when the eventdata is packed.

CLEARDONE - Clear EDF_DONE flag.

SETFREEZE - Set EDF_FREEZE flag. When his flag is set should event pakkers ignore this event. Clears the EDF_PACKED flag.

CLEARFREEZE - Clear EDF_FREEZE flag.

RESULT

Returns: 0 - Success.

5 - BBS name was not found.
30 - Failure.

```
EXAMPLE
address "BBSREAD"

/* Mark event 1 as deleted */

UPDATEBREVENT bbsname ' "Ultima"' eventnr 1 SETDELETED
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

NOTES

BUGS

SEE ALSO

1.39 BBSREAD/UPDATEBRMESSAGE

NAME

UPDATEBRMESSAGE -- Updates the flags of a message.

SYNOPSIS

```
UPDATEBRMESSAGE BBSNAME/A, CONFNAME/A, MSGNR/N/A, SETDELETED/S,
  CLEARDELETED/S, SETKEEP/S, CLEARKEEP/S, SETREPLIED/S, CLEARREPLIED/S,
  SETURGENT/S, CLEARURGENT/S, SETIMPORTANT/S, CLEARIMPORTANT/S,
  HAZELEVEL/N/K, SETCONFIDENTIAL/S, CLEARCONFIDENTIAL/S
```

FUNCTION

Lets you update the flags of a message.

INPUTS

BBSNAME - Name of bbs.
CONFNAME - Name of conference.
MSGNR - Number of message to change flag for.

SETDELETED - Set MDF_DELETED flag. Also unmarks the message if it is marked.

CLEARDELETED - Clear MDF_DELETED flag. Fails if MDF_UNRECOVERABLE flag is set.

SETKEEP - Set MDF_KEEP flag.
CLEARKEEP - Clear MDF_KEEP flag.
SETREPLIED - Set MDF_REPLIED.
CLEARREPLIED - Clear MDF_REPLIED flag.
SETURGENT - Set MDF_URGENT flag.
CLEARURGENT - Set MDF_URGENT flag.
SETIMPORTANT - Set MDF_IMPORTANT flag.
CLEARIMPORTANT - Clear MDF_IMPORTANT flag.

HAZELEVEL - Haze Level to set for message. Possible haze levels are 0 to 3.

- 0 - No hazing.
- 1 - The message is kept at least as long as GLOBALDATA.HAZELEVEL1 seconds.
- 2 - The message is kept at least as long as GLOBALDATA.HAZELEVEL2 seconds.
- 3 - The message is kept at least as long as GLOBALDATA.HAZELEVEL3 seconds.

SETCONFIDENTIAL - Set MDF_CONFIDENTIAL flag.

CLEARCONFIDENTIAL - Clear MDF_CONFIDENTIAL flag.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 7 - Conference name was not found.
 30 - Failure.

EXAMPLE

```
address "BBSREAD"
```

```
/* Mark message 7 in conference with keep flag */
```

```
UPDATEBRMESSAGE bbsname "Ultima" confname "THOR/Generelt",
  msgnr 7 SETKEEP
if(rc ~= 0) then
do
  say BBSREAD.LASTERROR
  exit
end
```

NOTES

BUGS

SEE ALSO

1.40 BBSREAD/VERSION

NAME

VERSION -- Returns version information. (V4)

SYNOPSIS

VERSION STEM

FUNCTION

Command to return version information. If the stem variable is given will additional version information bereturned in stem variables. The stem variables returned are:

<stem>.AREXXHANDLER.VERS - Version number for BBSREAD Arexx handler.
 <stem>.AREXXHANDLER.REV - Revision number for BBSREAD Arexx handler.

<stem>.BBSREADLIBRARY.VERS - Version number for bbsread.library.
 <stem>.BBSREADLIBRARY.REV - Revision number for bbsread.library.

INPUTS

STEM - Base name of the stem variables the version result should be put in.

RESULT

Returns: 0 - Success. The result variable is set to the version number of the BBSREAD Arexx handler.
 30 - Failure.

EXAMPLE

```
options results
address BBSREAD
```

```
VERSION VER
```

```
say 'Arexx Handler V' || VER.AREXXHANDLER.VERS || '.' ||,
    VER.AREXXHANDLER.REV
```

