

# **ARexxCommand**

**COLLABORATORS**

	<i>TITLE :</i> ARexxCommand		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 27, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>ARexxCommand</b>	<b>1</b>
1.1	Contents . . . . .	1
1.2	Syntax Information . . . . .	1
1.3	Command notation convention . . . . .	2
1.4	Return Codes . . . . .	3
1.5	commands . . . . .	3
1.6	addsearch . . . . .	5
1.7	closewin . . . . .	5
1.8	clrec . . . . .	5
1.9	Cycle To String . . . . .	6
1.10	delrec . . . . .	6
1.11	dosearch . . . . .	7
1.12	freeindex . . . . .	7
1.13	freeview . . . . .	7
1.14	getfield . . . . .	8
1.15	getrec . . . . .	8
1.16	goto . . . . .	9
1.17	insrec . . . . .	9
1.18	loadview . . . . .	10
1.19	newsearch . . . . .	10
1.20	next . . . . .	11
1.21	numrecs . . . . .	11
1.22	openfile . . . . .	12
1.23	openwin . . . . .	12
1.24	putfield . . . . .	12
1.25	query . . . . .	13
1.26	refresh . . . . .	14
1.27	report . . . . .	14
1.28	reqchoice . . . . .	15
1.29	reqfield . . . . .	15

---

---

1.30 reqmsg . . . . .	15
1.31 reqstring . . . . .	16
1.32 savefile . . . . .	16
1.33 selectrec . . . . .	16
1.34 setfile . . . . .	17
1.35 setindex . . . . .	17
1.36 setview . . . . .	18
1.37 showall . . . . .	18
1.38 sort . . . . .	19
1.39 String To Cycle . . . . .	19
1.40 updrec . . . . .	19
1.41 wintofront . . . . .	20

---

# Chapter 1

## ARexxCommand

### 1.1 Contents

QuickFile ARexx Command interface

QuickFile v3.17.2 24 April 1996 Copyright Alan Wigginton

QuickFile accepts ARexx commands from macros (ARexx scripts invoked from within QuickFile) and from external scripts which may be run from another application or from the shell or workbench.

QuickFile's ARexx port is named QUICKFILE.01 for the first copy of QuickFile running, QUICKFILE.02 for the next and so on. As QuickFile can open multiple files, it is simplest to use one copy.

Not all QuickFile functions can be executed via an ARexx command. If you would like to see some other commands added, let me know and I will see what I can do.

Syntax Notes

Notation for command description

Return Codes

Command Reference

### 1.2 Syntax Information

QuickFile expects commands to consist of a command name followed by a series of parameters separated by spaces. Parameters that contain spaces must be enclosed in quotes, therefore two sets of quotes are required in ARexx code as ARexx will remove the first set of quotes when it parses the

---

command.

For example, to pass the following command to QuickFile

```
ReqMsg 'This is a demo message'
```

the ARexx code required is

```
"ReqMsg 'This is a demo message'"
```

Both ARexx and QuickFile will accept either double or single quotes.

You should also include quotes around values included from variables if the variable value could contain spaces. For example:

```
pull filename  
address quickfile.01 "openfile '"filename'"
```

If filename was entered as "My file" this would result in the following being (correctly) passed to quickfile:

```
openfile 'My file'
```

If the single quotes had been omitted this would have caused an error. The easiest way to check the string being passed is to display it using say. This will usually make things clear.

For simplicity avoid using spaces in field names. While they can be used, you are prevented from using the GetRec command which uses field names to form compound variables.

QuickFile will process commands regardless of case. Case is preserved in field values.

Confusion with case is a common problem with ARexx. ARexx always converts any tokens to upper case unless they are in quotes. A common trap is that the PULL and ARG statements translate to upper case as they are equivalent to PARSE UPPER PULL and PARSE UPPER ARG. To preserve case use PARSE PULL and PARSE ARG instead.

