

# **Skyline 2.06 ARexx Port**

D.F. Duck

Copyright © CopyrightÂ©1993 OmniLink Corporation

---

**COLLABORATORS**

	<i>TITLE :</i> Skyline 2.06 ARexx Port		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	D.F. Duck	August 2, 2022	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1 Skyline 2.06 ARexx Port</b>	<b>1</b>
1.1 Skyline 2.06D ARexx Port	1
1.2 DOORS	1
1.3 COMMANDS	2
1.4 ADDTIME	3
1.5 BAUD	3
1.6 BROADCAST	3
1.7 BROADGETS	4
1.8 BUFFERFLUSH	4
1.9 DOWNLOAD	4
1.10 DROPCARRIER	4
1.11 GETCARRIER	4
1.12 GETCHAR	4
1.13 GETCONNECT	5
1.14 GETUSER	5
1.15 LOGENTRY	6
1.16 POLL	6
1.17 QUERY	6
1.18 RECEIVE	6
1.19 RESETMODEM	6
1.20 SCREENOUT	7
1.21 SENDFILE	7
1.22 SENDMODEM	7
1.23 SENDSTRING	7
1.24 SETINMASK	7
1.25 SETOUTMASK	8
1.26 SHUTDOWN	8
1.27 SKYPARSE	8
1.28 TRANSMIT	9
1.29 UPLOAD	9

---

---

1.30	VERSION . . . . .	9
1.31	ALTERING CALLER DATA . . . . .	9
1.32	CALLER DATA STRUCTURE . . . . .	9
1.33	MULTICHECK . . . . .	11
1.34	CREDITS . . . . .	12
1.35	HINTS & TIPS . . . . .	12

---

## Chapter 1

# Skyline 2.06 ARexx Port

### 1.1 Skyline 2.06D ARexx Port

Skyline 2.06 ARexx Port

~~~~~

DOORS

COMMANDS

CALLER DATA STRUCTURE

ALTERING CALLER DATA

MULTICHECK

HINTS & TIPS

CREDITS

### 1.2 DOORS

Writing Doors In ARexx

"Doors" are programs that can be installed into the BBS system to perform certain jobs. They are often games or utilities and can add great individuality to your system.

Several years ago a previous version of Skyline introduced a method of writing doors in ARexx, a widely used script language. This interface has been widely copied by other systems. Many of the doors written for previous versions of Skyline will work with small modifications in Skyline 2.0. These modifications are detailed below.

Version 2.0 of the Amiga operating system comes with ARexx built in. Persons running older versions of the Amiga OS will have to purchase ARexx if they desire to run doors. It is a very inexpensive program

---

and is available at your local Amiga dealer.

All programs written to run under Skyline 2.0 must begin with the following lines:

```
/* Arexx requires comment lines as the first line */
options results
arg line,second
shell VALUE 'sky_'line
```

This is required so the Arexx program can tell which line is calling it. the "shell" line then makes the Arexx program interactive with that BBS LINE. This allows the same door to be used on more than one line at once.

A potential problem exists with file writing. If two copies of the same ARexx program attempts to write to the same file at the same time, it is possible that nonsensical data will be written to the file, or that one of the file open calls will fail. It is necessary to "kludge" a workaround by checking to see if a file open was succesfull before writing; if it wasn't wait a few seconds and try again.

See:

MULTICHECK

## 1.3 COMMANDS

ADDTIME

BAUD

BROADCAST

BROADGETS

BUFFERFLUSH

DOWNLOAD

DROPCARRIER

GETCARRIER

GETCHAR

GETCONNECT

GETUSER

LOGENTRY

POLL

---

QUERY  
RECEIVE  
RESETMODEM  
SCREENOUT  
SENDFILE  
SENDMODEM  
SENDSTRING  
SETINMASK  
SETOUTMASK  
SHUTDOWN  
SKYPARSE  
TRANSMIT  
UPLOAD  
VERSION

## 1.4 ADDTIME

ADDTIME number

This adds "number" minutes to the caller's time limit for THIS call.

## 1.5 BAUD

BAUD

Changes baud when OFFLINE.

## 1.6 BROADCAST

BROADCAST string

This sends "string" to every BBS LINE that has its flag set.

See:

SETOUTMASK

---



## 1.7 BROADGETS

BROADGETS

This returns a line of text from one of the lines specified in

SETINMASK

. The FIRST BBS LINE to press RETURN is the one that makes it; all others are thrown away. The text itself is returned in RESULT, while the BBS LINE number that sent it is returned in SECOND.

## 1.8 BUFFERFLUSH

BUFFERFLUSH

This flushes the serial buffer. Any characters waiting to be read will be destroyed.

## 1.9 DOWNLOAD

DOWNLOAD

This begins a file reception FROM the caller, using his default protocol. Files are placed in the TMP: assignment.

## 1.10 DROPCARRIER

DROPCARRIER

This instantly hangs up the line on the caller and remembers caller's baud rate and other RS232 settings - for callback systems etc.

## 1.11 GETCARRIER

GETCARRIER

This returns the state of the carrier. TRUE is returned to the ARexx program if a carrier is present; FALSE is returned if not.

## 1.12 GETCHAR

GETCHAR

This gets a single character from the caller on this line as an invisible "hotkey" input. It does NOT echo the character.

---

## 1.13 GETCONNECT

GETCONNECT

Waits for the presence of a serial connection, then continues. TRUE is returned to the ARexx program if a connection is made; FALSE is returned if not. A possible use might be in a callback program.

## 1.14 GETUSER

GETUSER number

This returns information about the user based on the item specified by "number". NOTE that these numbers have CHANGED from Skyline version 1.3 to versions 2.0 and above.

- 0 Address of the Caller structure (see below)
  - 1 Name
  - 2 Password
  - 3 Security password
  - 4 City
  - 5 State
  - 6 Address
  - 7 Extra Question 1
  - 8 Extra Question 2
  - 9 Extra Question 3
  - 10 Extra Question 4
  - 11 Last day caller was on
  - 12 Sysop Comment
  - 13 Main Menu number
  - 14 Default Text area
  - 15 Caller Preference Flags (see addendum)
  - 16 Access Level
  - 17 (reserved)
  - 18 Minutes online today (if Last Day matches today)
  - 19 CLS string
  - 20 Default message area
  - 21 Default database area
  - 22 Total calls
  - 23 Credit (in lowest unit of country's currency)
  - 24 Transfer protocol number
  - 25 (reserved)
  - 26 Total uploads
  - 27 Total downloads
  - 28 Time limit
  - 29 Daily limit
  - 30 Download ratio
  - 31 Screen lines
  - 32 (reserved)
-

## 1.15 LOGENTRY

LOGENTRY string

Puts a one-line entry in your LOG file of the BBS LINE that the ARexx program is being run from.

## 1.16 POLL

POLL

Works exactly like

GETCHAR

except that it doesn't hang around

but returns immediatly. If nothing is received, LENGTH(result) will return 0. This WILL NOT catch every single character, but is usefull for detecting aborts etc.

## 1.17 QUERY

QUERY string

This is a combination command much like Basic's INPUT. It sends the string and waits for a response string. This response string is returned to the ARexx program. In the event of an error such as the caller has lost carrier, the string "###PANIC" will be returned to the ARexx program.

## 1.18 RECEIVE

RECEIVE

This gets a line of text from the user. The response string is returned to the ARexx program. In the event of an error such as the caller has lost carrier, the string "###PANIC" will be returned to the ARexx program.

## 1.19 RESETMODEM

RESETMODEM

This resets the modem to the state you have set up with the MODEM\_RESET keyword in the Config files.

---

## 1.20 SCREENOUT

SCREENOUT string

This sends a line of text to the screen only.

## 1.21 SENDFILE

SENDFILE string

This sends the textfile specified by "string". Textfiles sent using this command are processed by Skyline, so you can include embedded commands.

"string" must be a full pathname.

## 1.22 SENDMODEM

SENDMODEM string

This sends a line of text only to the serial port, without displaying it on the screen; carriage return added.

Use this instruction when talking to the modem for commands.

## 1.23 SENDSTRING

SENDSTRING string

This simply outputs a line of text, like  
                   TRANSMIT  
                   , but without a  
 trailing carriage return.

## 1.24 SETINMASK

SETINMASK number

This sets up which lines se will accept input from in the  
                   BROADGETS  
                   command. It is a 16-bit wide bitmask. To select one or more lines, ←  
                   add the  
 following numbers together as desired. (For instance, SETINMASK 5 will accept  
 input from lines 1 and 3.)

Line number    Bitmask value

---

|    |       |
|----|-------|
| 1  | 1     |
| 2  | 2     |
| 3  | 4     |
| 4  | 8     |
| 5  | 16    |
| 6  | 32    |
| 7  | 64    |
| 8  | 128   |
| 9  | 256   |
| 10 | 512   |
| 11 | 1024  |
| 12 | 2048  |
| 13 | 4096  |
| 14 | 8192  |
| 15 | 16384 |
| 16 | 32768 |

## 1.25 SETOUTMASK

SETOUTMASK number

This sets the line numbers that will receive output from the  
BROADCAST

command. "Number" is a 16-bit wide bitmask. It is set exactly the same ↔

way

SETINMASK  
is set.

## 1.26 SHUTDOWN

SHUTDOWN

This is the command you ALWAYS use to exit a Skyline ARexx program, and upon exit from a synchronous embedded ARexx command. It signals the board that it can take over once again.

## 1.27 SKYPARSE

SKYPARSE string

This is used to send a string of text which include Skyline embedded commands to be processed. This is used the same as

TRANSMIT  
, and is

followed with a carriage return.

## 1.28 TRANSMIT

TRANSMIT string

This sends a line of text to the screen and modem (IF carrier is present; if not, screen only); and follows it with a carriage return. If you wish to send a line of text that include Skyline embedded commands, use

SKYPARSE

.

## 1.29 UPLOAD

UPLOAD string

This uses the caller's default protocol to send him the file specified by "string". This must be a full pathname.

## 1.30 VERSION

VERSION

This returns a string identifying the version of Skyline being used.

## 1.31 ALTERING CALLER DATA

ALTERING THE CALLER DATA STRUCTURE

