

サカキバラUI

API and library documentation



English version

This documentation applies to:

- XCMD.LIB revision 220
- サカキバラUI version 0.3.8

2017-2020 サイバーヨッシー64, Mr. OH

Table of Contents

	About サカキバラUI	4
	Structure of サカキバラUI packages	5
	Optional additions to the app	6
1	XCMD.LIB	7
1.1	Usage of the library	8
1.2	Library commands	9 - ???
2	サカキバラUI API	???
2.1	Simple window features	???
2.2	Window functions	???
		???
3	Other information	???
3.1	Sources & Special thanks	???

About サカキバラUI

サカキバラUI: Nintendo 3DS Edition is a mixture of a Windows-like desktop and the good old Nintendo DSi Menu for ease of use.

This UI's desktop was built on top of a Midori-OS beta build and mostly supports Midori-OS packages. A list of incompatible packages can be found in the in-game manual under MANUAL on page 17.

In this reference, only サカキバラUI-exclusive features and API calls will be listed.

For Midori-OS-related API help, use the help app found in Midori-OS's desktop.



It has been confirmed that this version of サカキバラUI will not work on the Nintendo Switch software "SmileBASIC 4", not even with Petit Converter, but CyberYoshi64 will release a Nintendo Switch version of サカキバラUI's desktop anyway.

If you want to redistribute or modify サカキバラUI's source code, please credit CyberYoshi64 and the others listed on the Sources page. (see page 11)

Redistribution and Modification is therefore allowed but don't claim any of the resources as yours, since that's not true.

Since v0.3.7, the UI comes with all its features and extras unpacked, so you don't have to do that yourself.

If you want to use サカキバラUI, do these steps (considering you're on the 🏠HOME Menu):

1. Start SmileBASIC.
2. Select "Manage Projects/Files", then "Change Active Project" and then select the latest サカキバラUI build. At the time of writing this reference, it should be `"-SKKBAUI-037R1"` or `"-SAKAKIBARAU I -"`.
3. Go back to the TOP MENU and select "Create Programs with SmileBASIC".
4. Hold , tap `LOAD`, select `".SYS_LAUNCHER"` and press , followed by the START button.

If you launch サカキバラUI in any other way, you may encounter bugs or unintended side-effects, such as the UI closing itself without warnings.

Structure of サカキバラUI packages

サカキバラUI is designed to let you write your own apps for it without the need to modify the original source code. If you want to write a app yourself, you have to make a package for it. Don't worry. It's quite simple.

```
DEBUG=0 'To test your app, change it to 1 and reload the desktop.

APP_DEF$="APP_DEF" 'replace this with the DEF name
APP_NAME$="app" 'Name your app whatever you want
BCKCOLOR=#WHITE 'The background color of your app
POSX=120 'the app's offset from the left side of the screen
POSY=80 'the app's offset from the top of the screen
SIZEX=100 'the app's width
SIZEY=25 'the app's height
TYPE=1 '0 - No header; 1 - Show a header with app name

@GLOBALS
'custom globals go here
@GLOBALS_END

@DEF 'The DEF name has to be unique to your application
COMMON DEF APP_DEF WID,STARTX,STARTY,ENDX,ENDY
  GETPOINTERXY OUT CRX,CRY
  APP_DRAG WID,CRX,CRY,STARTX,STARTY,ENDX
  APP_MINIMIZE WID,CRX,CRY

  IF ON_OPEN(WID) THEN
    'Initialization goes here
    'This code in this IF case only runs once in the
    'app's life cycle
  ENDIF

  'rest of code goes here
  'Example:
  GPUTCHR STARTX,STARTY+1,"Hello world!".#BLACK

  'This command always goes on the bottom of the
  'DEF. Otherwise, the data may not get cleared
  'properly when using system globals.
  APP_CLOSE WID,CRX,CRY

END
```

TEMPLATE.PKG excerpt with more clearer comments

Optional additions to your apps

You can let your app perform optional behaviour:

- Let your app do something when the app is closed:

```
DEF APP_LOL WID,STARTX,STARTY,ENDX,ENDY
  GETPOINTERXY OUT CRX,CRY'get cursor coordinates
  APP_MINIMIZE WID,CRX,CRY'allow app to minimize
  GPUTCHR STARTX+4,STARTY+4," 101",#BLACK

  IF VARS_NUM[WID,0]==1 THEN
    'if user tried to close the app for the first time
    GPUTCHR STARTX,STARTY+14,"Nice try!"
    INC VARS_NUM[WID,1]'increment counter
  ENDIF

  APP_CLOSE WID,CRX,CRY
END

DEF APP_LOL_CLOSE(WID)
  VARS_NUM[WID,0]=1'let app know that user tried to close
  RETURN VARS_NUM[WID,1]>60'FALSE until 1 sec has passed
                                'after user closed once
  'force the user to attempt to close your app at least twice
END
```

The RETURN decides, whether the app closes or not.

