APPENDIX TO THE De Re BASIC! MANUAL ( Version 1.91, 2017-03-14) ENHANCEMENTS and CORRECTIONS as



**3.00**

APPENDIX TO THE MANUAL

(De Re BASIC! Version 1.91, 2017-03-14) ENHANCEMENTS and CORRECTIONS

A fork called OliBasic 3.00 based onRFO-BASIC!1.91 2018-08-28gt

Formatted like the manual, changes and corrections colored in dark Magenta the newest in turquoise.

What should you keep in mind, if you use source code that runs with BASIC! 1.91 perfectly.

1. The minSDK is **16**, the targetSDK is**29**
2. Look at FILE.EXISTS the second parameter is now a value or a String expression!~~3.~~
3. The automatic low memory warning is switched off. Use insteadOnLowMemory:.
4. The Global Value Backdoor, after an interrupt is trapped, is nowclosed.

If getting issues in conjunction with functions in older code created for RFO-Basic 1.91 use:

\*\*\*\*\*\*

Вилка под названием OliBasic 3.00 на основе RFO-BASIC! 1.91 2018-08-28gt

Отформатированный как руководство, изменения и исправления окрашены в темный пурпурный цвет, новейший в бирюзовый.

Что следует иметь в виду, если вы используете исходный код, который работает с BASIC! 1.91 отлично.

MinSDK - 16, targetSDK - 29

Посмотрите на FILE.EXISTS, второй параметр теперь является значением или строковым выражением! 3.

Автоматическое предупреждение о нехватке памяти отключено. Используйте взамен OnLowMemory :.

Глобальный бэкдор значений после прерывания прерывается и закрывается.

Если возникают проблемы в сочетании с функциями из старого кода, созданного для RFO-Basic 1.91, используйте:

\*\*\*\*\*\*

ON\*\*\*:

GLOBALS.ALL

….

GLOBALS.NONE

\*\*\*.RESUME

1. The RUN command has newpossibilities.
2. AndroidManifest.xml has the new permissionACCESS\_NETWORK\_STATE.
3. This version use absolute paths **and** relativepaths.
4. An internal directory is created at the firststart.
5. Add more system constants atFILE.ROOT
6. FILE.EXISTS returns readable and writeable infos,too.
7. FILE.LASTMODIFIED isnew.
8. FILE.EXISTS accepts string expressions **and** values. Seedescription.
9. BUNDLE.GET and DEBUG.DUMP.BUNDLE convert Java Objects tostrings.
10. BUNDLE.TYPEmodified
11. BigDecimalimplemented
12. ROUND() rounding option Legacy int()added
13. BYTE.COPY append switchadded
14. SOCKET.SERVER.READ.BYTE isnew
15. SOCKET.CLIENT.READ.BYTE is new
16. Scheduler commandsimplemented
17. Dialog.message, Dialog.select and Select can now be finished at a given timeinterval
18. At launching with a given program path and an Intent Extra with a key named "\_BASIC!" and a String expression “\_Editor” the program starts in the Editormode.
19. BUNDLE.OUT isnew.
20. On Runtime Error a Broadcast issend.
21. Editor improvement: Switching automatic word completions or -corrections tooff.

##### ATTENTION

###### *All result arguments of the BigDecimal commands are on the left side now.* I’m so sorry for this inconvenience!

1. EMAIL.SEND extended with CC, BCC, Attachments...
2. AUDIO.INFO is new.
3. AUDIO.RECORD.PEAK isnew.
4. AUDIO.RECORD.START has newoptions.
5. AUDIO.LOAD a http stream can also beloaded.
6. AUDIO.PLAY second parameter for outputchannel(s)
7. AUDIO.VOLUME third parameter for outer volumecontrol
8. DIR, FILE.DIR, FTP.DIR and ZIP.DIR have a time stamp option now.
9. GR.BITMAP.PUT replacedBundle.PP
10. GR.BITMAP.GET replacedBundle.GP
11. TEXT.INPUT improvements: Opportunity to switch off the text suggestion,Menu
12. USING$ needs for %d %o %x %X and %t no moreInt().
13. LIST.SORT is new. It sorts the contents of the given list, as an option also in language and country specificorder.
14. REPLACE$ has a new modifier using regularexpressions.
15. LIST.SPLIT is new. It splits a list into two new lists item byitem.
16. LIST.JOIN is new. It joins two lists into a new list item byitem.
17. LIST.MATCH is new. Returns a match found for a search argument in alist.
18. Delimiter and number-of-colums enhancements for SQL.NEW, SQL.INSERT and SQL.UPDATE. It takes a little more effort because legacy reasons, but it is needed if you want to import different tablesautomatically.
19. GR.OPEN improvement: Camera view in background, sorry only Android 6+today
20. GR.CAMERA.FOCUS is new in graphics mode in conjunction withGR.OPEN.
21. GR.CAMERA.FLASH is new in graphics mode in conjunction withGR.OPEN.
22. GR.CAMERA.ZOOM is new in graphics mode in conjunction withGR.OPEN.
23. GR.CAMERA.GETPARAMisnewingraphicsmode,sorryonlyAndroid5+today
24. GR.CAMERA.SETPARAMisnewingraphicsmode,sorryonlyAndroid6+today
25. GR.CAMERA.DIRECTSHOOT is new in graphics mode in conjunction withGR.OPEN.
26. Other GR.CAMERA. \*\*\* SHOOT commands enhanced with morearguments
27. The SELECT list view can be controlled via the layout parameter bundlenow.
28. DIR and FILE.DIR also have access to assetsnow.
29. CONSOLE.ORIENTATION is new. Works likeGR.ORIENTATION
30. CONSOLE.LAYOUT isnew.
31. CONSOLE.DEFAULT isnew.
32. CONSOLE.LINE.TOUCHED and SELECT also return a double touch tapnow. To save execution time the following GR commands are a little optimized.
33. GR.SCALE supports as a new option a translation in X and Ydirection.
34. GR.SCALE.TOUCH is new and the opposite ofGR.SCALE.

*Multi-touch with more than two touches are supported. New commands needed.*

1. GR.ARRAY.TOUCH is new and returns pairs of touch coordinates as anarray.
2. GR.LIST.TOUCH is new and returns pairs of touch coordinates aslist(s).
3. GR.LAST.TOUCH is new and returns the last touch index andcoordinates.
4. BYTE.OPEN allow by an internal cache access to converted assets filesnow.
5. BYTE.COPY break switch added, making an abort by an interruptpossible.
6. SHELL is new. Sends a shell command to the system and wait for aresult.
7. SENSORS.LIST is fixed and features new optional sensor infos (Android5+).
8. KB.SEND.KEYEVENT is new. Sends key events to theenvironment.
9. Some USB or Bluetooth keyboard function keys are supported in the Editornow.
10. STT.LISTEN has a new bundle argument to control moreoptions.
11. BACKGROUND checks if the display screen is off and if the device is lockednow.
12. FTP.OPEN new opportunity to use FTPS,also.
13. GLOBALS.FNIMP is new. It imports a variable from the main code into afunction.

#### ARRAY.BINARY.SEARCH is new. Veryfast!

1. LIST.BINARY.SEARCH is new. Veryfast!
2. GR.CAMERA.TAKEVIDEO isnew.
3. More Editor enhancements, sub menu, and options for code pre handling with Basicprograms
4. BYTE.WRITE.BUFFER improved for morespeed.
5. BYTE.READ.BUFFER, BYTE.WRITE.BUFFER, GRABFILE, GRABURL haveor have more options to select a character setnow.
6. INKEY$ returns the UTF-8 characteralso.
7. BIGD.FROMDOUBLE isnew.
8. BIGD.TODOUBLE isnew.
9. BIGD.FROMBASE like BIN, HEX, OCT, but an Integer part greater than 15 digits ispossible.
10. BIGD.TOBASE like BIN$, HEX$, OCT$, as an opposite toBIGD.FROMBASE. *The special Floating Point numbers NaN (Not a Number) and Infinity get better support now.*

#### IS\_NAN is new and detects NaN (Not a Number) Floating Pointnumbers

1. IS\_INFINITE is new and detects Infinity Floating Pointnumbers
2. WITHIN() is new. Logical function to search in geometryfigures.
3. REVERSE$() is new. Also known as MIRROR$() in other Basicdialects.
4. GR.CLS has a new option to delete all used bitmapsnow.
5. **If it is possible, bitmaps will be overwritten now. That have some consequences. See Bitmap Commands /Overview**
6. CONSOLE and SELECT are able to print text and images with HTML tagsnow.
7. New selection opportunities for CONSOLE andSELECT.
8. CLAMP() isnew.
9. LIST.JOIN added \_min and\_max
10. GR\_COLLISION New option for setting a distance collisionborder.
11. GR.BEHIND, GR.INFRONT, GR.TOBACK and GR.TOFRONT arenew.
12. Improvements in speed and power consumption associated with consoleoutput
13. BT.UTF\_8.READ and BT.UTF\_8.WRITE are new. Transfer full UTF-8 char.set.
14. UDP.READ and UDP.WRITE are new and support the UDPprotocol.
15. HTTP.REQUEST is new and enhances with more request types andoptions.
16. GR.COLOR supports also Porter-Duff maskingnow.
17. GR.PAINT.RESET resets Porter-Duffalso.
18. PROGRAM.INFO now extended with memoryinformation.
19. SENSORS.READ supports sensors which returns more than 3parameters.
20. In the Editor’s SubMenu item Previous is new. You can choose one of the ten different last loaded Basicfiles.
21. APP.SAR supportscreateChooser().
22. NOTIFY isenhanced.
23. NOTIFIY.CANCEL isnew.
24. CONSOLE.TITLE is enhanced with subtitle, icon and a hiding option **Note that the console title must be set manually now. Thus, no title "BASIC! ... Program Output" is displayed atruntime.**

#### Changes in the Editor’s Action Bar for more program namespace.

1. GR.BITMAP.DRAWINTO.END now has an option to suppress a runtimeerror
2. APP.SETTINGS opens the applicationsettings
3. DEVICE.OS is new and needs **no** Phonepermissions.
4. Permission description andcommands
5. FILE.ABSOLUTE is new and returns the absolute filepath.
6. IS\_GR is new and returns the graphic modestatus
7. SELECT is enhanced with subtitle, icon and a hidingoption
8. Drawables are now supported. (**Animated** Drawables, Android9+)
9. Broadcast messages are supported, but forgotten to inserthere.
10. GR.SCREEN isextended
11. ONGRSCREEN and GR.ONGERSCREEN.RESUME arenew.
12. APP.SAR supports getBooleanExtra()now.
13. JSONsupport
14. XML support in conjunction withJSON
15. BUNDLE supports Booleans, Drawables and JSONnow.
16. SPC$ is new. It returns a number of spaces as astring.
17. Android 9 cut out (notches) support in graphicmode.
18. GR.OPEN enhancement, a translucent or hidden navigationbar.
19. GR.SET.ACCELERATION isnew.
20. ARRAY.RND is new, which creates arrays with randomnumbers.
21. ARRAY.TO.DIMS is new, which copies and re-dimensions existingarrays.
22. SQL.SET\_LOCALE is new, set the current locale for a specified SQLdatabase
23. SQL.PING is new, returns information about a SQLdatabase.
24. REDIM is new, Re-Dimensions an existingarray
25. Array copy by B[ ] = A[ ] isnew
26. Functions are able to return arraysnow.
27. GR.HIDE and GR.SHOW support multiple argumentsnow.
28. GR.BITMAP.CLR is new and fills a bitmap complete withtransparency
29. PROGRAM.INFO returns also true if a program is started by a launchernow.
30. ONGRTOUCHMOVE:, ONGRTOUCHUP: and their Resume counterparts arenew
31. GR.CLIPOUT is new and compensates for the new limitations ofGr.clip.
32. DEVICE.USB is new and returns the parameters of plugged in USBdevices.
33. CONSOLE.LINE.TOUCHED returns swipesnow.
34. AlotofdecorstuffforStatus,ActionandNavigationBarslikeoff,on,colors,menus…
35. ONMENUITEM:, MENUITEM.RESUME and MENUITEM.GET.DATALINK arenew.
36. ONHTMLRETURN: and HTML.ONHTMLRETURN.RESUME arenew.
37. FILE.ABSOLUTE converts Document Paths into File Paths if possiblenow.
38. ADOC command group is new and takes care about AndroidDocuments.
39. IS\_HTML is new and returns the HTML modestatus
40. HTML.TO\_PDF is new and returns the content of a HTML view into a PDFdocument.
41. HTML.PAPERFORMATS is new and returns a bundle with possible paperformats
42. FILE.DIR extended with recursiveresults
43. GR.BITMAP.FILTER is new and provides bitmapfilters
44. Between ON\*: and \*.resume Format inserts alsospaces.
45. GR.SET.CAP is new and set the linecaps.
46. GR.PAINT.SET is new and provides Shaders as anexample
47. LIST.MAP.2D is new and maps translations, rotations andmultiplications
48. LIST.MAP.3D is also new and maps 3D operations in a 3Dspace
49. Editors’s Load and Delete file lists can be sorted by Date in reverse ordernow.
50. GR.BITMAP.SIZE allows direct file access and SVG filesalso.
51. GR.BITMAP.LOAD is enhanced with SVG, cropping and a backgroundcolor.
52. FILE.MD5 is new and returns the MD5 hash of afile.
53. HEX$ is enhanced and accepts a color definition string as inputalso.
54. ARRAY.MATH is new and increases the speed significantly.
55. LIST.SORT.BY is new and returns an index list, a sorted list by a givenlist.
56. GR.BITMAP.GET.PIXARR is new and returns the bitmap content asarrays.
57. GR.BITMAP.SET.PIXARR is new and returns a bitmap specified by arrays.
58. GR.BITMAP.GET.HISTOGRAM is new and returns color channelhistograms.
59. ARRAY.MAT.TRANSPOSE is new and transposes a two-dimensionalarray.

\*\*\*\*\*\*

Команда RUN имеет новые возможности.