```
say 'bbsread.library V' || VER.BBSREADLIBRARY.VERS || '.' ||,
    VER.BBSREADLIBRARY.REV
```

NOTES

BUGS

SEE ALSO

1.41 BBSREAD/WRITEBREVENT

NAME

WRITEBREVENT -- Write an event to the database.

SYNOPSIS

```
WRITEBREVENT BBSNAME/A,EVENT/N/A,STEM/K,UPDATEEVENTNR/K/N
```

FUNCTION

Adds an event to a bbs. The event tags are passed in stem variables.

Available eventtypes and what tags that are allowed to use with the different event types are defined in the BBSType for the BBS. Be sure to pass all the NeedTags with appropriate values. This function sets the BDF_EVENTS_CHANGED flag for this BBS.

The event types defined are:

```
EVE_ENTERMSG      = 0 - Enter message
EVE_REPLYMSG      = 1 - Reply message
EVE_JOINCONF      = 2 - Join conference
EVE_RESIGNCONF    = 3 - Resign conference
EVE_DOWNLOAD      = 4 - Download file
EVE_UPLOAD        = 5 - Upload file
EVE_CONFLIST      = 6 - Get conflist
EVE_CONFIGUREBBS = 7 - Set correct configuration on the bbs.
```

```

EVE_SENDUSERINFO = 8 - Send userinfo
EVE_GETMESSAGES = 9 - Get specified messages
EVE_GETNEWFILES = 10 - Get new files listing
EVE_DOCOMMAND = 11 - Command the script should do
EVE_AUTOLOGOFF = 12 - Autologoff on next session
EVE_RETRACTMSG = 13 - Retract message from bbs
EVE_RETRACTFILE = 14 - Retract file from bbs

```

The tag stem variables understood are:

```

<stem>.TONAME - Name to send to.
<stem>.TOADDR - Address to send to.
<stem>.SUBJECT - Subject of message. Must not be any longer
                than defined for the BBSType of the bbs.
<stem>.CONFERENCE - Name of conference.
<stem>.REFNR - The number of the message to reply. The
                message number is the number this message
                has in the local database.
<stem>.REFORIGINALNR - Number of the message on BBS to reply to.
<stem>.REFID - The idstring of the message to reply to.
<stem>.MSGFILE - Name of file with text. Filename is
                relative to the BBSPATH returned by

```

GETBBSDATA

. Each line in this file should not be longer than the LINELENGTH defined for this bbs/bbstype.

```

<stem>.PRIVATE - Message should be flagged as private if
                not 0.
<stem>.LOCALFILE - Path and name of local file.
<stem>.DIRECTORY - Directory for file to up/down load. This
                is the remote directory.
<stem>.FILENAME - Name of file to upload/download. The file
                name must not be longer than FILENAMELEN
                defined for the type of the bbs.
<stem>.DOWNLOADNOTIFY - Notify on download if non 0.
<stem>.FILEDESCR - Short file description. The description
                must not be longer than FILEDESCRLEN
                defined for the type of the bbs.
<stem>.DETAILEDFILEDESCR - Name of a file with detailed file
                description. Filename is relative to the
                BBSPATH returned by

```

GETBBSDATA

. Each line in this file should not be longer than the LINELENGTH defined for this bbs/bbstype.