If it returns FALSE (0), the app will continue.

If it returns TRUE (1), the app will close.

This feature becomes useful, e.g. for a text editor, when the user closes it without saving.

More optional features coming soon

Chapter 1: XCMD.LIB

This library is packed with various commands by the community.

1.1 Usage of the library

If you want to use the XCMD.LIB library in your own programs, make sure you type the following piece of code into your program:

```
exec "prg3:xcmd.lib"
```

Optional lines for setup of some commands:

Enable rendering of GPRINT and GPRINT_NEW

```
SetGPRINTrender true
```

Initialize the font width for GPRINT_NEW

(you need the DAT:SYSFNT.BIN file to use this command and the GPRINT_NEW command)

```
IF !IsFontInitialized() THEN INITFNT
```

Disable error dialogs (including the INITFNT dialog when DAT:SYSFNT.BIN is not found)

```
ShowErrDlgs false
```

Return variables of the library:

- | | | |
|-----------------------|---|--|
| IsFontInitialized() | - | Returns, whether the font width data is loaded |
| GetGPRINTsize out x,y | - | Returns the width and height of the box that GPRINT and GPRINT_NEW rendered the text |
| LibErr() | - | Returns an error ID of a specific command |
| IsSKKBAUI() | - | Returns whether the library runs from
サカキバラUI (foolable) |

(R-OS V, R-OS永遠に and other programs will return 0, but サカキバラUI returns 1.)

This manual applies to Revision 210 of XCMD.LIB and may not present all commands that are available or some that were removed.

1.2 Library commands

Here are all commands listed that are currently in the library

If the examples are used in a program, make sure you initialized the library before use of the commands, otherwise a `Syntax error` occurs.

Note: Only commands intended to be used will be explained.

Graphic commands

GLINE2 *startx, starty, endx, endy, thickness, color* - Command by GMAGE_LOVEmarine

Draw a thick line

startx, starty – Start point coordinate (X: 0-511, Y: 0-511)

endx, endy – End point coordinate (X: 0-511, Y: 0-511)

thickness – Thickness in pixel units – **1**

color – Color code (See SB's GCOLOR for further help)

GBOX2 *startx, starty, endx, endy, border_radius, color* - Command by GMAGE_LOVEmarine

Draw a box with rounded corners

startx, starty – Start point coordinate (X: 0-511, Y: 0-511)

endx, endy – End point coordinate (X: 0-511, Y: 0-511)

border_radius – Border radius in pixels

color – Color code (See SB's GCOLOR for further help)

GCIRCLE2 *centerx, centery, radius, color, fill_flag* - Command by GMAGE_LOVEmarine

Draw a better circle

centerx, centery – Center point coordinate (X: 0-511, Y: 0-511)

radius – Radius of circle

color – Color code (See SB's GCOLOR for further help)

fill_flag – Fill circle (**TRUE**: Fill, **FALSE**: don't fill)

GPATCHR3 *x, y, string\$, width, height, color, space* - Command by GMAGE_LOVEmarine

Draw text with a set character width

x, y – Top left coordinate of text (X: 0-511, Y: 0-511)

string\$ – String to display as text

width, height – Draw size (**1**: 100%, **2**: 200%, ...) – Integer scale only

color – Color code (See SB's GCOLOR for further help)

space – Space between characters (**negative**: closer, **positive**: further apart, **zero**: normal)

GPRINT *fid*, *x*, *y*, *string\$*, *width*, *height*, *color* - Command by CyberYoshi64

Draw text with special formatting

fid - Initial styling (see **bitmask 1**)

x, *y* - Top left coordinate of text (X: 0-511, Y: 0-511)

string\$ - String to display as text (see **String formatting**)

width, *height* - Draw size (**1**: 100%, **2**: 200%, ...) - Integer scale only

color - Color code (See SB's GCOLOR for further help)

Note from CyberYoshi64: Do not use this command in a desktop app as it slows down the performance to a snail's pace. Recommended use is for a game and other non-desktop application.