AndroidManifest.xml имеет новое разрешение ACCESS\_NETWORK\_STATE.

Эта версия использует абсолютные пути и относительные пути.

На первом этапе создается внутренний каталог.

Добавьте больше системных констант atFILE.ROOT

FILE.EXISTS также возвращает читаемую и записываемую информацию.

FILE.LASTMODIFIED isnew.

FILE.EXISTS принимает строковые выражения и значения. Смотри описание.

BUNDLE.GET и DEBUG.DUMP.BUNDLE преобразуют строки объектов Java.

BUNDLE.TYPEmodified

BigDecimalimplemented

Добавлена ​​опция округления ROUND () Legacy int ()

BYTE.COPY

SOCKET.SERVER.READ.BYTE isnew

SOCKET.CLIENT.READ.BYTE новый

Планировщик команд реализован

Dialog.message, Dialog.select и Select теперь могут быть завершены через заданный интервал времени

При запуске с заданным путем к программе и Intent Extra с ключом с именем "\_BASIC!" и строковое выражение «\_Editor», которое программа запускает в режиме редактора.

BUNDLE.OUT является новым.

При ошибке во время выполнения отправляется широковещательная рассылка.

Улучшение редактора: Переключение автоматического завершения слов или -коррекций tooff.

ВНИМАНИЕ

Все результирующие аргументы команд BigDecimal теперь находятся слева. Мне очень жаль за это неудобство!

EMAIL.SEND расширен с CC, BCC, приложениями ...

AUDIO.INFO новый.

АУДИО.РЕКОРД.ПИК новый.

AUDIO.RECORD.START имеет новые возможности.

AUDIO.LOAD http поток также может быть загружен.

AUDIO.PLAY второй параметр для выходного канала (ов)

AUDIO.VOLUME третий параметр для внешнего управления громкостью

DIR, FILE.DIR, FTP.DIR и ZIP.DIR теперь имеют функцию отметки времени.

GR.BITMAP.PUT replaceBundle.PP

GR.BITMAP.GET replaceBundle.GP

Улучшения в TEXT.INPUT: Возможность отключить текстовое предложение, Меню

ИСПОЛЬЗОВАНИЕ $ необходимо для% d% o% x% X и% t не более ().

LIST.SORT новый. Он сортирует содержимое данного списка, в качестве опции, также в зависимости от языка и страны.

REPLACE $ имеет новый модификатор, использующий регулярные выражения.

LIST.SPLIT новый. Он разбивает список на два новых элемента списка.

LIST.JOIN новый. Он объединяет два списка в новый элемент списка.

LIST.MATCH новый. Возвращает найденное совпадение для аргумента поиска в alist.

Улучшения разделителя и числа столбцов для SQL.NEW, SQL.INSERT и SQL.UPDATE. Это требует немного больше усилий из-за устаревших причин, но это необходимо, если вы хотите импортировать разные таблицы автоматически.

Улучшение GR.OPEN: обзор камеры в фоновом режиме, извините, только Android 6+ сегодня

GR.CAMERA.FOCUS является новым в графическом режиме в сочетании с GR.OPEN.

GR.CAMERA.FLASH является новым в графическом режиме в сочетании с GR.OPEN.

GR.CAMERA.ZOOM является новым в графическом режиме в сочетании с GR.OPEN.

GR.CAMERA.GETPARAMisnewingraphicsmode, sorryonlyAndroid5 + сегодня

GR.CAMERA.SETPARAMisnewingraphicsmode, sorryonlyAndroid6 + сегодня

GR.CAMERA.DIRECTSHOOT является новым в графическом режиме в сочетании с GR.OPEN.

Другое GR.CAMERA. \*\*\* Команды SHOOT улучшены дополнительными аргументами

Представлением списка SELECT можно управлять с помощью параметра компоновки bundlenow.

DIR и FILE.DIR также имеют доступ к активам сейчас.

КОНСОЛЬ.ОРИЕНТАЦИЯ новая. Работает как GR.ORIENTATION

CONSOLE.LAYOUT isnew.

КОНСОЛЬ. ПО УМОЛЧАНИЮ

CONSOLE.LINE.TOUCHED и SELECT также возвращают двойное касание. Для экономии времени выполнения следующие GR-команды немного оптимизированы.

GR.SCALE в качестве новой опции поддерживает перевод в X и Ydirection.

GR.SCALE.TOUCH является новым и противоположным GR.SCALE.

Мультитач с более чем двумя касаниями поддерживаются. Нужны новые команды.

GR.ARRAY.TOUCH является новым и возвращает пары координат касания в виде анаррея.

GR.LIST.TOUCH является новым и возвращает пары списков координат касания.

GR.LAST.TOUCH является новым и возвращает индекс последнего касания и координаты.

BYTE.OPEN разрешает посредством внутреннего кэша доступ к преобразованным файлам файлов сейчас.

Добавлен выключатель BYTE.COPY, что делает прерывание возможным по прерыванию.

SHELL новый. Посылает команду оболочки в систему и ждет результата.

Исправлен SENSORS.LIST с новой опциональной сенсорной информацией (Android5 +).

KB.SEND.KEYEVENT новый. Отправляет ключевые события в окружающую среду.

Некоторые функциональные клавиши клавиатуры USB или Bluetooth поддерживаются в редакторе сейчас.

STT.LISTEN имеет новый аргумент связки для управления дополнительными параметрами.

BACKGROUND проверяет, выключен ли экран дисплея и не заблокировано ли устройство сейчас.

FTP.OPEN также появилась новая возможность использовать FTPS.

GLOBALS.FNIMP является новым. Он импортирует переменную из основного кода в функцию.

ARRAY.BINARY.SEARCH является новым. Очень быстро!

LIST.BINARY.SEARCH новый. Очень быстро!

GR.CAMERA.TAKEVIDEO является новым.

Дополнительные улучшения редактора, подменю и опции для предварительной обработки кода с помощью Basicprograms

BYTE.WRITE.BUFFER улучшен для большей скорости.

BYTE.READ.BUFFER, BYTE.WRITE.BUFFER, GRABFILE, GRABURL имеют больше опций для выбора набора символов теперь.

INKEY $ возвращает символ UTF-8.

BIGD.FROMDOUBLE является новым.

BIGD.TODOUBLE является новым.

BIGD.FROMBASE, такой как BIN, HEX, OCT, но возможна целочисленная часть больше 15 цифр.

BIGD.TOBASE, как BIN $, HEX $, OCT $, как противоположность BIGD.FROMBASE. Специальные числа с плавающей точкой NaN (не число) и Infinity теперь получают лучшую поддержку.

IS\_NAN является новым и обнаруживает номера с плавающей точкой NaN (не число)

IS\_INFINITE является новым и обнаруживает бесконечные числа с плавающей точкой

WITHIN () является новым. Логическая функция для поиска в геометрии.

REVERSE $ () является новым. Также известный как MIRROR $ () в других основных диалектах.

В GR.CLS появилась новая опция для удаления всех используемых битмапнов.

Если это возможно, растровые изображения будут перезаписаны. Это имеет некоторые последствия. Смотрите битовые команды / обзор

CONSOLE и SELECT могут печатать текст и изображения с помощью HTML-тегов.

Новые возможности выбора для CONSOLE и SELECT.

CLAMP () является новым.

LIST.JOIN добавил \_min and\_max

GR\_COLLISION Новая опция для установки границы столкновения на расстоянии.

GR.BEHIND, GR.INFRONT, GR.TOBACK и GR.TOFRONT являются новыми.

Улучшения в скорости и энергопотреблении, связанные с выходом консоли

BT.UTF\_8.READ и BT.UTF\_8.WRITE являются новыми. Передача полного UTF-8 char.set.

UDP.READ и UDP.WRITE являются новыми и поддерживают протокол UDP.

HTTP.REQUEST является новым и расширяется с помощью большего количества типов запросов и опций.

GR.COLOR поддерживает также маскировку Портера-Даффа.

GR.PAINT.RESET сбрасывает Porter-Duffalso.

PROGRAM.INFO теперь расширен памятью информации.

SENSORS.READ поддерживает датчики, которые возвращают более 3 параметров.

В подменю редактора пункт Предыдущий новый. Вы можете выбрать один из десяти различных последних загруженных базовых файлов.

APP.SAR поддерживает createChooser ().

УВЕДОМЛЕНИЕ.

NOTIFIY.CANCEL isnew.

В CONSOLE.TITLE добавлены субтитры, значок и скрытие. Обратите внимание, что заголовок консоли теперь должен быть установлен вручную. Таким образом, заголовок «BASIC! ... Program Output» не отображается в рабочее время.

Изменения в панели действий редактора для большего пространства имен программ.

GR.BITMAP.DRAWINTO.END теперь имеет возможность подавить ошибку времени выполнения

APP.SETTINGS открывает настройки приложения

DEVICE.OS является новым и не требует никаких телефонных разрешений.

Описание разрешения и команды

FILE.ABSOLUTE является новым и возвращает абсолютный путь к файлу.

IS\_GR является новым и возвращает графический режим

В SELECT добавлены субтитры, иконки и скрытая опция

Drawables теперь поддерживаются. (Анимированные Рисунки, Android9 +)

Широковещательные сообщения поддерживаются, но забыты вставить сюда.

GR.SCREEN продлен

ONGRSCREEN и GR.ONGERSCREEN.RESUME являются новыми.

APP.SAR теперь поддерживает getBooleanExtra ().

JSONsupport

Поддержка XML в сочетании с JSON

BUNDLE поддерживает Booleans, Drawables и JSONnow.

SPC $ новый. Возвращает несколько пробелов в качестве вяжущих.

Поддержка Android 9 вырезана (метки) в графическом режиме.

GR.OPEN улучшение, полупрозрачная или скрытая панель навигации.

GR.SET.ACCELERATION является новым.

ARRAY.RND является новым, который создает массивы со случайными числами.

ARRAY.TO.DIMS является новым, который копирует и изменяет размеры существующих массивов.

SQL.SET\_LOCALE новый, установить текущую локаль для указанной базы данных SQL

SQL.PING является новым, возвращает информацию о базе данных SQL.

REDIM новый, Re-Dimensions существующий массив

Копирование массива B [] = A [] является новым

Функции могут возвращать массивы.

GR.HIDE и GR.SHOW поддерживают несколько аргументов теперь.

GR.BITMAP.CLR является новым и заполняет растровое изображение в полной прозрачности

PROGRAM.INFO также возвращает true, если программа запущена программой запуска.

ONGRTOUCHMOVE :, ONGRTOUCHUP: и их резюме резюме являются новыми

GR.CLIPOUT является новым и компенсирует новые ограничения Gr.clip.

DEVICE.USB является новым и возвращает параметры подключенных USB-устройств.

CONSOLE.LINE.TOUCHED возвращает swipesnow.

AlotofdecorstuffforStatus, ActionandNavigationBarslikeoff, на, цвет, меню ...

ONMENUITEM :, MENUITEM.RESUME и MENUITEM.GET.DATALINK новые.

ONHTMLRETURN: и HTML.ONHTMLRETURN.RESUME являются новыми.

FILE.ABSOLUTE преобразует пути документов в пути файлов, если это возможно сейчас.

Группа команд ADOC является новой и заботится о AndroidDocuments.

IS\_HTML является новым и возвращает HTML-статус

HTML.TO\_PDF является новым и возвращает содержимое представления HTML в PDF-документ.

HTML.PAPERFORMATS является новым и возвращает пакет с возможными форматами бумаги

FILE.DIR расширен рекурсивными результатами

GR.BITMAP.FILTER является новым и предоставляет битовые фильтры

Между ON \*: и \* .resume Формат вставляет также пробелы.

GR.SET.CAP является новым и установить linecaps.

GR.PAINT.SET является новым и предоставляет шейдеры в качестве примера

LIST.MAP.2D является новым и отображает переводы, вращения и умножения

LIST.MAP.3D также является новым и отображает 3D-операции в 3D-пространстве

Списки загрузки и удаления редакторов могут быть отсортированы по дате в обратном порядке.

GR.BITMAP.SIZE позволяет прямой доступ к файлам и SVG-файлам.

GR.BITMAP.LOAD улучшен за счет SVG, обрезки и фона.

FILE.MD5 является новым и возвращает хеш MD5 файла.

HEX $ улучшен и принимает строку определения цвета как inputalso.

ARRAY.MATH является новым и значительно увеличивает скорость.

LIST.SORT.BY является новым и возвращает индексный список, отсортированный список по заданному списку.

GR.BITMAP.GET.PIXARR является новым и возвращает asarrays содержимого растрового изображения.

GR.BITMAP.SET.PIXARR является новым и возвращает растровое изображение, указанное массивами.

GR.BITMAP.GET.HISTOGRAM является новым и возвращает гистограммы цветовых каналов.

ARRAY.MAT.TRANSPOSE является новым и транспонирует двумерный массив.

\*\*\*\*\*\*

Some fixes

Thanks to Alberto, Andrey, Bob, Brochi, Carlos, Chris, Emile, Guillermo, Humpty, Janusz, Jean-Manuel, Jürgen, knoWare, Mog, Michael, Nicolas, Roy, Spike for testing and support. Special thanks to Paul for creating RFO-Basic and Marc for the great enhancements and criticisms with substance.

Special thanks also to:

* + Emile for textcorrections
  + Humpty for the source of Globals.fnimp and other great stuff I have takenover
  + Janusz for the hints of the needs of visually impairedusers
  + Stephen C (Stackloverflow) about "What About Reading Unknown Numberof Bytes?"
  + Brochi for the bit shifting code and other wonderful hints and codeexamples
  + Guillermo for testing display cutouts
  + KnoWare for the SQLknowhow
  + Spike for creating a knowledge basewiki
  + Jürgen for a lot of excellent examples and testing newstuff
  + Paul LeBeau for best support about SVGcropping.