### 1.3 Command notation convention

The following convention is used in the command descriptions

Command names use Initial Capitals eg GetRec

Values in square brackets [ ] are optional

You must include one of the values in braces {A | D}

Repeated values are indicated by ellipsis ...

Keywords are shown in UPPER CASE

---

Lower case words represent values you provide

## 1.4 Return Codes

The return codes from QuickFile have the following general meanings. See the command descriptions for details

```
5 Warning or object not found.
10 Command failed
12 Unknown command
15 System error - usually memory allocation error
20 Syntax error
```

## 1.5 commands

AddSearch

OpenWin

CloseWin

PutField

ClrRec

Query

CycleToString

Refresh

DelRec

Report

DoSearch

ReqChoice

FreeIndex

ReqMsg

FreeView

ReqString

GetField

SaveFile

GetRec

SetFile

GoTo

SetIndex

InsRec

SetView

LoadView

Sort

NewSearch

StringToCycle

Next

UpdRec

NumRecs

WintoFront

OpenFile

New in version 3.23

ReqField

SelectRec

ShowAll



## 1.6 addsearch

```
AddSearch {AND | OR} fieldname operator value1 [value2]
```

Adds a new search criteria to the existing criteria. Use the `NewSearch` command to replace any existing criteria and use `DoSearch` to start the search.

`AND | OR` Determines how multiple criteria are to be handled. See `QuickFile` docs for full details.

See the `NewSearch` command for details of the remaining parameters.

Results  
None

Return codes  
10 Unknown field or operator  
20 Insufficient parameters or first parameter is not AND or OR

Example  
/\* select records where surname is brown or smith \*/  
"newsearch surname equal brown"  
"addsearch or surname equal smith"  
"dosearch"

## 1.7 closewin

CloseWin

Closes the current window and file. You cannot close the last window with an ARexx command. The first window in `QuickFiles` internal list becomes the current window,

Results  
None

Return codes  
5 You tried to close the last window.

Example  
"closewin"

## 1.8 clrrec

ClrRec

---

Clears all fields in the current record. Use this to clear all fields before setting up the values for a new record

Results  
None

Return codes  
None

Example  
"ClrRec"  
"putfield surname Brown"  
"putfield firstname John"  
"putfield address '23 George Street'"  
"insrec"

## 1.9 Cycle To String

CycleToString field string

Finds the string corresponding to a string number for a cycle field. The cycle field value is not changed.

Results  
String from the cycle field.

Return codes  
5 String number out of range for cycle field  
10 Unknown field name or not a cycle field  
20 Parameter missing

Example  
given a cycle field called Status with "Active,Pending,Cancelled"  
  
"stringtocycle Status 2"  
say result /\* ==> Cancelled \*/

## 1.10 delrec

DelRec

Deletes the current record. The next record becomes the new current record.

Results  
None

Return codes  
5 There are no records in the file.

Example  
"getfield expirydate"

---

```
    if result < today then
"delrec"
```

## 1.11 dosearch

DoSearch

Searches through the data base and selects records matching the search criteria. The criteria are established using the

```
    NewSearch
    and
```

```
    AddSearch
    commands.
```

It places the matching records in an index named SELECTED. It does not change the current index.

Results

```
    RESULT contains the number of records matched.
```

Return codes

```
    None
```

Example

```
    "newsearch surname equal brown"
    "addsearch or surname equal smith"
    "dosearch"
    "setindex selected"
```

## 1.12 freeindex

FreeIndex

Discards the current index and frees the storage used. Only temporary indexes, ie those named SORTED and SELECTED, can be freed.

Results

```
    None
```

Return codes

```
    10 Not a temporary index
```

Example

```
    "setindex selected"
    "freeindex"
```

## 1.13 freeview

---

## FreeView

Discards the current view from memory and frees associated storage. The first view in the loaded list becomes the current view. The display is redrawn. You must keep at least one view. It does not affect the disk file containing the view.