Bitmask 1:	0 (no) - normal (no styling) 1 (b0) - bold 2 (b1) - monospacing 4 (b2) - underlining 8 (b3) - strikethrough	If <i>fid</i> is above 15 or below 0, it'll either give an error or it won't run entirely. The string "%S4F" will be interpreted as "E", basically only 1 char is read as hexadecimal as the styling ID. (Note: <i>Italic is not supported.</i>)
String formatting	%S[id] - Set styling %C[rrggbb] - Change color %W[wdt] - Change spacing between characters %L[hgt] - Change line height %% - Draw a % sign %E - Add line break %Q - Add quotation mark %g - Rainbow color %p - Randomized color %g[r] - Change wave radius	See Bitmask 1; id is hexadecimal Example: "%S5Link" → Link The values are rr,gg,bb which are hexadecimal. Transparent text is impossible. Example: "%CFF8000Hello!" → Hello! Default is set by <i>color</i> (see argument explanation) Spacing: Standard + wdt Negative numbers are allowed wdt = 0 is default Height should be 8 or higher but other values are supported. "%L-10" will add lines above each other. hgt = 8 is default Self-explanatory Self-explanatory Self-explanatory <small>Only used when not set to 0</small> Will cycle through colors when drawn constantly Apply a random color to the text coming after it Create a wave effect by setting it non-zero Remove the wave effect by setting it to 0.

Warning: When using **FORMAT\$** as *string\$*, use "%%" for one relevant instance of a % sign, e.g. **FORMAT\$("%S1%S%S0 lol", "Hello")** → **Hello lol**

This includes a simple % sign, being 4 % signs instead: **FORMAT\$("5%%%")** → 5%

ISOPAQUE(*color*) - Function by CyberYoshi64

Check whether the color code represents an opaque color (Is Alpha channel set to 255?)

color – Color code (See SB's GCOLOR for further help)

Return values

Return boolean: **TRUE** when opaque and **FALSE** if not

OPAQUE(*color*) - Function by CyberYoshi64

Return a opaque color code (Alpha channel set to 255)

color – Color code (See SB's GCOLOR for further help)

Return values

Return color code: opaque RGB value

INVERTCOL(*color*) - Function by CyberYoshi64

Invert the given color code

color – Color code (See SB's GCOLOR for further help)

Return values

Return color code: inverted opaque RGB value

GRADIENT *startx, starty, endx, endy, color1, color2, cstep* - Command by CyberYoshi64

Draw a vertical gradient with a given color step

startx, starty – Start (top-left corner) coordinate (X/Y: 0-511)

endx, endy – Start (bottom-right corner) coordinate (X/Y: 0-511)

color1 – Start color code (See SB's GCOLOR for further help)

color2 – End color code

cstep – Color step

GCOLOR3(*color1, color2, perc%*) - Function by GIMAGE_LOVEmarine

Mix 2 colors together with a specific ratio

color1 – Color code 1 (See SB's GCOLOR for further help)

color2 – Color code 2

perc% – Mix ratio in % (0: 0%; 50: 50%; 100: 100%)

RNDCOL() - Function by CyberYoshi64

Output a random color

Return values

Return color code: Randomized, opaque color code

Progress bars

PROGRESS_BAR *sx, sy, ex, ey, perc%* – Command by SilverBlue

Draw a Windows Vista-like progress bar

sx, sy – Start (top-left corner) coordinate (X/Y: 0-511)

ex, ey – End (bottom-right corner) coordinate (X/Y: 0-511)

perc% – Percentage to show (0: 0%; 50: 50%; 100: 100%) – Values above 100/below 0 allowed

PROGBAR *sx, sy, ex, ey, perc%, fgc, bgc, shc* – Command by CyberYoshi64

Draw a simple progress bar with optional animation using **PROGBAR_DISP**

sx, sy – Start (top-left corner) coordinate (X/Y: 0-511)

ex, ey – End (bottom-right corner) coordinate (X/Y: 0-511)

perc% – Percentage to show (0: 0%; 50: 50%; 100: 100%)

fgc – Custom bar color code (See SB's GCOLOR for further help) – 0 is default

bgc – Custom background color code – 0 is default

shc – Custom shine color code – 0 is default

Note: If *ex* or *ey* is smaller than *sx* or *sy* respectively, they'll be corrected to be *sx* + 7 or *sy* + 7.

PROGBAR_DISP – Command by CyberYoshi64

Update shine animation for **PROGBAR**

Note: It's advised to run this command in a loop with **VSYNC** for best use of this command duo.

Chapter 2: API

Generic structure of a サカキバラUI app:

```
DEF APP_MYAPP WID,STARTX,STARTY,ENDX,ENDY
  GETPOINTERXY OUT CRX,CRY
  APP_DRAG WID,CRX,CRY,STARTX,STARTY,ENDX
  APP_MINIMIZE WID,CRX,CRY

  'this is where your code goes

  APP_CLOSE WID,CRX,CRY
END
```

2.1 Simple window features

GETPOINTERXY OUT CRX,CRY
Get current mouse position
Return values
CRX – Mouse X offset from the left of the screen CRY – Mouse Y offset from the top of the screen

APP_DRAG WID,CRX,CRY,STARTX,STARTY,ENDX
Make the window draggable (Type 1-windows only)
WID – Current window ID CRX, CRY – Mouse position retrieved with GETPOINTERXY STARTX – Window's top-left corner position STARTY – Window's top-left corner position ENDX – Window's bottom-right corner position

APP_CLOSE WID,CRX,CRY
Render a button on the header and let it close the window
WID – Current window ID CRX, CRY – Mouse position retrieved with GETPOINTERXY
Note: When a close routine is included with the app, you may not be able to close it until the app is ready to be closed.