\*\*\*\*\*\*

Некоторые исправления

Спасибо Альберто, Андрею, Бобу, Броши, Карлосу, Крису, Эмилю, Гильермо, Шалтайу, Янушу, Жан-Мануэлю, Юргену, НОУ, Могу, Майклу, Николасу, Рою, Спайку за тестирование и поддержку. Отдельное спасибо Полу за создание RFO-Basic и Марку за отличные улучшения и критику по существу.

Особая благодарность также:

Эмиль для исправления текста

Шалтай для источника Globals.fnimp и других замечательных вещей, которые я приобрел

Януш за подсказки о потребностях слабовидящих

Стивен С (Stackloverflow) о «Как насчет чтения неизвестного числа байтов?»

Броши для сдвигающего бит кода и других замечательных подсказок и примеров кода

Гильермо для тестирования вырезов дисплея

KnoWare для SQLknowhow

Спайк для создания базы знаний вики

Юргену за множество отличных примеров и тестирование новинок

Пол Лебо за лучшую поддержку в SVGcropping

\*\*\*\*\*\*

.

Happy coding Gregor Tiemeyer

## Gregor’s Stuff

**The Closed Process Circle**

Send Result and Finish

Broadcast

StartActivity or

an.Android.App

com.rfo.basicOli

Broadcast

StartActivityForResult

##### App.installed <flag\_nexp>,

**<package\_sexp>{{{,<versionName\_sexp>},<versionCode\_nexp>},<pmRaw\_sexp>}** If the android package is installed, flag\_nexp returns 1, else 0. Insert the package id like "com.rfo.basic”" or "all.sub.My.APP". VersionName and versionCode are results from the APK manifest. Package manager returns a raw string list in pmRaw\_sexp.

\*\*\*\*\*\*

если пакет аndroid установлен, flag\_nexp возвращает 1, иначе 0. Вставьте идентификатор пакета, например «com.rfo.basic» или «all.sub.My.APP». VersionName и versionCode являются результатами манифеста APK. Диспетчер пакетов возвращает необработанный список строк в pmRaw\_sexp.

\*\*\*\*\*\*

##### App.load <package\_sexp>

can be found in the Sample Program file, f00a\_download\_manual.bas.)

Package$ = "file://" + "/android\_asset/" + "Bookworm\_en.apk" (Protected, copy

the Apk first to the data path.)

App.Load Package$

\*\*\*\*\*\*

можно найти в файле примера программы, f00a\_download\_manual.bas.) Пакет $ = "file: //" + "/

android\_asset /" + "Bookworm\_en.apk" (Защищено, сначала скопируйте Apk в путь к данным.)

App.Load Package $

\*\*\*\*\*\*

**APP.SAR <pointer\_nexp>**

**S**tart **A**nd **R**eceive an **App,** Intent or Broadcast with a Java like Interface. The background references are cut off!

Eg. FileBrowser, BarCodeScanner, GoogleMaps, ... are supported.

Note: The Blackmoon FileBrowser is suggested, because it does not need the clipboard for multiple files.

There is also APP.START with an other interface and less features (and still an internal design issue), but in some cases an option.

The APP.BROADCAST command does not work work properly yet, use BROADCAST (faster) or APP.SAR for complex wishes.

\*\*\*\*\*\*

Запустите и получите приложение, намерение или трансляцию с Java-подобным интерфейсом. Фоновые ссылки обрезаны!

Например. FileBrowser, BarCodeScanner, GoogleMaps, ... поддерживаются.

Примечание. Рекомендуется Blackmoon FileBrowser, поскольку он не требует буфера обмена для нескольких файлов.

Существует также APP.START с другим интерфейсом и меньшим количеством функций (и все же внутренняя проблема дизайна), но в некоторых случаях опция.

Команда APP.BROADCAST пока не работает должным образом, используйте BROADCAST (быстрее) или APP.SAR для сложных пожеланий.

\*\*\*\*\*\*

Example:

LIST.CREATE **S**, **commandList**Pointer LIST.ADD commandListPointer~

"new **I**ntent(**I**ntent.ACTION\_GET\_CONTENT);" ~ "setPackage(**\"**com.blackmoonit.android.FileBrowser**\"**);" ~ "setType(**\"**\*/\***\"**);" ~ %From Ex.: intent.setType( "\*/\*"); "addCategory(Intent.CATEGORY\_DEFAULT);" ~

!%From Ex.: **i**ntent.addCategory(**I**ntent.CATEG... "addFlags(**I**ntent.FLAG\_ACTIVITY\_EXCLUDE\_FROM\_RECENTS);" ~

! → **);" ~ ←** is important "EOCL"

LIST.CREATE **S**, **resultList**Pointer LIST.ADD resultListPointer~ "**theFilePath$**=getData().getPath();" ~ "EORL"

BUNDLE.CREATE appVarPointer

BUNDLE.PL appVarPointer,**"\_CommandList"**, commandListPointer BUNDLE.PL appVarPointer,**"\_ResultList"**, resultListPointer BUNDLE.PUT appVarPointer,"**theFilePath$**", "No file selected!"

APP.SAR appVarPointer

ATTENTION: In Java variable, class and function names are case sensitive. In most Java examples → **i**ntent. ← is a variable of the class Intent and have to be deleted.

Instead → **I**ntent. ← is the class and strongly not to delete.

**\_CommandList** is a required key expression.

**\_ResultList** is a required key expression if you want results.

**\_Broadcast** is a required key expression if you want to send a Broadcast.

Basic! Variables used in the CommandList (only the bracketed variables with exclamation like “!variable$!”) and ResultList have to be put in the transfer bundle, too.

Ignore and delete → this. ←.

Note: The maximum size for the intent bundle is limited to about 1 MB when transferring data with intents.

TIP: If the expected results is not returned then PRINT the expressions first to debug yourcode.

\*\*\*\*\*\*

ВНИМАНИЕ: В переменной Java имена классов и функций чувствительны к регистру. В большинстве примеров Java → намерение. ← является переменной класса Intent и должна быть удалена.

Вместо этого → Намерение. ← это класс и сильно не удалять.

\_CommandList является обязательным ключевым выражением.

\_ResultList является обязательным ключевым выражением, если вам нужны результаты.

\_Broadcast является обязательным ключевым выражением, если вы хотите отправить трансляцию.

Основные! Переменные, используемые в CommandList (только заключенные в квадратные скобки переменные с восклицательным знаком, как «! Variable $!») И ResultList, также должны быть помещены в пакет передачи.

Игнорировать и удалить → это. ←.

Примечание. Максимальный размер пакета намерений ограничен примерно 1 МБ при передаче данных с намерениями.

СОВЕТ: Если ожидаемые результаты не возвращаются, тогда напечатайте выражения сначала для отладки вашего кода.

\*\*\*\*\*\*

Supported:

new Intent setAction setPackage

putExtra only Bundle, String, Long, Double and the arguments true and false addCategory

addFlags

setComponent(new ComponentName(

setData setDataAndType setType

Caution: If you want to set both the URI and MIME type, don't call setData() and setType() because they each nullify the value of the other. Always use setDataAndType() to set both URI and MIME type.

getStringExtra getIntExtra getDoubleExtra getBundleExtra getBooleanExtra getData().getPath()

getLaunchIntentForPackage() getParcelableArrayListExtra(Intent.EXTRA\_STREAM)

Out of The box createChooser

"createChooser("+ CHR$(34) + "Chooser Title, share with ..." + CHR$(34) +");" ~ This list will maybe expanded. Stay tuned!

In the underlying Java machine were two main types of application starts implemented. First type: **startActivity** used by App.Start and App.ASR without \_ResultList

In this case the activity is **not launched** as a **sub**-activity!

Second type: **startActivityForResult** used by App.ASR with \_ResultList

Important: If you apply App.ASR with \_ResultList use the suggested ReorderToFront() function below before finishing, which prevents trouble by hitting the home button at sub intent runtime. In this case Console.front and Gr.front fail.

If the called intent was written in Java or an other language as BASIC!, use the appropriate commands.

\*\*\*\*\*\*

В базовой машине Java были реализованы два основных типа запуска приложений. Первый тип: startActivity, используемый App.Start и App.ASR без \_ResultList

В этом случае действие не запускается как вспомогательное действие!

Второй тип: startActivityForResult, используемый App.ASR с \_ResultList

Важное замечание: Если вы применяете App.ASR с \_ResultList, используйте предложенную функцию ReorderToFront (), приведенную ниже, перед завершением, что предотвращает проблемы, нажимая кнопку «Домой» во время выполнения намерения. В этом случае Console.front и Gr.front терпят неудачу.

Если вызываемое намерение было написано на Java или другом языке как BASIC !, используйте соответствующие команды.

\*\*\*\*\*\*

A look at <https://developer.android.com/guide/components/intents-filters>is recommended. See also Bundle.In, Bundle.Out

Examples:

!----------------- ReorderToFront() also part of TestTest.bas -------------------

FN.DEF ReorderToFront() PROGRAM.INFO b

BUNDLE.GET b, "\_PackageName", pN$ LIST.CREATE S, commandListPointer LIST.ADD commandListPointer~

"new Intent(Intent.ACTION\_MAIN);" ~ "setPackage("+ CHR$(34) + pN$ + CHR$(34) +");" ~ "addCategory(Intent.CATEGORY\_DEFAULT);" ~

"addFlags(Intent.FLAG\_ACTIVITY\_REORDER\_TO\_FRONT);" ~ "EOCL"

BUNDLE.PL appVarPointer,"\_CommandList",commandListPointer APP.SAR appVarPointer

FN.RTN appVarPointer FN.END

!----------------- Rest of TestTest.bas -------------------

PROGRAM.INFO bPointer

BUNDLE.GET bPointer, "\_BasName", bN$ BUNDLE.GET bPointer, "\_PackageName", pN$ BUNDLE.GET bPointer, "\_MyProcessId", mpid$

CONSOLE.TITLE bN$ + " runs on the " + pN$ + " Basic Engine"

? "\_MyProcessId", mpid$ file.root fp$,"\_Programpath"

?fp$

BUNDLE.IN action$, data$, bi

?"action$ ";action$

? "data$ "; data$ p = 2000

? "Waiting " + int$(p/1000) + " seconds before killing" PAUSE p

BUNDLE.PUT resultPointer, "EXTRA\_RESULT", "EXTRA\_RESULT Strings Test" BUNDLE.PUT resultPointer, "Test", "Strings Test"

BUNDLE.PUT resultPointer, "TestNum", 4711 BUNDLE.OUT a$, d$, resultPointer

!To prevent a Home-button-hitting-issue if you need automatic itself-returning ReorderToFront()

EXIT

If you use in the next examples the variable basicEngine$ you can use: com.rfo.basicOli ,

com.rfo.basicFellow , NEW com.rfo.basic, with limitations

Or crate your own one with:

BUNDLE.IN action$, data$, bi FILE.EXISTS ok,data$

IF ok THEN RUNdata$

sel = -4000 %NEW If negative the dialog.message will close after <sel> Milliseconds DIALOG.MESSAGE "Program File not Found!", data$, sel

EXIT

If you call a second BASIC! engine start maybe with com.rfo.basicFellow from OliBasicFellow\*\*.apk and call a **different** engine maybe com.rfo.basicOli.

\*\*\*\*\*\*

Если вы называете второй бейсик! запуск двигателя может быть с com.rfo.basicFellow от OliBasicFellow \*\*. apk и вызов другого двигателя, возможно com.rfo.basicOli.

\*\*\*\*\*\*

!----------------- SubIntentWithResult.bas -------------------

FN.DEF SubIntentWithResult(basEngine$, basProgramPath$, mode, testMode) eMode$ = ""

IF mode > 0 THEN eMode$ = "\_Editor" LIST.CREATE S, commandListPointer LIST.ADD commandListPointer~

"new Intent(Intent.ACTION\_MAIN);" ~

"setData("+ CHR$(34) + basProgramPath$ + CHR$(34) +");" ~

"new ComponentName(**\""** + basEngine$ + **"\""** + ","+**"\""**+ basEngine$ + ".Basic" + **"\""**+");" ~ "addCategory(Intent.CATEGORY\_EMBED);" ~

!Starts program in Editor mode, if eMode$ = "\_Editor"!