```

<stem>.FROMMESSAGENR - Remote message number to start at.
<stem>.TOMESSAGENR - Remote message number to end at.
<stem>.COMMANDSTRING - String with command to execute at bbs.
<stem>.BOOLEAN - Boolean value.
<stem>.DATE - Date in seconds since 1.January 1978
<stem>.PGPSIGNID - Id for key to sign message with. It's
                preferable that the keyid is used to identify a key. (e.g.
                0x5B4231FD) Using a '*' signs with the first key in the
                secret keyring.
<stem>.PGPENCRYPTID - Id for key(s) to encrypt message with.

```

It's preferable that the keyid is used to identify a key.
(e.g. 0x5B4231FD) If more than one key are used to encrypt should
the id for each key be separeated with a space.

<stem>.REFCONFERENCE - Name of the conference of the message to reply
to. This tag is to be used when replying messages to other
conferences than the conference of the original message. Should only
be used when the bbs type has the TDF_GLOBAL_REPLIES flag set.

<stem>.URGENT - Flag event as urgent. Boolean value. Default is 0
(FALSE).

<stem>.IMPORTANT - Flag event as important. Boolean value. Default is
0 (FALSE).

<stem>.CONFIDENTIAL - Flag event as confidential. Boolean value.
Default is 0 (FALSE).

<stem>.RETURNRECIEPT - Set return reciept flagg for event. Boolean
value. Default is 0 (FALSE).

<stem>.ENCODE8BIT - Encode outgoing text if it is 8 bit. Boolean
value. Default is 0 (FALSE).

INPUTS

BBSNAME - Name of bbs.

EVENT - The event type to write.

STEM - Base name of the stem variables the tag data is in. Must be
supplied when adding or updating the tags for a event.

UPDATEEVENTNR - Update the event with this eventnumber. When updating
tags, all old tags are discarded.

RESULT

Returns: 0 - Success. The result variable is set to the number of
the written event.

5 - BBS name was not found.

30 - Failure.

EXAMPLE

options results

address BBSREAD

EVE_UPLOAD = 5 /* Upload file */

drop EVENT. /* Important */

EVENT.FILENAME = 'QToolsR1.LHA'

EVENT.FILEDESCR = 'The best tool for doing quick searches and CD'

EVENT.LOCALFILE = 'work:qtools/qtoolsr11.lha'

WRITEBREVENT bbsname '"Ultima"' event EVE_UPLOAD stem EVENT

if(rc ~= 0) then

do

say BBSREAD.LASTERROR

exit

end

```
say 'Eventnumber: ' result
exit
```

NOTES

BUGS

SEE ALSO

1.42 BBSREAD/WRITEBRFILE

NAME

WRITEBRFILE -- Write an entry to the file database.

SYNOPSIS

```
WRITEBRFILE BBSNAME/A,FAREANAME/A,STEM/K,UPDATEFILENR/K/N,
DELETEFILE/S
```

FUNCTION

Writes an entry to a file area in the file database for a bbs. The file entry tags are passed in stem variables. The tag stem variables understood are:

<stem>.NAME - The name the file is identified with on the bbs. This tag must be supplied when adding or updating file tags.

<stem>.DATE - The date the file has on the bbs. The date is in seconds since 1.January 1978.

<stem>.SIZE - The size of the file. (bytes)

<stem>.DOWNLOADS - How many times the file has been downloaded from the bbs.

<stem>.DESCRIPTION.COUNT - Number of lines in description.

<stem>.DESCRIPTION.<l-count> - The text lines in the description. No newline characters are allowed.

INPUTS

BBSNAME - Name of bbs.

FAREANAME - File area to write the file entry to.

STEM - Base name of the stem variables the tag data is in. Must be supplied when adding or updating the tags for a file entry.

UPDATEFILENR - Update the file with this filename. When updating tags is all old tags discarded.

DELETEFILE - Mark the file in UPDATEFILENR as deleted.

RESULT

Returns: 0 - Success. The result variable is set to the number of the written file.

5 - BBS name was not found.

6 - File area was not found.

30 - Failure.

EXAMPLE

```

options results

address "BBSREAD"

drop FILE. /* Important */

FILE.NAME = 'QToolsR1.LHA'
FILE.DATE = 497404800 /* Equals: 6.October 1993 */
FILE.SIZE = 26789
FILE.DOWNLOADS = 4
FILE.DESCRPTION.COUNT = 2
FILE.DESCRPTION.1 = 'QuickTools Release 1'
FILE.DESCRPTION.2 = 'The best tool for doing quick searches and CD'

WRITEBRFILE bbsname '"Ultima"' fareaname '"UTS"' stem FILE
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

say 'Filenumber: ' result
exit

NOTES

BUGS

SEE ALSO

```

1.43 BBSREAD/WRITEBRIEFMSG

NAME

WRITEBRIEFMSG -- Write a message to a file in BRIEF.

SYNOPSIS

WRITEBRIEFMSG FILENAME/A,BBSNAME/A,CONFNAME/A,MSGNR/N/A

FUNCTION

This function writes the specified message to a file using the [B]BS[R]ead [I]nterchang[e] [F]ormat. The message is appended to the end of the file if it exists.

INPUTS

FILENAME - Name of file to write to. If the file doesn't exist will it be created.
 BBSNAME - Name of bbs where the message is.
 CONFNAME - Name of conference where the message is.
 MSGNR - Messagenumber to write to the file.

RESULT

Returns: 0 - Success.
 5 - BBS name was not found.
 7 - Conference name was not found.
 30 - Failure.