### Result

None

### Return codes

5 You tried to free the last view

### Example

```
"setview mylist.view"
"freeview"
```

## 1.14 getfield

GetField field

Returns value of field from the current record. See also

GetRec

which

obtains all fields for the current record.

### Result

Field value is placed in RESULT.

### Return Codes

10 Field requested does not exist

20 Field name omitted

### Example

```
"goto Smith"
"getfield surname"
say "Found name" result /* Displays==> Found name Smith */
```

## 1.15 getrec

GetRec stem

Returns values for all fields from the current record in compound variables in the form of stem.fieldname where fieldname is the field name from the file definition (not always the title in the view).

You cannot use this if your field names contain spaces; use

GetField

instead.

### Results

```
stem.fieldname variables updated.
```

Return codes

```
20 Syntax error. stem not provided.
```

Example

```
"next"  
"getrec val"  
say val.firstname val.surname
```

## 1.16 goto

GoTo value1 [value2...]

Sets the current position to the first record with the current key matching the specified values. Value1 is the value for index field 1 and Value2 is the value for index field 2 etc.

QuickFile does not tell you if the requested record was not found. It will happily position on the next record. You must compare the record with the requested key values to check if they were found.

Result

```
None
```

Return Codes

```
5 End of file encountered. Requested key higher than any record.  
10 More values specified than index fields
```

Example

```
"goto Brown John"  
"getrec val"  
if val.surname ~= "Brown" | val.firstname ~= "John" then  
say "Record not found"
```

## 1.17 insrec

InsRec

Adds a new record to the file. You use

```
PutField  
to set the field values
```

before issuing InsRec. Any fields not updated with PutField will retain the values from the previous record. Use

```
ClrRec  
to clear all values
```

before setting up your new record.

Results

```
None
```

Return codes

---

10 Error occurred. Probably duplicate key.

Example

```
"ClrRec"          /* omit this to use values from previous record */
"putfield surname Brown"
"putfield firstname John"
"putfield address '23 George Street'"
"insrec"
```

## 1.18 loadview

LoadView path

Loads the named view. The full name including extension must be specified. This does not change the display. Use the

```
SetView
command to change the
```

current view.

Result

None

Return codes

```
10 File not found
10 path name not specified
```

Example

```
"loadview df0:QuickFile/Example1/Product.view"
```

## 1.19 newsearch

NewSearch fieldname operator value1 [value2]

Replaces any existing search criteria with a new one. Use the

```
AddSearch
command to add additional criteria, if required, and the
DoSearch
command to start the actual search.
```

operator The search operator. One of the following:

```
like
equal
between
notlike
notequal
sounds
```

value1 The value to be searched for

value2 The second value required for the between operator. Can also be used for others. See QuickFile docs for more details.

Results  
None

Return codes  
10 Unknown field or operator  
20 Insufficient parameters

Example  
/\* find all current memberships that expire in March \*/  
"newsearch status equal current"  
"addsearch and 'expiry date' between 01-mar-95 31-mar-95"  
"dosearch"

## 1.20 next

Next [n]

Moves n records forward or backward through the file. If n is omitted 1 is assumed. If the resulting position is outside the range of the file, the position will be the first or last record. n can be positive or negative.

Result  
None.

Return codes  
5 No more records. Positioned at first or last record

Example  
"Next -9999" /\* move to start of file \*/  
"Next" /\* move to second record \*/

## 1.21 numrecs

NumRecs

Returns number of records in the current index. If 'Selected' is the current index, this will be the number of selected records.

Result  
Number of records in RESULT

Return codes  
10 No file open

Example  
"SetIndex name"  
"NumRecs"  
say result /\* ==> 253 \*/  
"SetIndex selected"  
say result /\* ==> 16 \*/

## 1.22 openfile

OpenFile pathname

Opens a new file in the current window. Closes the existing file, if any. To open an additional file, issue the

OpenWin  
command first to obtain a

new window.