"putExtra("+ CHR$(34) + "\_BASIC!" + CHR$(34) + ","+CHR$(34)+ eMode$ + CHR$(34)+");" ~

"addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);" ~ "EOCL"

LIST.CREATE S, resultListPointer IF testMode > 0 THEN

LIST.ADD resultListPointer~ "theDataPath$=getData().getPath();" ~

"EXTRA\_RESULT$=getStringExtra("+ CHR$(34) + "EXTRA\_RESULT"+ CHR$(34) + ");" ~

"Test$=getStringExtra("+ CHR$(34) + "Test"+ CHR$(34) + ");" ~ "TestNumN=getDoubleExtra("+ CHR$(34) + "TestNum"+ CHR$(34) + ");" ~ "EORL"

ELSE

LIST.ADD resultListPointer~ "EORL"

ENDIF

BUNDLE.PL appVarPointer,"\_CommandList",commandListPointer BUNDLE.PL appVarPointer,"\_ResultList",resultListPointer

IF testMode > 0 THEN

BUNDLE.PUT appVarPointer,"theDataPath$",basProgramPath$ BUNDLE.PUT appVarPointer,"EXTRA\_RESULT$", EXTRA\_RESULT$

BUNDLE.PUT appVarPointer,"Test$", Test$ BUNDLE.PUT appVarPointer,"TestNumN", TestNum ENDIF

APP.SAR appVarPointer FN.RTN appVarPointer FN.END

FN.DEF ReturnedItems$(appVarPointer, pr) BUNDLE.KEYS appVarPointer, keyListPointer LIST.SIZE keyListPointer, listSize

IF pr

? "listSize " , listSize-4 FOR I = 1 TO listSize

LIST.GET keyListPointer, I, item$

IF item$ = "\_CommandList" | item$ = "\_ResultList" | item$ = "%succededResults%" ~

| item$ = "DebugSI" ELSE

PRINT item$ ENDIF

NEXT i ENDIF

BUNDLE.GET appVarPointer,"%succededResults%", Results$ FN.RTN Results$

FN.END

basEngine$ ="com.rfo.basicOli" % Your favorite BASIC! engine

FILE.ROOT fp$ , "\_External" % Start point of the public part of the external file system basProgramPath$ = "file://" + fp$ + "/rfo-basic/source/TestTest.bas"

mode = 0 %Only execution mode

! mode = 1 %Editor mode

! testMode = 0 % Without results testMode = 1 % With results

appVarPointer = SubIntentWithResult(basEngine$, basProgramPath$, mode, testMode)

IF testMode

myReturnedItems$ = ReturnedItems$(appVarPointer, 0) PRINT myReturnedItems$

BUNDLE.GET appVarPointer,"EXTRA\_RESULT$",EXTRA\_RESULT$ PRINT "EXTRA\_RESULT=: "; "<"+EXTRA\_RESULT$+">"

BUNDLE.GET appVarPointer,"Test$",Test$ PRINT "Test=: "; "<"+Test$+">"

BUNDLE.GET appVarPointer,"TestNumN",TestNum PRINT "TestNum=: "; "<";TestNum;">"

ENDIF END

!----------------- IndependentLaunch.bas -------------------

FN.DEF IndependentLaunch(basEngine$, basProgramPath$, mode) eMode$ = ""

IF mode > 0 THEN eMode$ = "\_Editor" LIST.CREATE S, commandListPointer LIST.ADD commandListPointer~

"new Intent(Intent.ACTION\_MAIN);" ~

"setData("+ CHR$(34) + basProgramPath$ + CHR$(34) +");" ~

"new ComponentName(\""+ basEngine$ + "\""+ ","+"\""+ basEngine$ + ".Basic" +"\""+");" ~ "addCategory(Intent.CATEGORY\_LAUNCHER);" ~

"putExtra("+CHR$(34) + "\_BASIC!" + CHR$(34) + ","+CHR$(34)+ eMode$ + CHR$(34)+");" ~

"addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);" ~

! vv "addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);" ~

"addFlags(Intent.FLAG\_ACTIVITY\_MULTIPLE\_TASK);" ~ "addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);" ~

! ^^ "EOCL"

BUNDLE.PL appVarPointer,"\_CommandList",commandListPointer APP.SAR appVarPointer

FN.END

basEngine$ ="com.rfo.basicOli" % You favorite BASIC! engine

file.root fp$ , "\_External" % Start point of the public part of the external file system basProgramPath$ = "file://" + fp$ + "/rfo-basic/source/TestTest.bas"

! mode = 1 % Program starts in BASIC!'s Editor mode = 0

CALL IndependentLaunch(basEngine$, basProgramPath$, mode) END

##### App.settings {<package\_sexp>}

Calls the application settings of the app with the package-id <package\_sexp>. If the optional package-id is not given, application settings of the running Basic engine or the settings of the executed application created with Basic! is opened.

App.settings is asynchronous on runtime. Use Dialog.message or Dialog.select to halt the program execution.

\*\*\*\*\*\*

Вызывает настройки приложения приложения с идентификатором пакета <package\_sexp>. Если необязательный идентификатор пакета не указан, то настройки приложения работающего ядра Basic или настройки исполняемого приложения, созданные с помощью Basic! открыт.

App.settings является асинхронным во время выполнения. Используйте Dialog.message или Dialog.select, чтобы остановить выполнение программы.

\*\*\*\*\*\*

Example:

App.settings %"com.rfo.basicOli"

Dialog.message "Settings", "Shoud we check again?", sel, "Yes", "No" IF sel = 1 THENGetPernissions()

##### Browse <url\_sexp>

If <url\_sexp> starts with "http…" then the internet site specified by <url\_sexp> willbe opened anddisplayed.

If <url\_sexp> starts with "file://" + absolutePath$ + "/" + fileName$then the file will be open by a linkedapplication.

Note: You can also use the HTML commands to display (and interact with) web pages located on your device or on the web.

\*\*\*\*\*\*

Если <url\_sexp> начинается с «http…», то интернет-сайт, указанный в <url\_sexp>, будет открыт и отображен.

Если <url\_sexp> начинается с "file: //" + absolutePath $ + "/" + fileName $, тогда файл будет открыт связанным приложением.

Примечание. Команды HTML также можно использовать для отображения (и взаимодействия) веб-страниц, расположенных на вашем устройстве или в Интернете.

\*\*\*\*\*\*

See also Gr.camera.takeVideo

##### Http.post <url\_sexp>, <list\_nexp>, <result\_svar>{{{{{, <ok\_svar>},

**<use\_caches\_lexp>}, <charset\_sexp>}, <connect\_timeout\_nexp>},**

**<read\_timeout\_nexp>}**

Execute a Post command to an Internet location.

<url\_sexp> contains theurl("http(s):// ") that will accept thePost.

<list\_nexp> is a pointer to a string list which contains the Name/Value pairs needed for the Post. <result\_svar> is where the Post response will be placed.

If the server returns an error if <ok\_svar> is not "ok". An error handler has to be placed behind this command, if needed.

If you want a cache <use\_caches\_lexp> has to be > 0.

To choose a character set use <charset\_sexp>. Default is "\_UTF-8" for text and use "\_ISO-8859-1" for binary data.

If you want to specify a connect timeout use <connect\_timeout\_nexp> with a value > 0. If you want to specify a read timeout use <read\_timeout\_nexp> with a value > 0.

\*\*\*\*\*\*

Выполните команду «Отправить» в Интернет-хранилище.

<url\_sexp> содержит theurl ("http (s): //"), который будет принимать пост.

<list\_nexp> - указатель на список строк, который содержит пары Имя / Значение, необходимые для публикации. <result\_svar> - это место, где будет размещен ответ Post.

Если сервер возвращает ошибку, если <ok\_svar> не в порядке. При необходимости за этой командой должен быть размещен обработчик ошибок.

Если вы хотите, чтобы кеш <use\_caches\_lexp> должен быть> 0.

Чтобы выбрать набор символов, используйте <charset\_sexp>. По умолчанию «\_UTF-8» для текста и «\_ISO-8859-1» для двоичных данных.

Если вы хотите указать время ожидания соединения, используйте <connect\_timeout\_nexp> со значением> 0. Если вы хотите указать время ожидания чтения, используйте <read\_timeout\_nexp> со значением> 0.

\*\*\*\*\*\*

##### Http.request <request\_type\_sexp>, <url\_sexp>, <list\_nexp>,<result\_svar>{{{{{,

**<ok\_svar>}, <use\_caches\_lexp>}, <charset\_sexp>},<connect\_timeout\_nexp>},**

**<read\_timeout\_nexp>}**

Execute a request command to an Internet location.

Available request types are "\_DELETE", "\_GET", "\_PATCH", "\_POST", "\_PUT" with write and read access and "\_HEAD", "\_OPTIONS", "\_TRACE" only with read access.

<url\_sexp> contains theurl("http(s):// ") that will accept the request.

<list\_nexp> is a pointer to a string list which contains the Name/Value pairs needed for the request. If you have nothing to write, create an empty list and inset this list pointer.

<result\_svar> is where the request response will be placed.

If the server returns an error if <ok\_svar> is not "ok". An error handler has to be placed behind this command, if needed.

If you want a cache <use\_caches\_lexp> has to be > 0.

To choose a character set use <charset\_sexp>. Default is "\_UTF-8" for text and use "\_ISO-8859-1" for binary data.

If you want to specify a connect timeout use <connect\_timeout\_nexp> with a value > 0. If you want to specify a read timeout use <read\_timeout\_nexp> with a value > 0.

\*\*\*\*\*\*

Выполните команду запроса в Интернет-хранилище.

Доступные типы запросов: "\_DELETE", "\_GET", "\_PATCH", "\_POST", "\_PUT" с доступом для записи и чтения, а "\_HEAD", "\_OPTIONS", "\_TRACE" только с доступом для чтения.

<url\_sexp> содержит theurl ("http (s): //"), который примет запрос.

<list\_nexp> - указатель на список строк, который содержит пары Имя / Значение, необходимые для запроса. Если вам нечего написать, создайте пустой список и вставьте этот указатель списка.

<result\_svar> - это место, где будет размещен ответ на запрос.

Если сервер возвращает ошибку, если <ok\_svar> не в порядке. При необходимости за этой командой должен быть размещен обработчик ошибок.

Если вы хотите, чтобы кеш <use\_caches\_lexp> должен быть> 0.

Чтобы выбрать набор символов, используйте <charset\_sexp>. По умолчанию «\_UTF-8» для текста и «\_ISO-8859-1» для двоичных данных.

Если вы хотите указать время ожидания соединения, используйте <connect\_timeout\_nexp> со значением> 0. Если вы хотите указать время ожидания чтения, используйте <read\_timeout\_nexp> со значением> 0.

**Home**

The **HOME** command does exactly what tapping the HOME key would do. The Home Screen is displayed while the BASIC! program continues to run in the background.

AtAndroid 4.2.1 with 512 MB RAM the BASIC! program does not continue under a lot of circumstances. Beginning with Android 5 tasks are running better in background.

The opposite are the commands Console.front, Gr.front and the suggested ReorderToFront() function. (See examples under APP.SAR) The last one includes all cases inclusive HTML WebView.

\*\*\*\*\*\*

Команда HOME делает именно то, что делает нажатие клавиши HOME. Главный экран отображается пока BASIC! Программа продолжает работать в фоновом режиме.

На Android 4.2.1 с 512 МБ ОЗУ ОСНОВНОЙ! Программа не продолжается при многих обстоятельствах. Начиная с Android 5 задачи работают лучше в фоновом режиме.

Противоположными являются команды Console.front, Gr.front и предлагаемая функция ReorderToFront (). (См. Примеры в разделе APP.SAR) Последний включает все случаи, включая HTML WebView.

\*\*\*\*\*\*

##### Gr.front flag

Determines whether the graphics screen or the Output Console will be the front-most screen. If flag = 0, the Output Console will be the front-most screen and seen by the user. If flag <> 0, the graphics screen will be the front-most screen and seen by the user.

~~One use for this command is to display the Output Console to the user while in graphicsmode. Use~~ **~~Gr.front 0~~** ~~to show text output and~~ **~~Gr.front 1~~** ~~to switch back to the graphicsscreen.~~

~~Note: When the Output Console is in front of the graphics screen, you can still draw (butnot render) onto the graphics screen. The~~ **~~Gr.front 1~~** ~~must be executed before any~~**~~Gr.render~~**~~.~~

Print commands will continue to print to the Output Console even while the graphic screen is in front.

If you get trouble bringing the application from background to front (Android < 5) use a user defined function like ReorderToFront(). (See examples underAPP.SAR) You can use this example for Graphic and Console mode, it returns to the actual used mode automatically.

ButthereisnoguaranteeinAndroid<5 thattheapplicationsreturnsautomaticallyontop.

You should know, that the Console mode is waiting until the Graphic mode finished. If you use Console.front or Gr.front 0 in Graphic mode **BASIC! will hang**. Use instead maybe Dialog.message, Dialog.select or Select. See also their enhanced commanddescriptions.

\*\*\*\*\*\*

Определяет, будет ли графический экран или Консоль вывода самым передним экраном. Если флаг = 0, Консоль вывода будет самым передним экраном и будет видна пользователю. Если флаг <> 0, графический экран будет самым передним экраном и будет виден пользователю.

Одним из вариантов использования этой команды является отображение Консоли вывода для пользователя в графическом режиме. Используйте Gr.front 0, чтобы показать вывод текста, и Gr.front 1, чтобы переключиться обратно на графический экран.

Примечание. Когда консоль вывода находится перед графическим экраном, вы все равно можете рисовать (но не визуализировать) на графическом экране. Gr.front 1 должен быть выполнен до anyGr.render.

Команды печати будут продолжать печатать в Консоль вывода, даже если графический экран находится впереди.

Если у вас возникли проблемы с переносом приложения из фона (Android <5), используйте пользовательскую функцию, например ReorderToFront (). (См. Примеры под APP.SAR). Этот пример можно использовать для графического и консольного режимов, он автоматически возвращается в фактически используемый режим.

Но это не гарантия, что Android <5 автоматически возвращает приложения.

Вы должны знать, что консольный режим ожидает завершения графического режима. Если вы используете Console.front или Gr.front 0 в графическом режиме BASIC! будет висеть Вместо этого можно использовать Dialog.message, Dialog.select или Select. Смотрите также их расширенные описания команд

\*\*\*\*\*\*

# Console I/O

### Output Console

BASIC! has four types of output screens: The Output Console, the Graphics Screen, ~~and~~the HTML Screen and the Select Screen (last one as a dialog). This section deals with the Output Console. See the section on Graphics for information about the Graphics Screen. See the section on HTML for information about the HTML screen.

Information is printed to screen using the Print command. BASIC! Run-time error messages are also displayed on this screen.

There is no random access to locations on this screen. Lines are printed one line after the other.

Although no line numbers are displayed, lines are numbered sequentially as they are printed, starting with 1. These line numbers refer to lines of text output, not to locations onthe screen.

##### Print {<exp> {,|;}} ...

**? {<exp> {,|;}} ...**

Evaluates the expression(s) <exp> and prints the result(s) to the Output Console. You can use a question mark (**?**) in place of the command keyword **Print**.

If the comma (**,**) separator follows an expression then a comma and a space will be printed after the value of the expression.

If the semicolon (**;**) separator is used then nothing will separate the values of the expressions.

If the semicolon is at the end of the line, the output will not be printed until a **Print**

command without a semicolon at the end is executed.

**Print** with no parameters prints a newline.

\*\*\*\*\*\*

BASIC! имеет четыре типа экранов вывода: Консоль вывода, Графический экран, Экран HTML и Экран выбора (последний из которых является диалоговым окном). Этот раздел посвящен выходной консоли. См. Раздел «Графика» для получения информации об графическом экране. См. Раздел HTML для получения информации об экране HTML.