```

EXAMPLE
address BBSREAD

/* Write message nr 1 in conference THOR
   on bbs Ultima to a file called ram:Msg */

WRITEBRIEFMSG "ram:msg" "Ultima" "THOR" 1
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

NOTES

BUGS

SEE ALSO

```

1.44 BBSREAD/WRITEBRMESSAGE

NAME

WRITEBRMESSAGE -- Write a message to the database.

SYNOPSIS

```

WRITEBRMESSAGE BBSNAME/A,CONFNAME/A,STEM/K/A,DONTMARKMESSAGE/S,
  PRIVATE/S,READ/S,URGENT/S,IMPORTANT/S,CONFIDENTIAL/S,
  TOFROMUSERSTATUS/K

```

FUNCTION

Writes a message to the database. The message tags are passed in stem variables. The stem variables understood are:

<stem>.FROMNAME - The name of the user this message is written by.
This tag must be supplied.

<stem>.FROMADDR - The address of the user this message is written by.
This is the author's net-address. The name needn't redundantly be repeated in this field, if it's already in FROMNAME. Only use this if the message is in a network conference.

<stem>.TONAME - The name of the user this message is written to. When adding messages with no receiver (to all) don't use this tag.
Same format as FROMNAME.

<stem>.TOADDR - The address of the user this message is written to.
Same format as FROMADDR.

<stem>.MSGID - The message ID string. Use this tag when the message has a string as identifier. This id is considered to be unique.

<stem>.ORIGINALNR - The number this message has on the BBS. Use this tag when the message has a number as identifier.

<stem>.REFID - If this message is an answer, is this the ID string

for the message this message refer to.

- <stem>.REFNR - If this message is an answer, is this the message number for the message this message refers to.
- <stem>.CREATIONDATE - The time the message was created. The time is in seconds since 1.January 1978.
- <stem>.CREATIONDATETXT - The time the message was created. The forming of this string is free. Use this when it is impossible to use CREATIONDATE.
- <stem>.SUBJECT - The subject of the message. This tag must be supplied.
- <stem>.REPLYCONF - The name of the conference a reply to this message should use. If the message is private, should it only be moved if it is possible to have private messages in this conference.
- <stem>.REPLYNAME - Name of the user a reply of this message should go to. Same format as FROMNAME.
- <stem>.REPLYADDR - The address of the user a reply of this message go to. Same format as FROMADDR.
- <stem>.COMMENT.COUNT - Number of lines in comment concerning this message.
- <stem>.COMMENT.<l <= n <= count> - The lines in the comment. No newline characters are allowed in the lines. Header information not used in anoter tag could be put here.
- <stem>.TEXT.COUNT - Number of text lines in the message.
- <stem>.TEXT.<l <= n <= count> - The lines in the message text. No newline characters are allowed. This tag must be supplied. Any information that not belong to the original text must not be put here. The text may be of any size. Lines may be of any length. It is up to the reader program to wrap lines if needed.
- <stem>.PART.COUNT - The number of parts this message consists of. Each part is *either* a binary part, a text part *or* a message part.
- <stem>.PART.n.BINARY - Filename of file containing the binary data of n'th part. The filename should be relative to the BBSPATH returned by
GETBBSDATA
, but absolute paths are allowed.
- <stem>.PART.n.BINARY.DESC - Description for the binary data of the n'th part.
- <stem>.PART.n.BINARY.COMMENT.COUNT - Number of lines in the comment for the binary data of the n'th part.
- <stem>.PART.n.BINARY.COMMENT.m - The m'th line in the comment for the binary data of the n'th part. No newline characters are allowed.
-

<stem>.PART.n.TEXT.COUNT - Number of lines of text in the n'th part.

<stem>.PART.n.TEXT.m - The m'th line of text in the n'th part. No newline characters are allowed. The text may be of any size. Lines may be of any length. It is up to the reader program to wrap lines if needed.

<stem>.PART.n.TEXT.COMMENT.COUNT - Number of lines in the comment for the text of the n'th part.

<stem>.PART.n.TEXT.COMMENT.m - The m'th line in the comment for the text of n'th part. No newline characters are allowed.

<stem>.PART.n.MSG.<msgtags> - Msgtags can be any legal message tag, also PART.n.MSG.

INPUTS

BBSNAME - Name of bbs.
CONFNAME - Name of conference.
STEM - Base name of the stem variables the tag data is in.
DONTMARKMESSAGE - Don't mark message as unread.
READ - Flag message as read by reciever.
URGENT - Flag message as urgent.
IMPORTANT - Flag message as important.
CONFIDENTIAL - Flag message as confidential.
TOFROMUSERSTATUS - The status of the MDF_TO_USER and MDF_FROM_USER flags this message should have. Flags must be given as a hex string. See rexx/examples/GlobalDefs.br for definitions. *When this parameter is used will *not* the sender or receiver be matched against the name and/or address of the user. When using this parameter is the parser responsible for checking if the message is to or from the user.

RESULT

Returns: 0 - Success. The result variable is set to the number of the written message.
5 - BBS name was not found.
7 - Conf area was not found.
30 - Failure.

EXAMPLE