Result

None

Return codes

10 The file could not be opened. NB Any previous file will have been closed.  
20 pathname not specified

Example

"openfile 'ram disk:Membership'" /\* care with spaces in path names \*/

## 1.23 openwin

OpenWin

Opens a new window which becomes the current window. It will have no open file. Until a file is opened, the only commands that can be issued are

OpenFile  
and  
CloseWin  
.

Result

None

Return Codes

10 The window could not be opened

Example

pull filename  
"openwin"  
"openfile '"filename"' " /\* quotes in case filename contains spaces \*/

## 1.24 putfield

PutField field value

Sets the contents of field for the current record to value. The change will not take effect until you issue the

UpdRec  
command. value must be



consistent with the field's type.

For a cycle field the value should contain a string matching one of the values for the cycle field. Matching is not case sensitive.

Result

None

Return codes

5 The value is inconsistent with the field type. The field is not updated

10 The field does not exist

20 Either field or value were omitted

Example

```
"next"  
newdate = "31-Mar-1996"  
"putfield 'expiry date' " newdate  
"updrec"
```

## 1.25 query

Query {FIELD | INDEX} stem [name]

Obtains details about the current file.

stem A stem variable to receive the details

name An optional name of an object to be queried. If omitted, a summary of all objects of the requested type are returned.

Results

FIELD without name

Returns the number of fields in stem.0 and the field names in stem.1 to stem.n where n is the number of fields.

FIELD with name

Returns field type in stem.type and the maximum length in stem.length. Name is a field name

INDEX without name

Returns the number of indexes in stem.0 and the index names in stem.1 to stem.n

INDEX with name

Returns the number of index fields in stem.0 in the field names in stem.1 to stem.n. Name is an index name.

Return codes

10 Requested name does not exist

15 Syntax error. Either a required field was omitted or type was not FIELD or INDEX.

Example

---

```
"query field fld"
do i = 1 to fld.0 /* displays all field names */
say fld.i
end
```

## 1.26 refresh

Refresh

Redraws the screen display. This should not be required in a macro, but will be required if you want to update the display to reflect changes made from an external script.

Result

None

Return codes

None

Example

```
"refresh"
```

## 1.27 report

Report count [target [title]]

Produces the report defined for the current view. Writes 'count' records starting at the current record. The parameters are positional so 'title' cannot be specified without 'target'.

count Number of records to process. Specify -1 for all.

target Specify 'printer' to write report to the printer, otherwise the report is written to the screen.

title Up to 50 characters to be used instead of the title defined for the report. Don't forget the extra quotes.

Note that the report definition is not changed.

Result

None

Return codes

```
10 No report has been defined for the current view.
15 Could not allocate memory.
```

Example

```
"report -1 printer 'Report Title over-ride'" /* note quotes */

"report 10 screen"
```

## 1.28 reqchoice

ReqChoice line1 [line2]

Displays a requester that displays two lines of text with OK and CANCEL buttons. Don't forget the extra set of quotes.

Result

None

Return codes

0 User chose OK.

5 User chose CANCEL.

20 No message text specified

Example

```
"reqchoice 'Delete requested' 'Are you sure' "  
  if rc = 0 then  
    "delrec"  
  else  
    say "Delete cancelled"
```

## 1.29 reqfield

ReqField

Displays a requester including the names of the fields of the current base. If the user clicks on a field name then OK, the name of the selected field is returned. A double click is equivalent to a click on the OK gadget.

Result

RESULT will contain the selected field name.

Return codes

0 User pressed OK

5 User pressed CANCEL

20 An error occurred (the requester was not opened)

Example

```
"ReqField"  
  if rc = 0 then  
    fieldname = result
```

## 1.30 reqmsg

ReqMsg message

Displays the message in a requester. The message must be enclosed in quotes if it is more than one word.