Информация выводится на экран с помощью команды «Печать». BASIC! Сообщения об ошибках во время выполнения также отображаются на этом экране.

На этом экране нет произвольного доступа к местоположениям. Линии печатаются одна строка за другой.

Хотя номера строк не отображаются, строки нумеруются последовательно по мере их печати, начиная с 1. Эти номера строк относятся к строкам вывода текста, а не к местам на экране.

Распечатать {<exp> {, |;}} ...

? {<exp> {, |;}} ...

Оценивает выражение (я) <exp> и выводит результат (ы) в консоль вывода. Вы можете использовать знак вопроса (?) Вместо ключевого слова команды Print.

Если разделитель запятой (,) следует за выражением, то после значения выражения будет напечатана запятая и пробел.

Если используется разделитель точки с запятой (;), то ничто не будет разделять значения выражений.

Если точка с запятой находится в конце строки, вывод не будет напечатан до

Команда без точки с запятой в конце выполняется.

Печать без параметров печатает новую строку.

\*\*\*\*\*\*

Examples:

PRINT"New","Message" % Prints: New,Message

PRINT"New";"Message" % Prints: NewMessage

PRINT "New" +"Message" % Prints: NewMessage

? 100-1;"Luftballons" % Prints: 99.0Luftballons

? FORMAT$("##", 99);"Luftballons" % Prints: 99Luftballons

PRINT"A";"B";"C"; % Prints:nothing

PRINT"D";"E";"F" % Prints:ABCDEF

##### Print with User-Defined Functions

Note: Some commands, such as **Print**, can operate on either strings or numbers. Sometimes it has to try both ways before it knows what to do. First it will try to evaluate an expression as a number. If that fails, it will try to evaluate the same expression as a string. If this happens, and the expression includes a function, the function will be called twice. If the function has side-effects, such as printing to the console, writing to a file, or changing a global parameter, the side-effect action will also happen twice.

Eventually this problem should be fixed in BASIC!, but until then you should be careful not to call a function, especially a user-defined function, as part of a **Print** command. Instead, assign the return value of the function to a variable, and then **Print** the variable. An assignment statement always knows what type of expression to evaluate, so it never evaluates twice.

\*\*\*\*\*\*

Примечание. Некоторые команды, такие как «Печать», могут работать как со строками, так и с числами. Иногда он должен попробовать оба пути, прежде чем он знает, что делать. Сначала он попытается оценить выражение как число. Если это не удастся, он попытается вычислить то же выражение, что и строка. Если это произойдет, и выражение включает в себя функцию, функция будет вызвана дважды. Если у функции есть побочные эффекты, такие как печать на консоль, запись в файл или изменение глобального параметра, действие побочного эффекта также будет происходить дважды.

В конечном итоге эта проблема должна быть исправлена ​​в BASIC !, но до тех пор вы должны быть осторожны, чтобы не вызывать функцию, особенно пользовательскую функцию, как часть команды Print. Вместо этого присвойте возвращаемое значение функции переменной, а затем напечатайте переменную. Оператор присваивания всегда знает, какой тип выражения оценивать, поэтому он никогда не вычисляет дважды.

\*\*\*\*\*\*

! Do this:

y = MyFunction(x) Print y

! NOT this:

Print MyFunction(x)

##### Print with HTML tags for text formatting and image including

With the Console.Layout (also Select with a layout bundle) command you are able to format your string output in a HTML manner. Thus **colored** letters or included bitmaps may be as bitmap fonts are possible.

\*\*\*\*\*\*

С помощью команды Console.Layout (также выберите Select with layout layout) вы можете форматировать строковый вывод в формате HTML. Таким образом, цветные буквы или включенные растровые изображения могут быть как возможными растровыми шрифтами.

\*\*\*\*\*\*

##### The Output Console and power consumption

The console output is often updated internally. Depending on the device, this process requires more or less power, especially in connection with bitmaps.

Remember to use the SELECT command if possible.

The PAUSE command is also possible, but events are ignored during this period. Behind the screens:

RFO-Basic1.91 creates a new console list view in its main loop each time, if the console buffer is not empty!

Starting with OliBasicXXII in its main loop, a new console list view is created only if the console buffer is not empty or the hash of the console buffer is changed!

That is, the content of the buffer has changed in this case.

If you use the SELECT command, only a new SELECT list view is created and the main loop is paused until the selection is done with return.

The difference is hot, slightly above body temperature and cool with a HTC device. Therefore, it is good practice to use the SELECT command instead of the console with all BASIC! versions, if it is possible.

\*\*\*\*\*\*

Выходная консоль и энергопотребление

Вывод на консоль часто обновляется внутри. В зависимости от устройства этот процесс требует большей или меньшей мощности, особенно в связи с растровыми изображениями.

Не забудьте использовать команду SELECT, если это возможно.

Команда PAUSE также возможна, но события игнорируются в течение этого периода. За экранами:

RFO-Basic1.91 каждый раз создает новый вид списка консоли в своем основном цикле, если буфер консоли не пуст!

Начиная с OliBasicXXII в его главном цикле, новое представление списка консоли создается, только если буфер консоли не пуст или хэш консольного буфера изменен!

То есть содержимое буфера изменилось в этом случае.

Если вы используете команду SELECT, создается только новое представление списка SELECT, и основной цикл приостанавливается до тех пор, пока выбор не будет завершен с возвратом.

Разница горячая, чуть выше температуры тела и прохладная с устройством HTC. Следовательно, хорошей практикой является использование команды SELECT вместо консоли со всеми бейсиками! версии, если это возможно.

##### Console.front

Brings the Output Console to the front where the user can see it.

If BASIC! is running in the background with no screen visible, this command brings it tothe foreground. If you have a different application running in the foreground, it will be pushed to thebackground.

If BASIC! is running in the foreground, but the Graphics or HTML screen is in the foreground, this command brings the Console to the foreground. BASIC! remains in Graphics or HTML mode.

If you get trouble bringing the application from background to front (Android < 5) use a user defined function ReorderToFront() function. (See examples under APP.SAR)

See also Gr.front.

\*\*\*\*\*\*

Выводит консоль вывода вперед, где ее сможет увидеть пользователь.

Если бейсик! работает в фоновом режиме без видимого экрана, эта команда выводит его на передний план. Если на переднем плане запущено другое приложение, оно будет перенесено на задний план.

Если бейсик! выполняется на переднем плане, но на переднем плане находится экран «Графика» или «HTML», эта команда выводит консоль на передний план. BASIC! остается в режиме графики или HTML.

Если у вас возникли проблемы с переносом приложения из фона (Android <5), используйте пользовательскую функцию ReorderToFront (). (См. Примеры в разделе APP.SAR)

Смотрите также Gr.front.

\*\*\*\*\*\*

##### Console.orientation <nexp>

The value of the <nexp> sets the orientation of screen as follows:

-1 = Orientation depends upon the sensors. 0 = Orientation is forced to Landscape.

1 = Orientation is forced to Portrait.

2 = Orientation is forced to Reverse Landscape. 3 = Orientation is forced to Reverse Portrait.

You can monitor changes in orientation by reading the screen width and height using the

**Screen** command.

\*\*\*\*\*\*

See also: GR.open, GR.orientation, Select

Вы можете отслеживать изменения в ориентации, считывая ширину и высоту экрана, используя

Экранная команда.

Смотрите также: GR.open, GR.orientation, Select

\*\*\*\*\*\*

##### Console.layout <layout\_bundle\_nexp>

The layout bundle <layout\_bundle\_nexp> controls the console output layout:

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** |  |
| **\_TextSize** | numeric |  |
| **\_TextColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Note, \_TextFont or \_TextStyle is needed also! |
| \_**TextBackgroundColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Has to be "0,0,0,0" if you want a background color, wallpaper or bitmap  Note, \_TextFont or \_TextStyle is needed also! |
| **\_TextFont** | \_Default |  |
| \_Serif |  |
| \_Sans\_Serif |  |
| \_Monospace |  |
| **\_TextStyle** | \_Normal |  |
| \_Bold |  |
| \_Bold\_Italic |  |
| \_Italic |  |

|  |  |  |
| --- | --- | --- |
| **\_TextHtml** | 0 or 1 (numeric) | Returns displayable styled text from the provided HTML string. But not all tags are supported. Any <img> tags in the HTML will display an image. Absolute ("file://") and relative paths are allowed.  The image size has to be scaled before, because h= and w= are ignored. See\_HtmlBitmapScale.    Uses parts of TagSoup library to handle real HTML, including allof the brokenness found in thewild.  <a href="...">  <b>  <big>  <blockquote>  <br>  <cite>  <dfn>  <div align="...">? Use instead chr$(1564) [Arabic Letter] at line begin for align=’ right’  <em>  <font size="..." color="..." face="...">  <h1>, <h2>, <h3>, <h4>, <h5>,  <h6>  <i>  <img src="...">  <p>  <small>  <strike>? < A.7  <strong>  <sub>  <sup>  <tt>?  <u>  **Replace Space with &#160,**  **& with &amp,**  **< with &lt,**  **> with &gt,**  **" with &quot if necessary.** |
| **\_HtmlTextSelectable** | 0 or 1 (numeric) | Does only work in conjunction with  \_TextHtml, but the item selection works only with a **long** click. |

|  |  |  |
| --- | --- | --- |
| **\_HtmlBitmapScale** | -1, 0, > 0 (numeric) | Does only work in conjunction with  \_TextHtml. Scales the included bitmaps in the following ways:  -1 no scaling,  0 (default) only scaling proportional to the screen resulution  > 0 proportional to the font size  If it is 1 the bitmap height is the same as the font size. |
| **\_DividerColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) |  |
| **\_DividerFilename** | bitmap file path |  |
| **\_DividerHeight** | numeric |  |
| **\_BackgroundWallpaper** | 0 or 1 (numeric) | ~~Min. Jelly Bean 4.1 (API 16)~~ |
| **\_BackgroundColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) |  |
| **\_BackgroundFilename** | bitmap file path | ~~Min. Jelly Bean 4.1 (API 16)~~ |
| **\_SetSelection** | numeric | Sets a pre selected item. The item will not be selected but it will still be positioned appropriately. If the specified selection position is less  than 1, then the item at position 1 will be selected. |
| **\_StackFromBottom** | 0 or 1 (numeric) | If 1 pin the view's content to the bottom edge, 0 to pin the view's  content to the top edge |

For simpleness you can use the same layout bundle for Console.layout and the Select command. But the \_Orientation key will be ignored. In this case also use Console.orientation.

If you want to change something in the layout bundle, it is sufficient to make the changes only in the bundle. Note that every change affects the whole thing. Place a layoutchange as soon as possible in the Basic! Code. Maybe you have to place a little (PAUSE)break.

See also: Console.orientation, Select

\*\*\*\*\*\*

Для простоты вы можете использовать один и тот же пакет макетов для Console.layout и команду Select. Но ключ \_Orientation будет игнорироваться. В этом случае также используйте Console.orientation.

Если вы хотите что-то изменить в пакете макетов, достаточно внести изменения только в пакет. Обратите внимание, что каждое изменение влияет на все это. Разместите макет как можно скорее в Basic! Код. Может быть, вам нужно сделать небольшой (PAUSE) перерыв.

Смотрите также: Console.orientation, Выбрать

\*\*\*\*\*\*

##### Console.default

Resets the console layout to the default settings.

##### Console.line.touched <line\_nvar> {, <touch\_nvar>}

After an **OnConsoleTouch** interrupt indicates the user has touched the console, this command returns information about the touch.

The number of the line that the user touched is returned in the <line\_nvar>.

If the optional <touch\_nvar> is present, the type of user touch a short tap (0), a long press

(1) or a double tap (2) is returned in the <press\_nvar>.The delay for detecting the double tap conforms to the default Android system settings.

If <touch\_nvar> returns numbers greater than 9 it provides swipe directions as follows

10 → right, 11 ↓ down, 12 ← left, 13 ↑ up. If you want to select the line also, keep in mind, that line has to be high enough.

\*\*\*\*\*\*

После того, как прерывание OnConsoleTouch указывает, что пользователь коснулся консоли, эта команда возвращает информацию о касании.

Номер строки, к которой прикоснулся пользователь, возвращается в <line\_nvar>.

Если присутствует необязательный <touch\_nvar>, тип пользователя касается короткого нажатия (0), длительного нажатия

(1) или двойное нажатие (2) возвращается в <press\_nvar>. Задержка для обнаружения двойного нажатия соответствует настройкам системы Android по умолчанию.

Если <touch\_nvar> возвращает числа больше 9, он предоставляет направления пролистывания следующим образом

10 → вправо, 11 ↓ вниз, 12 ← влево, 13 ↑ вверх. Если вы также хотите выбрать линию, имейте в виду, что эта линия должна быть достаточно высокой.

\*\*\*\*\*\*

##### Console.title {{ <title\_sexp>}, <options\_bundle\_nexp>}

Changes the title of the console window. If the <title\_sexp> parameter is omitted, the title is changed to the default title, "BASIC! Program Output".

##### Starting with OliBasic 3.00 the console title is empty by default.

##### \*\*\*\*\*\*

##### Изменяет заголовок окна консоли. Если параметр <title\_sexp> пропущен, заголовок заменяется на заголовок по умолчанию «ОСНОВНОЙ! Вывод программы».