```
options results
address BBSREAD

drop MSG. /* Important */

MSG.FROMNAME = 'Eivind Nordseth'
MSG.TONAME = 'Petter Nilsen'
MSG.ORGINALNR = 1
MSG.SUBJECT = 'This is a test'

MSG.TEXT.COUNT = 2
MSG.TEXT.1 = 'Text line 1'
MSG.TEXT.2 = 'Text line 2'

MSG.PART.COUNT = 3
```



```
MSG.PART.1.BINARY = 'file.ttt'
MSG.PART.1.BINARY.DESC = 'This is the file description'
MSG.PART.1.BINARY.COMMENT.COUNT = 2
MSG.PART.1.BINARY.COMMENT.1 = 'Part binary Comment line 1'
MSG.PART.1.BINARY.COMMENT.2 = 'Part binary Comment line 2'
```

```
MSG.PART.2.TEXT.COUNT = 2
MSG.PART.2.TEXT.1 = 'Part text line 1'
MSG.PART.2.TEXT.2 = 'Part text line 2'
```

```
MSG.PART.3.MSG.FROMNAME = 'Petter Nilsen'
MSG.PART.3.MSG.SUBJECT = 'MsgPart Subject'
```

```
MSG.PART.3.MSG.TEXT.COUNT = 2
MSG.PART.3.MSG.TEXT.1 = 'Msg part text line 1'
MSG.PART.3.MSG.TEXT.2 = 'Msg part text line 2'
```

```
MSG.PART.3.MSG.PART.COUNT = 1
```

```
MSG.PART.3.MSG.PART.1.TEXT.COUNT = 1
MSG.PART.3.MSG.PART.1.TEXT.1 = 'Msg part text part'
```

```
WRITEBRMESSAGE ' "Ultima" ' "Generelt" ' stem MSG URGENT
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end

say 'Msg number: ' result
exit
```

Some examples for splitting addresses in 'Name' and 'Address':

RFC:

```
"Martin Horneffer <maho@balrog.dfv.rwth-aachen.de>"
-> name: "Martin Horneffer"
   address: "maho@balrog.dfv.rwth-aachen.de"
```

```
"horneff@pool.informatik.rwth-aachen.de (Martin Horneffer)"
-> name: "Martin Horneffer"
   address: "horneff@pool.informatik.rwth-aachen.de"
```

```
"horneff@pool.informatik.rwth-aachen.de"
-> name: "horneff"
   address: "horneff@pool.informatik.rwth-aachen.de"
```

FidoNet:

```
"Martin Horneffer at 2:242/7.9"
-> name: "Martin Horneffer"
   address: "2:242/7.9"
```

```
"Joerg Gutzke at 2:242/7"
-> name: "Joerg Gutzke"
   address: "2:242/7"
```

NOTES

BUGS

SEE ALSO

1.45 BBSREAD/WRITEBRUSER

NAME

WRITEBRUSER -- Writes an user entry to the user database.

SYNOPSIS