Result

None

---

Return codes

20 No message specified

Example

```
"ReqMsg 'This message from ARexx script'"
```

### 1.31 reqstring

ReqString default [title]

Displays a requester with a string gadget to obtain input from the keyboard.

default The default value to be placed in the string gadget. Specify ' ' if no default

title The title to be displayed in the requester window.

Result

RESULT will contain the users input string

Return codes

0 User pressed OK  
5 User pressed CANCEL  
20 No message specified

Example

```
"reqstring ' ' 'Please enter your name'"  
if rc = 0 then  
name = result
```

### 1.32 savefile

SaveFile

Writes any updates to disk, if any changes have been made.

Result

None

Return codes

None

Example

```
"savefile"
```

### 1.33 selectrec

---

### SelectRec

This command selects the current record if the display is using ALL records, in the other case (selected records only) the current record is unselected. You can use the

```
ShowAll
    command to switch from one mode to the other.
```

Note: you must use the

```
Refresh
    command to update the display if you want it
to be updated before the end of the script.
```

Result  
None

Return codes  
None

Example  
"SelectRec"

## 1.34 setfile

SetFile [filename]

Selects window containing filename as the current window or returns current file name if 'filename' is not specified.

The file must have been opened previously. 'Filename' is the file name part only, not including disk and directory names.

Results  
Current file name if issued without filename, otherwise none.

Return code  
5 Unknown file name

Example  
"setfile"  
say result  
  
"setfile Images"

## 1.35 setindex

SetIndex [indexname]

Makes indexname the current index. Specify a name as SORTED to use the last sort sequence and SELECTED to use the last selection.

---

Returns name of the current index if issued without 'indexname'.

#### Results

Name of current index if issued without name, otherwise none.

#### Return codes

5 Could not find index

#### Example

```
"setindex sorted"
```

## 1.36 setview

SetView [viewname]

Makes the named view the current view and redraws the display using the new view. The viewname must include the extension, if any.

If 'viewname' is not specified, returns the name of the current view.

#### Result

Name of the current view if issued without 'viewname', otherwise none.

#### Return codes

5 Could not find view

#### Example

```
"setview namelist.view"
```

## 1.37 showall

```
ShowAll {ON | OFF}
```

This command is equivalent to using the All/Sel gadget.

OFF Show selected records only

ON Show all records

Note: you must use the

```
Refresh
```

command to update the display if you want it to be updated before the end of the script.

#### Result

None

#### Return codes

0 command OK

20 Parameter missing

#### Example

```
"ShowAll ON"
```

---

## 1.38 sort

```
Sort field1 {A | D} [field2 {A | D} ] ...
```

Creates a new index named SORTED in the specified sequence and makes 'sorted the current index.

field The field name to sort over

order A for ascending, D for descending

Both field and order must be specified for each sort field. You can switch between indexes, including SORTED and SELECTED using

```
SetIndex
```

.

Result

None

Return codes

10 One of the sort fields was not known.

20 Either no parameters specified or there was not an even number of parameters

Example

```
"sort country a surname a firstname a"
```

## 1.39 String To Cycle

```
StringToCycle field string
```

Converts a string to a cycle field string number. The numbering starts at zero. The cycle field value is not changed.

Results

Cycle field string number.

Return codes

5 String not found in cycle field

10 Unknown field name or not a cycle field

20 Parameter missing

Example

```
given a cycle field called Status with "Active,Pending,Cancelled"
```

```
"stringtocycle Status 'cancelled' "
```

```
say result /* ==> 2 */
```

## 1.40 updrec

```
UpdRec
```

---

Commits an updated record to the file. The update is not necessarily written back to disk at this point.

Results  
None

Return codes  
10 Internal error occurred. Possibly a duplicate key occurred

Example  
"putfield ExpiryDate 31-Mar-95"  
"updrec"

## 1.41 wintofront

WinToFront

Moves the current window to the front of the display.

Results  
None

Return codes  
None

Example  
"wintofront"

---