##### Начиная с OliBasic 3.00 заголовок консоли по умолчанию пуст.

##### \*\*\*\*\*\*

The optional options bundle <options\_bundle\_nexp> controls the Action and Navigation bar layouts:

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** | **Description** |
| **\_Subtitle** | String | Set the action bar's subtitle. |
| **\_TitleShow** | 0 or 1 (numeric) | If 1 (default)  Show the Action bar if it is not currently showing. It will resize application content to fit the new space available.  If 0  Hide the Action bar if it is currently showing. It will resize application content to fit the new space available. |
| **\_TitleIcon** | Icon file path | Add a large icon to the notification content view.<http://romannurik.github.io/>  AndroidAssetStudio/index.html |
| **\_TitleHomeEnabled** | 0 or 1 (numeric) | Set whether to include the application home accordance in the action bar. Home is presented as an activityicon.  Have to be 1 if you want to show the icon.  Have to be 0 if you want to hide the icon.  The default setting is API dependent. |
| **\_TitleBackground** | Background file path |  |

|  |  |  |
| --- | --- | --- |
| **\_TitleHtml** | 0 or 1 (numeric) | Returns displayable styled text from the provided HTML string. But not all tags are supported.    Uses parts of TagSoup library to handle real HTML, including all of the brokenness found in the wild.  <b>  <big>  <font size="..." color="..." face="...">  <h1>, <h2>, <h3>, <h4>, <h5>,  <h6>  <i>  <small>  <strike>? < A.7  <strong>  <sub>  <sup>  <tt>?  <u>  **Replace Space with &#160,**  **& with &amp,**  **< with &lt,**  **> with &gt, " with &quot if necessary.**  Usable for Title andSubtitle.  Keep in mind that theActionbar height will not beexpanded. |
| **\_StatusbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |
| **\_StatusbarLight** | 0 or 1 (numeric) | If 0 (default)  The Status bar background is dark. In this case the **bar content** will be **light**.  If 1  The Status bar background is light. In this case the **bar content** will be **dark**.  Min. Lollipop 5.0 (API 21) |

|  |  |  |
| --- | --- | --- |
| **\_ShowNavigationbar** | 0, 1 or 2 (numeric) | If 1 (default)  The Navigation bar will be displayed.  If 2  The Navigation bar will be transparent displayed.  Min. Lollipop 5.0 (API 21)  If 0  The Navigation bar will be hidden to the background.  Min. Nougat 7.0 (API 24)  Will be switched to option 2 or 1 if the current API level is lower. |
| **\_NavigationbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |
| **\_NavigationbarLight** | 0 or 1 (numeric) | If 0 (default)  The Navigation bar background is dark.  In this case the **bar content** will be  **light**. If 1  The Navigation bar background is light.  In this case the **bar content** will be  **dark**.  Min. Lollipop 5.0 (API 21) |
| **\_Menu** | Menu Bundle Pointer | Creates additional menu entries. Behind "Stop" and "Editor" if it is running in the development mode. A successful selection will be returned as a human readable  JSON string. |

See also: Console.layout, Select

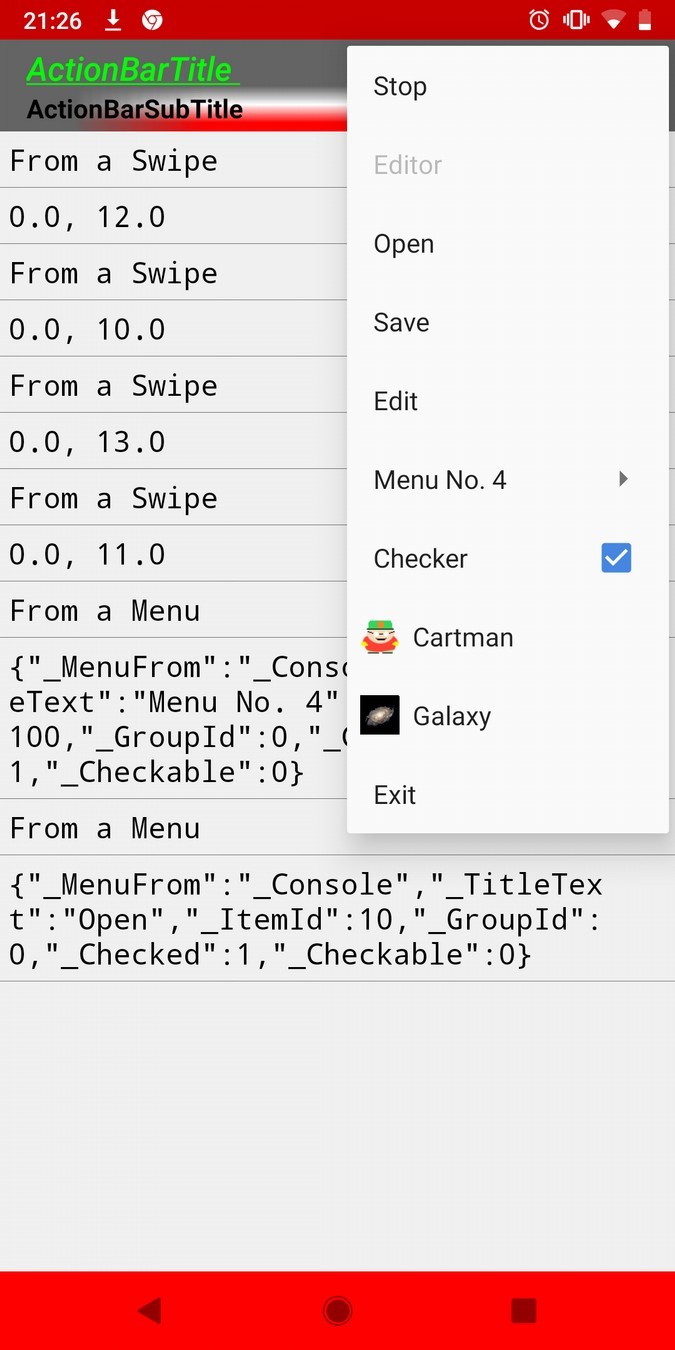
Example

bundle.put c, "\_Subtitle", " My Sub Title" bundle.put c, "\_TitleHomeEnabled", 1 bundle.put c, "\_TitleBackground", "bgColor.png" bundle.put c, "\_TitleShow", 1

bundle.put c, "\_TitleIcon", "cartman.png"

Console.title "My Console Title", c DO

PAUSE 50 % Saves power consumption! UNTIL 0

! Example: How to Create a Menu FN.DEF setMenuLayoutBundle()

! "Stop" default in development mode

! "Editor" default in development mode BUNDLE.PUT item10, "\_Title", "Open" BUNDLE.PB menuLayout, "10", item10 BUNDLE.PUT item30, "\_Title", "Save" BUNDLE.PB menuLayout, "30", item30 BUNDLE.PUT item40, "\_Title", "Edit" BUNDLE.PB menuLayout, "40", item40 BUNDLE.PUT item100, "\_Title", "Menu No. 4" BUNDLE.PUT item100, "\_SubMenuStart", 1 BUNDLE.PB menuLayout, "100", item100

BUNDLE.PUT item110, "\_Title", "Sub Menu No. 1" BUNDLE.PUT item110, "\_GroupId", 1 BUNDLE.PUT item110, "\_Checked", 1 BUNDLE.PB menuLayout, "110", item110 BUNDLE.PUT item120, "\_Title", "Sub Menu No. 2"

! Next Line is important! Has to be placed behind the last SubMenu item. BUNDLE.PUT item120, "\_GroupCheckable", 1

! Next Line is important! Has to be placed behind the last SubMenu item. BUNDLE.PUT item120, "\_GroupExclusive", 1

BUNDLE.PUT item120, "\_GroupId", 1 BUNDLE.PB menuLayout, "120", item120 BUNDLE.PUT item200, "\_Title", "Checker"

! Next Line is important!

! Has to be placed at the next entry behind the last SubMenu item. BUNDLE.PUT item200, "\_AfterSubMenuEnd", 1

BUNDLE.PUT item200, "\_Checkable", 1 BUNDLE.PUT item200, "\_Checked", 1 BUNDLE.PB menuLayout, "200", item200 BUNDLE.PUT item210, "\_Title", "Cartman" BUNDLE.PUT item210, "\_Icon", "cartman.png" BUNDLE.PB menuLayout, "210", item210 BUNDLE.PUT item220, "\_Title", "Galaxy" BUNDLE.PUT item220, "\_Icon", "galaxy.png"

! ~~BUNDLE.PUT item220, "\_Enable", 0 %~~??? Does still not work. BUNDLE.PB menuLayout, "220", item220

BUNDLE.PUT item230, "\_Title", "Exit" BUNDLE.PB menuLayout, "230", item230 FN.RTN menuLayout

FN.END

BUNDLE.PB layoutBundle , "\_Menu", setMenuLayoutBundle()

**Keep in mind, that only one sub menu level is possible on Android.**

OnMenuItem:

PRINT "From a Menu" MENUITEM.GET.DATALINK data$

PRINT data$

! TONE 600, 200

IF IS\_IN("Exit", data$) & IS\_IN("\_Console", data$) THEN END

IF IS\_IN("Exit", data$) & IS\_IN("\_HTML", data$) THEN HTML.CLOSE : myCloser = 1

IF IS\_IN("Exit", data$) & IS\_IN("\_Graphic", data$) THEN EXIT % GR.CLOSE : myCloser = 1 MENUITEM.RESUME

##### Console.save <filename\_sexp>

The current contents of the Console is saved to the text file specified by the filenamestring expression. If the graphics or HTML mode is enabled an error accrues and asks forclosing thismode.

\*\*\*\*\*\*

Текущее содержимое консоли сохраняется в текстовом файле, указанном в выражении filenamestring. Если включен графический режим или режим HTML, возникает ошибка и запрашивается закрытие этого режима.

\*\*\*\*\*\*

See also IS\_GR, IS\_HTML

### ~~Screen rotation, size[], realsize[], density~~

Returns inform………..

**It is a description fault!**

The command is not implemented and returns a Syntax Error. Use instead Screen.size and Screen.rotation.

\*\*\*\*\*\*

Возвращает сообщить ……… ..

Это ошибка описания!

Команда не реализована и возвращает синтаксическую ошибку. Используйте вместо Screen.size и Screen.rotation.

\*\*\*\*\*\*

### JSON

JSON ("**J**ava**S**cript **O**bject **N**otation") defines a lightweight data format in which information such as objects, arrays, and other variables can be stored in readable form.In OliBasic, JSON strings can be transformed as Bundles corresponding key strings or converted to XMLstrings.

JSON is often used to easily exchange information between client and server and isa handy alternative toXML.

See the documentation at https:/[/www.w3schools.com/js/js\_json\_intro.asp.](http://www.w3schools.com/js/js_json_intro.asp)

A JSON object begins with a "{" and ends with a "}".

\*\*\*\*\*\*

JSON («JavaScript Object Notation») определяет облегченный формат данных, в котором информация, такая как объекты, массивы и другие переменные, может храниться в читаемой форме. В OliBasic строки JSON могут быть преобразованы как Связки соответствующих ключевых строк или преобразованы в XML-строки.

JSON часто используется для простого обмена информацией между клиентом и сервером и является удобной альтернативой XML.

См. Документацию по адресу https://www.w3schools.com/js/js\_json\_intro.asp.

Объект JSON начинается с "{" и заканчивается "}".

\*\*\*\*\*\*

Workaround for known bundle put issue:

GRABFILE j$, "project.json"

j$ = REPLACE$(j$, "[]", "[" + "\"" + "\"" +"]") % Fill **empty Arrays** with a "" member

! Now fill **empty Strings** with n.a.m. = "not a member" j$ = REPLACE$(j$, "\"" + "\"", "\"" + "n.a.m."+ "\"")

##### Is\_Json (<json\_sexp>)

Checks if <json\_sexp> is an acceptable or better well done JSON string. Returns 1 if true otherwise 0 iffalse.

See also JsonToXml$(),Bundle.GJ

\*\*\*\*\*\*

Проверяет, является ли <json\_sexp> приемлемой или лучше сделанной строкой JSON. Возвращает 1, если true, иначе 0 iffalse.

Смотрите также JsonToXml $ (), Bundle.GJ

\*\*\*\*\*\*

**XmlToJson$ (<xml\_sexp>{, <space\_nexp>})** Retruns a JSON string converted from a XML string. Needs an acceptable or better well done JSON string.

If the optional <spaces\_nexp> returns a number > -1, a structural printout is delivered. The number of spaces defines the tabulator distance.

If <spaces\_nexp> is -1 or not given, a compact printout is returned.

There may be some post-processing needed, especially if JSON objects beginning with a "{" and ending with a "}" are involved.

If an error occurred, an empty string will be returned. See also JsonToXml$(), Bundle.PJ

\*\*\*\*\*\*

Строка JSON, преобразованная из строки XML. Требуется приемлемая или лучше сделанная строка JSON.

Если необязательный <space\_nexp> возвращает число> -1, структурная распечатка доставляется. Количество пробелов определяет расстояние табуляции.

Если <space\_nexp> равно -1 или не задано, возвращается компактная распечатка.

Может потребоваться некоторая постобработка, особенно если задействованы объекты JSON, начинающиеся с "{" и заканчивающиеся "}".