```
WRITEBRUSER BBSNAME/A, STEM/K, UPDATEUSERNR/K/N, DELETEUSER/S,
  ONLYIFEXIST/S
```

FUNCTION

Writes an entry to the user database for a bbs. The user entry tags are passed in stem variables. The tag stem variables understood are:

<stem>.NAME - The name the user has on this bbs. This stem must be supplied when adding or updating a user.

<stem>.ADDRESS - The net address the user has on this bbs.

<stem>.ALIAS - Alias to use for referring to this user. This alias can be used to look up this user in the database.

<stem>.COMMENT.COUNT - Numer of lines in user comment.

<stem>.COMMENT.<l-count> - The text lines in the description. No newline characters are allowed.

<stem>.ENCODE8BITMSG - Encode 8 bit private messages to this user. Boolean value. Default is 0 (FALSE). Only appliable when the TDF_SUPPORTS_ENCODE_8BIT_MAIL flag is set for the bbstype of this bbs.

<stem>.PGPKEYID - The PGP key ID for the PGP key of this user.

INPUTS

BBSNAME - Name of bbs.

STEM - Base name for the stem variables the tag data is in. Must be supplied when adding or updating the tags for a user entry.

UPDATEUSERNR - Update the user with this usernumber. When updating tags, is all old tags discarded.

DELETEUSER - Mark the user in UPDATEUSERNR as deleted.

ONLYIFEXIST - Only write the user to the database if he don't exist in the database. Is ignored when updating users with UPDATEUSERNR.

RESULT

Returns: 0 - Success. The result variable will be set to the number of the written user.
 5 - BBS name was not found.
 30 - Failure.

EXAMPLE

options results

address "BBSREAD"

drop USER. /* Important */

```
USER.NAME      = 'Eivind Nordseth'
USER.ADDRESS  = 'eivindno@idt.unit.no'
USER.ALIAS    = 'Leo'
USER.COMMENT.COUNT = 2
USER.COMMENT.1 = 'SnailMail: Herman Krag Vei 17-21'
USER.COMMENT.2 = '              7035 Trondheim, NORWAY'
```

```
WRITEBRUSER bbsname '"Ultima"' stem USER
if(rc ~= 0) then
do
  say BBSREAD.LASTEROR
  exit
end
```

say 'Usernumber: ' result

NOTES

BUGS

SEE ALSO

1.46 BBSREAD/WRITEPASSCONFLIST

NAME

WRITEPASSCONFLIST -- Write list to passive conference list.

SYNOPSIS

WRITEPASSCONFLIST BBSNAME/A,STEM/A

FUNCTION

Writes a list of conference names to the passive conference list datafiles. The old list is deleted before writing this new one. The passive conference list is a list of *all* available conferences at the bbs. It should be used when the user want to send a join event.

The list should only be written when the parser has got a list over all conferences at the bbs.

The list of conference names are passed in stem variables. The list is expected to be in the following stem variables:

<stem>.COUNT - Number of conferences in the list.

<stem>.<1 <= n <= count>.NAME - The conference name.
<stem>.<1 <= n <= count>.DESCR - Conference description. If this stem variable is empty or omitted, is no conference description written for this conference.

<stem>.<1 <= n <= count>.BBSCONFNR - The internal number this conference has on the bbs. If this stem variable is omitted, is this value set to -1.

<stem>.<1 <= n <= count>.FLAGS - Flags for conference. Same as defined for GETCONFDATA.FLAGS. If this stem value is omitted, is no flags set.

INPUTS

BBSNAME - Name of bbs.
STEM - Base name for the stem variables used to pass the conference list.

RESULT

Returns: 0 - Success.
5 - BBS name was not found.
30 - Failure.

EXAMPLE

```
address "BBSREAD"
```

```
CDF_MEMBER_OF = '00000001'x /* Member of this conference. */
```

```
LIST.1.NAME = 'Thor/Dev'  
LIST.2.NAME = 'Thor/Beta'  
LIST.2.DESCR = 'Thor beta testing conference'  
LIST.2.BBSCONFNR = 13  
LIST.2.FLAGS = CDF_MEMBER_OF  
LIST.3.NAME = 'Thor/Support'  
LIST.3.DESCR = 'Conference for registered Thor users'  
LIST.4.NAME = 'Thor/Generelt'  
LIST.COUNT = 4
```

```
WRITEPASSCONFLIST bbsname ' "Ultima" ' stem LIST  
if(rc ~= 0) then  
do  
    say BBSREAD.LASTERROR  
    exit  
end
```

NOTES

BUGS

SEE ALSO