~~~~~

It is possible to permanently alter a user record from an ARexx program by using

GETUSER 0

to find the caller structure, then doing an offset to alter the correct field. It cannot be stressed strongly enough that this is NOT an exercise for novice programmers. If you use this feature, you MUST be absolutely certain you are doing it correctly before distributing your program to others.

If you do not understand the caller structure, or how to use it to find offsets, it is a sure bet that you shouldn't be tinkering with it.

## 1.32 CALLER DATA STRUCTURE

CALLER DATA STRUCTURE

~~~~~

This data structure is included for completeness. We do NOT recommend playing around with the board's data unless you are an experienced

programmer. Even then, do not alter any field named RESERVED. These structures are NOT guaranteed to remain the same in future versions.

```
struct User {
char Name[22];
char Alias[22];
char Password[12];
char Pass2[6];
char City[22];
char State[12];
char Country[12];
char Street[40];
char PostalCode[14];
char Extra1[21];
char Extra2[21];
char Extra3[21];
char Extra4[21];
char SysopComment[33];
int Baud;
char Phone1[14];
char Phone2[14];
ULONG PrefFlags;
ULONG Pref2Flags;
ULONG PrivFlags;
ULONG Priv2Flags;
ULONG CustomFlags;
ULONG Custom2Flags;
UBYTE Access;
UBYTE Editor;
UBYTE Columns;
UBYTE Lines;
UBYTE MinsToday;
UBYTE CLS[8];
UBYTE BKSPC[6];
UBYTE MainMenu;
short Tbase;
short Mbase;
short Dbase;
short TotalCalls;
long TotalMsgs;
ULONG TotalUploads;
ULONG TotalDownloads;
short Credit;
UBYTE XferProtocol;
UBYTE GrafProtocol;
short LastDay;
short TimeLimit;
short DayLimit;
ULONG FirstMail;
short DLRatio;
UBYTE RatioType;
USHORT Highs[50];
short Confs[50];
UBYTE Reserved;
UBYTE Reserved;
USHORT Reserved;
ULONG Reserved[4];
```

---