Если произошла ошибка, будет возвращена пустая строка. Смотрите также JsonToXml $ (), Bundle.PJ

\*\*\*\*\*\*

##### HTML Example

<!DOCTYPE html>

<html>

<body>

<h2>Store and retrieve data from local storage.</h2>

<p id="JsonOut"></p>

<p id="lastNameOut"></p>

<p id="postalCodeOut"></p>

<script>

var myObj, myJSON, text, obj, obj2;

//Storing data:

myObj = { "firstName":"John",

"lastName": "Smith", "age": 25,

"address": {

"streetAddress": "21 2nd Street", "city": "New York",

"state": "NY", "postalCode": 10021

},

"phoneNumbers": [

{

"type": "home",

"number": "212 555-1234"

},

{

"type": "fax",

"number": "646 555-4567"

}

]

};

myJSON = JSON.stringify(myObj); localStorage.setItem("testJSON", myJSON); document.getElementById("JsonOut").innerHTML = myJSON;

// Retrieving data:

text = localStorage.getItem("testJSON");

// First level

obj = JSON.parse(text); document.getElementById("lastNameOut").innerHTML = obj.lastName;

// Second level obj2 =obj.address;

document.getElementById("postalCodeOut").innerHTML = obj2.postalCode;

</script>

</body>

</html>

##### OliBasic Example

! Stringified JSON object

myJSON$ = "{\"firstName\": \"John\",\"lastName\": \"Smith\",\"age\": 25," +~ "\"address\": {\"streetAddress\": \"21 2nd Street\",\"city\": \"New York\"," +~ "\"state\": \"NY\",\"postalCode\": 10021},\"phoneNumbers\": " +~

"[{\"type\": \"home\",\"number\": \"212 555-1234\"}," +~

"{\"type\": \"fax\",\"number\": \"646 555-4567\" }] }" PRINT "JsonOut:", myJSON$

PRINT "Is\_json", IS\_JSON( myJSON$)

! First level

!Workaround for known bundle put issue:

! Fill **empty Arrays** with a "" member

myJSON$ = REPLACE$(myJSON$, "[]", "[" + "\"" + "\"" +"]")

! Now fill **empty Strings** with n.a.m. = "not a member"

myJSON$ = REPLACE$(myJSON$, "\"" + "\"", "\"" + "n.a.m."+ "\"") BUNDLE.PJ testJSON, myJSON$

BUNDLE.GET testJSON, "lastName", lastName$ PRINT "LastNameOut:", lastName$ BUNDLE.GET testJSON, "address",obj\_address$

! Second level

PRINT "Is\_json", IS\_JSON( obj\_address$) BUNDLE.PJ obj\_address, obj\_address$

BUNDLE.GET obj\_address, "postalCode", postalCode$ PRINT "PostalCodeOut:", postalCode$

### XML

XML (E**x**tensible **M**arkup **L**anguage) is a text-based format for the exchange of structured information. These can be documents, configurations, books, invoices, and more. OliBasic supports XML not in all aspects and is limited by possibilities of the JSON-XML and the XML-JSON parser. So only well formed expressions are parsed. For a well formed output add

<?xml version="1.0" encoding="utf-8"?> at the begin of the string.

Unfortunately is a additional DOCTYPE declaration (DTD) entry for a valid XML document needed. See the documentation at https://[www.w3schools.com/xml/default.asp.](http://www.w3schools.com/xml/default.asp)

\*\*\*\*\*\*

XML (Extensible Markup Language) - это текстовый формат для обмена структурированной информацией. Это могут быть документы, конфигурации, книги, счета и многое другое. OliBasic поддерживает XML не во всех аспектах и ​​ограничен возможностями JSON-XML и синтаксического анализатора XML-JSON. Таким образом, анализируются только правильно сформированные выражения. Для хорошо сформированного вывода добавить

<? xml version = "1.0" encoding = "utf-8"?> в начале строки.

К сожалению, требуется дополнительная запись объявления DOCTYPE (DTD) для правильного XML-документа. См. Документацию по адресу https://www.w3schools.com/xml/default.asp.

\*\*\*\*\*\*

##### Is\_Xml (<xml\_sexp>)

Checks if <Xml\_sexp> is an acceptable or better well formed XML string. Returns 1 if true otherwise 0 if false.

\*\*\*\*\*\*

Проверяет, является ли <Xml\_sexp> приемлемой или лучше правильно сформированной строкой XML. Возвращает 1, если истина, иначе 0, если ложь.

\*\*\*\*\*\*

See alsoXmlToJson$()

##### JsonToXml$ (<json\_sexp>{{,<spaces\_nexp>}, <shrink\_lexp>})

Retruns a XML string converted from a JSON string. Needs an acceptable or better well formed XML string.

If the optional <spaces\_nexp> returns a number > -1, a structural printout is delivered. The number of spaces defines the tabulator distance.

If <spaces\_nexp> is -1 or not given, a compact printout is returned. If an error occurred, an empty string will be returned.

\*\*\*\*\*\*

Возвращает строку XML, преобразованную из строки JSON. Требуется приемлемая или лучше правильно сформированная строка XML.

Если необязательный <space\_nexp> возвращает число> -1, структурная распечатка доставляется. Количество пробелов определяет расстояние табуляции.

Если <space\_nexp> равно -1 или не задано, возвращается компактная распечатка. Если произошла ошибка, будет возвращена пустая строка.

\*\*\*\*\*\*

Note, that

<example name:"Marc" age:31 male:true /> returns

<example><name>Marc</name><age>31</age><male>true</male></example> if the optional <shrink\_lexp> is switched to off by 0.

See alsoXmlToJson$()

### Bundles

A Bundle is a group of values collected together into a single object. A bundle object may contain any number of string, numeric values and also there arrays, lists, stacks, bundles, booleans, drawables and bitmaps. There is no fixed limit on the size or number of bundles. You are limited only by the memory of your device.

The values are set and accessed by keys. A key is string that identifies the value. For example, a bundle might contain a person’s first name and last name. The keys for accessing those name strings could be "first\_name" and "last\_name". An age numeric value could also be placed in the Bundle using an "age" key.

A new, empty bundle is created by using the **Bundle.create** command. The command returns a pointer to the empty bundle. Because the bundle is represented by a pointer, bundles can be placed in lists and arrays. Bundles can also be contained in other bundles. This means that the combination of lists and bundles can be used to create arbitrarily complex data structures.

After a bundle is created, keys and values can be added to the bundle using the **Bundle.put** command. Those values can be retrieved using the keys in the **Bundle.get** command. There are other bundle commands to facilitate the use of bundles.

\*\*\*\*\*\*

Пакет - это группа значений, собранных в один объект. Объект пакета может содержать любое количество строк, числовых значений, а также массивов, списков, стеков, пакетов, логических значений, элементов рисования и растровых изображений. Не существует фиксированного ограничения на размер или количество пачек. Вы ограничены только памятью вашего устройства.

Значения устанавливаются и доступны с помощью клавиш. Ключ - это строка, которая идентифицирует значение. Например, пакет может содержать имя и фамилию человека. Ключами для доступа к этим строкам имен могут быть «first\_name» и «last\_name». Числовое значение возраста можно также поместить в Bundle с помощью клавиши «age».

Новый пустой пакет создается с помощью команды Bundle.create. Команда возвращает указатель на пустой пакет. Поскольку пакет представлен указателем, пакеты могут быть размещены в списках и массивах. Связки могут также содержаться в других связках. Это означает, что комбинация списков и пакетов может использоваться для создания сколь угодно сложных структур данных.

После создания пакета ключи и значения могут быть добавлены в пакет с помощью команды Bundle.put. Эти значения могут быть получены с помощью ключей в команде Bundle.get. Существуют и другие команды комплектов, облегчающие использование комплектов.

\*\*\*\*\*\*

##### Bundle.copy <SourcePointer\_nexp>, <DestinationPointer\_nexp>

Copies the source bundle to the destination Bundle. The background references are cut off!

\*\*\*\*\*\*

Копирует исходный комплект в целевой комплект. Фоновые ссылки обрезаны!

\*\*\*\*\*\*

##### Bundle.create <pointer\_nvar>

A new, empty bundle is created. The bundle pointer is returned in <pointer\_nvar>. Example:

\*\*\*\*\*\*

Новый пустой пакет создан. Указатель пакета возвращается в <pointer\_nvar>. Пример:

\*\*\*\*\*\*

BUNDLE.CREATE bptr

##### Bundle.type <pointer\_nexp>, <key\_sexp>, <type\_svar>

Returns the value type (string or numeric) of the specified key in the specified string variable. The <type\_svar> will contain an uppercase "N" if the type is numeric. The

<type\_svar> will contain an uppercase "S" if the type is a string. For arrays the dimensions are added like S[6] or N[2,2,5,..]. The other types result in "List", "Stack or Object" and "Bundle". If the bundle does not exist or does not contain the requested key, the command generates a run-time error. If you get "Stack or Object" and you are not sure, dump the Bundle with BUNDLE.DUMP.BUNDLE first. In doubt use the type Object.

\*\*\*\*\*\*

Возвращает тип значения (строковое или числовое) указанного ключа в указанной строковой переменной. <Type\_svar> будет содержать заглавную букву «N», если тип числовой.

<type\_svar> будет содержать заглавную букву "S", если тип является строкой. Для массивов размеры добавляются как S [6] или N [2,2,5, ..]. Другие типы приводят к «Списку», «Стеку или Объекту» и «Связке». Если пакет не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения. Если вы получили «Стек или объект» и не уверены, сначала выгрузите пакет с помощью BUNDLE.DUMP.BUNDLE. В случае сомнений используйте тип Object.

\*\*\*\*\*\*

Example:

BUNDLE.TYPE bptr, "age", type$

PRINT type$ % will print N, see also Bundle.put

##### Bundle.put <pointer\_nexp>, <key\_sexp>, <value\_nexp>|Array[]|<value\_sexp>|

###### *Array$[]*

The value expression or array will be placed into the specified bundle using the specified key. If the bundle does not exist, a new one may be created.

The type of the value will be determined by the type of the value expression.

\*\*\*\*\*\*

Выражение значения или массив будут помещены в указанный пакет, используя указанный ключ. Если пакет не существует, может быть создан новый.

Тип значения будет определяться типом выражения значения.

\*\*\*\*\*\*

Example:

BUNDLE.PUT bptr, "first\_name", "Frank" BUNDLE.PUT bptr,"age", 44

##### Bundle.PL <pointer\_nexp>, <key\_sexp>, <list\_ptr\_nexp>

Puts a list into a bundle. Read Bundle.Put.List instead Bundle.PL

The list will be placed into the specified bundle using the specified key. If the bundle does not exist, a new one may be created.

The type of the value is a list pointer.

\*\*\*\*\*\*

Размещает список в связку. Читайте Bundle.Put.List вместо Bundle.PL

Список будет помещен в указанный пакет, используя указанный ключ. Если пакет не существует, может быть создан новый.

Тип значения - указатель списка.

\*\*\*\*\*\*

Example:

BUNDLE.PL bptr, "names", llistPtr

##### Bundle.PS <pointer\_nexp>, <key\_sexp>, <stack\_ptr\_nexp>

Puts a stack into a bundle. Read Bundle.Put.Stack instead Bundle.PS

The stack will be placed into the specified bundle using the specified key. If the bundle does not exist, a new one may be created.

The type of the value is a stack pointer.

\*\*\*\*\*\*

Складывает стопку в пачку. Читайте Bundle.Put.Stack вместо Bundle.PS

Стек будет помещен в указанный пакет, используя указанный ключ. Если пакет не существует, может быть создан новый.

Тип значения - указатель стека.

\*\*\*\*\*\*

Example:

BUNDLE.PL bptr, "toDos", stackPtr

##### Bundle.PV <pointer\_nexp>, <key\_sexp>, <boolean\_nexp|Array[]>

Puts a **boolean** value or array into a bundle. Read Bundle.Put.Verum instead Bundle.PV

**V**erum is Latin for Truth.

The **boolean** will be placed into the specified bundle using the specified key. If the bundle does not exist, a new one may be created.

The type of the **boolean** value is a numeric expression or array. If <boolean\_nexp> or an entry of an array is > 0 the bundle entry will saved as **true**. Otherwise **false** is saved. The background references are cut off!

\*\*\*\*\*\*

Помещает логическое значение или массив в пакет. Читайте Bundle.Put.Verum вместо Bundle.PV

Верум на латыни означает Истина.

Логическое значение будет помещено в указанный пакет, используя указанный ключ. Если пакет не существует, может быть создан новый.

Тип логического значения - числовое выражение или массив. Если <boolean\_nexp> или запись массива> 0, запись пакета будет сохранена как true. В противном случае ложь сохраняется. Фоновые ссылки обрезаны!

\*\*\*\*\*\*

Example:

BUNDLE.PV bptr, "globals", 1

##### Bundle.PB <pointer\_nexp>, <key\_sexp>, <bundle\_pointer\_nexp>

Puts a bundle into a bundle. Read Bundle.Put.Bundle instead Bundle.PB

The bundle <bundle\_pointer\_nexp> will be placed into with <pointer\_nexp> specified bundle using the specified key. If the bundle specified <pointer\_nexp> does not exist, a new one may be created.

The type of the value is a bundle pointer. The background references are cut off!

\*\*\*\*\*\*

Складывает пачку в пачку. Читайте Bundle.Put.Bundle вместо Bundle.PB

Пакет <bundle\_pointer\_nexp> будет помещен в указанный пакет <pointer\_nexp>, используя указанный ключ. Если указанный пакет <pointer\_nexp> не существует, может быть создан новый.

Тип значения - указатель пакета. Фоновые ссылки обрезаны!

\*\*\*\*\*\*

Example:

BUNDLE.PL bptr, "globals", bundlePtr

##### Bundle.PJ <pointer\_nexp>, <json\_sexp>

Puts the content of a JSON string into a bundle. Read Bundle.Put.JSON instead Bundle.PJ

The first level of the JSON content is parsed to bundle-supported types.

Arrays of JSON objects will be stored as Arrays of strings and have to beparsed outside the bundle. Which should be also parsed asbundles.

\*\*\*\*\*\*

Помещает содержимое строки JSON в пакет. Читайте Bundle.Put.JSON вместо Bundle.PJ

Первый уровень содержимого JSON анализируется для типов, поддерживаемых пакетами.

Массивы объектов JSON будут храниться как массивы строк и должны быть обработаны вне пакета. Которые тоже должны быть разобраны на распайки.

\*\*\*\*\*\*

Example:

jsonString$ = XmlToJson$ ("<name>Nicolas</name>") % Native XML string

!Workaround for known bundle put issue:

! Fill empty Arrays with a "" member

jsonString$ = REPLACE$(jsonString$, "[]", "[" + "\"" + "\"" +"]")

! Now fill **empty Strings** with n.a.m. = "not a member"

jsonString$ = REPLACE$(jsonString$, "\"" + "\"", "\"" + "n.a.m."+ "\"")

BUNDLE.PJ bptr, jsonString$

BUNDLE.PJ bptr, "{\"age\":38, \"city\":\"Paris\"}" % Native JSON string BUNDLE.PV bptr, "male", 1 % Boolean is true

preJSON$ = " ’street’ : ’Main Street’ , ’no’ : 12 , ’FR’ : false “

BUNDLE.PJ bptr, "{" + Replace$(preJSON$, "’", chr$(34)) + "}" % chr$(34) equals “\”” BUNDLE.GJ bptr, newJson$, -1 % Returns a compact JSON string

PRINT newJson$ % Prints the result

“{"street":"Main Street","US":"true","no":12,"age":38,"city":"Paris","male":true,"name":"Nicolas"}”

See also XmlToJsonl$(), Is\_Json(), Is\_Xml()

##### Bundle.PP <pointer\_nexp>, <key\_sexp>, <bitmap\_pointer\_nexp> depreciatedGr.bitmap.put <bundle\_pointer\_nexp>, <key\_sexp>, <bitmap\_pointer\_nexp>

Puts a bitmap into a bundle.