APP_MINIMIZE WID,CRX,CRY
Render a button on the header and let it minimize the window
WID – Current window ID CRX, CRY – Mouse position retrieved with GETPOINTERXY

APP_DESTROY WID
Close the window
WID – Current window ID
Note: When a close routine is included with the app, you may not be able to close it until the app is ready to be closed.

APP_INIT(TYPE,FNAME\$,NAME\$,BNRCLR,BCKCLR,POSX,POSY,SZX,SZY)
Render a button on the header and let it minimize the window
TYPE – Window type (0: headerless, 1: normal) FNAME\$ – App DEF to run NAME\$ – App name (Text on header) BNRCLR – Window frame color (Deprecated, leave it as 0) BCKCLR – App background color POSX – Window horizontal offset POSY – Window vertical offset SZX – Window canvas width SZY – Window canvas height
<i>Return value</i>
Child window ID -1 if app couldn't be started (too many apps open, DEF not found, etc.)

2.2 Window functions

APP_WIDTH(<i>WID</i>)
Receive the window's canvas width (= <i>ENDX</i> - <i>STARTX</i>)
<i>WID</i> – Current window ID
<i>Return value</i>
Window canvas width

APP_HEIGHT(<i>WID</i>)
Receive the window's canvas height (= <i>ENDY</i> - <i>STARTY</i>)
<i>WID</i> – Current window ID
<i>Return value</i>
Window canvas height

ON_OPEN(<i>WID</i>)
Check, whether the app was opened on the current display frame
<i>WID</i> – Current window ID
<i>Return value</i>
Boolean: TRUE when the app was opened on the current frame, otherwise FALSE

APPRUN(<i>FNAME\$</i>)
Check, whether a specific app is running
<i>FNAME\$</i> – App DEF to check
<i>Return value</i>
Boolean: TRUE when the app is running and FALSE if it isn't

MaxNumberOfNumVars()
Check, how many number variables an app can have (This number may differ between サカキバラUI builds)
<i>Return value</i>
Number of variables available in <i>VARS_NUM</i> []

MaxNumberOfStrVars()
Check, how many string variables an app can have (This number may differ between サカキバラUI builds)
<i>Return value</i>
Number of variables available in VARS_STR\$[]

MaxNumberOfAppsOpen()
Check, how many apps are able to run at once (This number may differ between サカキバラUI builds)
<i>Return value</i>
Number of apps able to run at once

Chapter 3: Other information

3.1 Sources & Special Thanks

Sources

3DS OS by andritolion:

<http://smilebasicsource.com/page?pid=516>

GIMAGE graphic commands by GIMAGE_LOVEmarine:

<http://web.archive.org/web/20171030120739/https://miiverse.nintendo.net/posts/AYMHAAADAAB2V0gbizqdxA>

R-OS Classic 3.5 by RGamesOfficial:

<https://smilebasicsource.com/page?pid=1121>

Alpha OS X by SilverBlue:

<http://wiki.hosiken.jp/petc3gou/?Toukou%2FALPHA%20OS%20X>

Midori-OS by Autz64:

<https://smilebasicsource.com/page?pid=1213>

Solid Gunner by SmileBoom Co.Ltd.:

<https://smilebasicsource.com/page?pid=1331>

<http://smilebasic.com/en/library/publickey/>

Any other piece of data, code or text belongs to either CyberYoshi64 or SmileBoom Co.Ltd.

Special Thanks

SmileBASIC Source members – bugreports, programming-wise help

andritolion – Inspiration to make サカキバラUI with 3DS OS as the oldest base for it

Autz64 – Permission to use Midori-OS v0 as current base

HTV04 – Suggestions and some decision choices

the_squat1115 – Suggestions

GamerCymreig – Translations into Welsh and Japanese

RGamesOfficial – Suggestions

SmileBoom Co.Ltd. – Opportunity to learn programming & sharing my work for the first time

2015-2017 andritolion

2017-2020 CyberYoshi64, Mr. OH

2018-2020 RGamesOfficial

2018-2020 Autz64

Thank you for reading this documentation. When my Nintendo Switch arrives, I'll make sure the documentation is ready unlike this documentation for the Nintendo 3DS version. :(