```
USHORT Scores[10];
short SigopHiDB;
short SigopLoDB;
short SigopHiMB;
short SigopLoMB;
ULONG SigopFlags;
ULONG TotalMins;
ULONG TotalRead;
USHORT Reserved;
USHORT Reserved;
UBYTE Archive;
UBYTE EOLType;
UBYTE Reserved[2];
ULONG Reserved[8];
};
```

### 1.33 MULTICHECK

MULTICHECK

The following bit of code, properly placed, will prevent any problems with mulit-writes to the same file or prevent a single ARexx program from being run more than once at the same time.

```
call MultiCheck_Start
```

```
/* your file write goes here, possibly write a game hiscore */
```

```
call MultiCheck_End
```

```
/* more of your program goes here */
```

```
/* the following should go near the bottom of your ARexx program, probably
   with your other error checking routines */
```

```
MultiCheck_Start:
parse SOURCE CMD BOOL EXT
SourceF = compress(word(EXT,1))
if index(show(clip),SourceF) then do forever
  sendstring '.'
  call delay(30)
  if ~index(show(clip),SourceF) then break
end
call setclip(SourceF,'BUSY')
return
```

```
MultiCheck_End:
call setclip(SourceF,'')
return
```

---



The above is for multi-write situations. If you want to prevent a program from being run more than once at the same time, place the "call MultiCheck\_Start" just below the "shell VALUE 'sky\_'line".

Replace the above "MultiCheck\_Start:" routine with the one below.

```
MultiCheck_Start:
parse SOURCE CMD BOOL EXT
SourceF = compress(word(EXT,1))
if index(show(clip),SourceF) then do
  transmit ' Sorry, this program is in use. Try again in a few minutes.'
  call delay(120)
  shutdown;exit
end
call setclip(SourceF,'BUSY')
return
```

## 1.34 CREDITS

- o Skyline BBS @ 2.06 is distributed by;

OmniLink Corporation 101-17 121st Richmond Hill, Queens, N.Y. 11419

MIDI MAGIC BBS 718 846-6941

- o ARexx is Copyright ©1987 by;

William S. Hawes P.O. Box 308 Maynard, MA 01754

508 568-8695

## 1.35 HINTS & TIPS

HINTS & TIPS FROM D.F. DUCK

Learn to use the ARexx "CALL" and "SIGNAL" instructions in their proper place. The easiest way I can think to explain it is to equate it with BASIC. A "CALL" in ARexx is the same as a "GOSUB" in BASIC, taking you to a subroutine and expecting a "RETURN". An ARexx "SIGNAL" is the same as a "GOTO" in BASIC, transferring control to the new routine.

Never use a "DO" loop to simulate a "DELAY". Doing so -may- cause problems with multi-line systems, and is just plain poor programming style. The rexxsupport.library has a DELAY() function built into it

You can load the rexxsupport.library from your startup-sequence or user-startup by adding the following lines;

```
loadlib libs:rexxsupport.library
;
```

```
;rexxmast -you should already have this line somewhere.
;
rxlib rexxsupport.library 0 -30 0
```

or you can add a small routine to any ARexx scripts that use this libraries functions. Many existing SkyRexx programs already have this routine in them, usually near the top. Be sure to add the following routine BEFORE the line "ADDRESS VALUE 'sky\_'line" or "SHELL VALUE 'sky\_'line".

```
if ~show('L','rexxsupport.library') then do
  if addlib('rexxsupport.library',0,-30,0) then nop
  else do
    say 'LIBS:rexxsupport.library is not available'
    exit 10
  end
end
```

When adding a PANIC ALERT check routine to your SkyRexx program, add it only after a RECEIVE or QUERY. These two instructions check for a PANIC ALERT and will return the string "###PANIC" to the ARexx program. The GETCHAR and POLL instructions do not check for a PANIC ALERT.

The built in Skyline embedded command "@R" also checks for a PANIC ALERT.

---