The bitmap will be placed into the specified bundle using the specified key. If the bundle does not exist, a new one may be created.

The type of the value is a bundle pointer. The background references are cut off! Needs Graphic Mode.

\*\*\*\*\*\*

Помещает растровое изображение в пакет.

Растровое изображение будет помещено в указанный пакет, используя указанный ключ. Если пакет не существует, может быть создан новый.

Тип значения - указатель пакета. Фоновые ссылки обрезаны! Требуется графический режим.

\*\*\*\*\*\*

Example:

Gr.bitmap.put bptr, "picture1", bitmapPtr

##### Gr.drawable.put <bundle\_pointer\_nexp>, <key\_sexp>, <drawable\_pointer\_nexp>

Puts a drawable into a bundle.

The drawable will be placed into the specified bundle using the specified key. If the bundle does not exist, a new one may be created.

The type of the value is a bundle pointer. The background references are cut off! Needs Graphic Mode.

\*\*\*\*\*\*

Размещает Drawable в связку.

Drawable будет помещен в указанный пакет, используя указанный ключ. Если пакет не существует, может быть создан новый.

Тип значения - указатель пакета. Фоновые ссылки обрезаны! Требуется графический режим.

\*\*\*\*\*\*

Example:

Gr.drawable.put bptr, "picture1", drawablePtr

##### Bundle.get <pointer\_nexp>, <key\_sexp>, ~

**<value\_nexp>|Array[]|<value\_sexp>|Array$[]**

Places the value specified by the key string expression into the specified numeric or string variable. The type (array, string or numeric) of the destination variable must match the type stored with the key. The exception is an Object as an incoming data type. This Object will be converted as much as possible into a string value. If the bundle does not exist or does not contain the requested key, the command generates a run-time error.

\*\*\*\*\*\*

Помещает значение, указанное в строковом выражении ключа, в указанную числовую или строковую переменную. Тип (массив, строка или число) целевой переменной должен соответствовать типу, хранящемуся в ключе. Исключением является объект как тип входящих данных. Этот объект будет максимально преобразован в строковое значение. Если пакет не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения.

\*\*\*\*\*\*

Example:

BUNDLE.GET bptr,"first\_name", first\_name$ BUNDLE.GET bptr,"age", age

BUNDLE.GET bptr,"professions", professions$[]

##### Bundle.GL <pointer\_nexp>, <key\_sexp>, <list\_ptr\_nexp>

Read Bundle.Get.List instead Bundle.GL

Places the list specified by the key string expression into the pointer specified list. The type (string or numeric) of the destination list must match the type stored with the key. If the bundle does not exist or does not contain the requested key, the command generates a run-time error.

\*\*\*\*\*\*

Читайте Bundle.Get.List вместо Bundle.GL

Помещает список, указанный в ключевом строковом выражении, в указатель указанного списка. Тип (строковый или числовой) списка адресатов должен соответствовать типу, сохраненному в ключе. Если пакет не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения.

\*\*\*\*\*\*

Example:

BUNDLE.GET bptr,"names", listPtr

##### Bundle.GS <pointer\_nexp>, <key\_sexp>, <stack\_ptr\_nexp>

Read Bundle.Get.Stack instead Bundle.GS

Places the stack specified by the key string expression into the pointer specified stack. The type (string or numeric) of the destination stack must match the type stored with the key. If the bundle does not exist or does not contain the requested key, the command generates a run-time error.

\*\*\*\*\*\*

Читайте Bundle.Get.Stack вместо Bundle.GS

Помещает стек, указанный выражением строки ключа, в указатель указанного стека. Тип (строковый или числовой) целевого стека должен соответствовать типу, хранящемуся в ключе. Если пакет не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения.

\*\*\*\*\*\*

Example:

BUNDLE.GET bptr,"toDos", stackPtr

##### Bundle.GV <pointer\_nexp>, <key\_sexp>,<boolean\_nval|Array[]>

Read Bundle.Get.Verum instead Bundle.GV. **V**erum is Latin forTruth.

Places the **boolean** specified by the key string expression into with <boolean\_nval> specified numeric variable which can be also an array. If the **boolean** bundle content is **true** it returns 1. Otherwise is the content **false** 0 is returned. If the bundle does not exist or does not contain the requested key, the command generates a run-time error.

\*\*\*\*\*\*

Читайте Bundle.Get.Verum вместо Bundle.GV. Верум это латынь для истины.

Помещает логическое значение, указанное в выражении ключевой строки, в указанную числовую переменную <boolean\_nval>, которая также может быть массивом. Если булево содержимое пакета равно true, оно возвращает 1. В противном случае возвращается значение false 0. Если пакет не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения.

\*\*\*\*\*\*

Example:

BUNDLE.GET bptr,"boolean", myBoolean

##### Bundle.GB <pointer\_nexp>, <key\_sexp>, <bundle\_ptr\_nvar>

Read Bundle.Get.Bundle instead Bundle.GB

Places the bundle specified by the key string expression into the pointer

<bundle\_ptr\_nvar> specified bundle. If the bundle specified by <pointer\_nexp> does not exist or does not contain the requested key, the command generates a run-time error.

If the bundle specified by <**bundle\_ptr**\_nvar> does not exist a new one will be created.

\*\*\*\*\*\*

Читайте Bundle.Get.Bundle вместо Bundle.GB

Помещает в указатель пакет, указанный выражением ключевой строки

<bundle\_ptr\_nvar> указанный пакет. Если пакет, указанный в <pointer\_nexp>, не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения.

Если пакет, указанный в <bundle\_ptr\_nvar>, не существует, будет создан новый.

\*\*\*\*\*\*

Example:

BUNDLE.GET bptr,"bundleContainer", bundlePtr

##### Bundle.GJ <pointer\_nexp>, <json\_sexp>{, <spaces\_nexp>}

Gets the content of a bundle into a JSON string. Read Bundle.Get.JSON instead Bundle.GJ.

Supported data types are Strings, Doubles, Booleans as single value or Array.

If the optional <spaces\_nexp> returns a number > -1, a structural printout is delivered. The number of spaces defines the tabulator distance.

If <spaces\_nexp> is -1 or not given, a compact printout is returned.

There may be some post-processing needed, especially if JSON objects beginning with a "{" and ending with a "}" are involved.

Note that multi-dimensional arrays are converted into **one-dimensional** arrays! Example:

\*\*\*\*\*\*

Получает содержимое пакета в строку JSON. Прочитайте Bundle.Get.JSON вместо Bundle.GJ.

Поддерживаемые типы данных: Strings, Doubles, Booleans как одно значение или Array.

Если необязательный <space\_nexp> возвращает число> -1, структурная распечатка доставляется. Количество пробелов определяет расстояние табуляции.

Если <space\_nexp> равно -1 или не задано, возвращается компактная распечатка.

Может потребоваться некоторая постобработка, особенно если задействованы объекты JSON, начинающиеся с "{" и заканчивающиеся "}".

Обратите внимание, что многомерные массивы преобразуются в одномерные массивы!

\*\*\*\*\*\*

BUNDLE.GJ bptr, jsonString$ Byte.open w, ftb, path$+fileName$ Byte.write.buffer ftb, jsonString$ Byte.close ftb

See also JsonToXml$(), Is\_Json()

##### Bundle.GP <pointer\_nexp>, <key\_sexp>, <bitmap\_pointer\_nexp> depreciated

**Gr.bitmap.get <bundle\_pointer\_nexp>, <key\_sexp>, <bitmap\_ptr\_nexp>**

Gets a bitmap from a bundle.

Places the bitmap specified by the key string expression into the pointer specified bitmap. If the bundle does not exist or does not contain the requested key, the command generates a run-time error.

Needs Graphic Mode.

\*\*\*\*\*\*

Получает растровое изображение из пакета.

Помещает растровое изображение, указанное в ключевом строковом выражении, в указатель указанного растрового изображения. Если пакет не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения.

Требуется графический режим.

\*\*\*\*\*\*

Example:

Gr.bitmap.get bptr,"Picture2", bitmapPtr

##### Gr.drawable.get <bundle\_pointer\_nexp>, <key\_sexp>, <drawable\_ptr\_nexp>

Gets a drawable from a bundle.

Places the drawable specified by the key string expression into the pointer specified drawable. If the bundle does not exist or does not contain the requested key, the command generates a run-time error.

Needs Graphic Mode.

\*\*\*\*\*\*

Получает отрисовку из связки.

Помещает объект drawable, указанный в выражении ключевой строки, в указатель, указанный drawable. Если пакет не существует или не содержит запрошенный ключ, команда генерирует ошибку во время выполнения.

Требуется графический режим.

\*\*\*\*\*\*

Example:

Gr.drawable.get bptr,"Picture2", drawablePtr

##### Bundle.in <recAction\_sexp>, <retData\_svar>, <retBundleIndex\_nvar>

Receives an Intent from a calling app or launcher. The parameter recAction returns the calling action. With retData you get the Data URI string from the Intent. The retBundleIndex returns a bundle. If no data is broadcasting retData returns ""and retBundleIndex returns an empty Bundle. Use DECODE$ with the type "URL" and the Qualifier "charset" like DECODE$ ("URL", "charset", retData$) if necessary.

Parameters in the retData String have to be in URI style, only.

The returned Bundle with the retBundleIndex pointer contains the received Extras.

\*\*\*\*\*\*

Получает Намерение от вызывающего приложения или модуля запуска. Параметр recAction возвращает вызывающее действие. С помощью retData вы получаете строку URI данных из намерения. RetBundleIndex возвращает пакет. Если данные не передаются, retData возвращает "", а retBundleIndex возвращает пустой Bundle. При необходимости используйте DECODE $ с типом «URL» и классификатором «charset», например DECODE $ («URL», «charset», retData $).

Параметры в строке retData должны быть только в стиле URI.

Возвращенный Bundle с указателем retBundleIndex содержит полученные дополнения.

\*\*\*\*\*\*

##### GitHub#174

**Bundle.out <sendAction\_sexp>, <sendData\_sexp>,**

**<sendExtraBundle\_pointer\_nexp>**

Sends a result as an Intent return to a calling app or launcher, when the program is finished. The App or BASIC! has to be called **inclusive component name** (normally package name +”.Basic”). If you launch the program in the Editor, you get normally **no result**.

The parameter sendAction sends an action (extremely seldom) back. With sendData you send a Data Extra string (seldom) back. Android accept only URIs! The retBundleIndex sends a bundle back.

The sent Bundle with the sendExtraBundleIndex pointer contains the sent Extras. In this case we have two bundle levels:

First level:

Supporting **only** the Intent Extras Types **Double, String and Bundle. Other types are automatically removed.**

Second level:

If you use on the first level a Bundle that contains also one or more Bundles, you can use these as full featured BASIC! Bundles.

\*\*\*\*\*\*

Посылает результат как возврат Intent в вызывающее приложение или модуль запуска, когда программа завершена. Приложение или бейсик! должно называться включающим имя компонента (обычно имя пакета + ”. Basic”). Если вы запустите программу в редакторе, вы, как правило, не получите результата.

Параметр sendAction отправляет действие (крайне редко) назад. С sendData вы отправляете строку данных Extra (редко) обратно. Android принимает только URI! RetBundleIndex отправляет пакет обратно.

Отправленный пакет с указателем sendExtraBundleIndex содержит отправленные дополнения. В этом случае у нас есть два уровня связки:

Первый уровень:

Поддерживаются только типы Intent Extras Double, String и Bundle. Другие типы автоматически удаляются.

Второй уровень:

Если вы используете на первом уровне комплект, содержащий также один или несколько комплектов, вы можете использовать их как полнофункциональный бейсик! Связки.

\*\*\*\*\*\*

##### Bundle.save <pointer\_nexp>, <fileName\_sexp>

Saves a bundle into a file. Only recommended for temporary use, because the internal coding of the result is operating system version depended. It seems, that newer Android versions are able to read older bundle files. This is important, if the user want to upgrade his operating system. But this behavior is without any guarantee.

Bitmaps in bundles are also not supported in this case. Maybe there is a size limitation. Not tested yet.

\*\*\*\*\*\*

Сохраняет пакет в файл. Рекомендуется только для временного использования, поскольку внутренняя кодировка результата зависит от версии операционной системы. Кажется, что новые версии Android могут читать старые файлы пакета. Это важно, если пользователь хочет обновить свою операционную систему. Но это поведение без каких-либо гарантий.

Растровые изображения в пакетах также не поддерживаются в этом случае. Может быть, есть ограничение по размеру. Еще не проверено.

\*\*\*\*\*\*

Example:

BUNDLE.SAVE bptr, "saveState.bun"

See also Bundle.PJ, Byte.write.buffer, Grabfile, Bundle.GJ

##### Bundle.load <pointer\_nexp>, <fileName\_sexp>

Loads a bundle from a file. Only recommended for temporary use, because the internal coding is operating system version depended.

Bitmaps in bundles are also not supported in this case. Maybe there is a size limitation. Not tested yet.

\*\*\*\*\*\*

Загружает пакет из файла. Рекомендуется только для временного использования, поскольку внутренняя кодировка зависит от версии операционной системы.

Растровые изображения в пакетах также не поддерживаются в этом случае. Может быть, есть ограничение по размеру. Еще не проверено.

\*\*\*\*\*\*

Example:

BUNDLE.LOAD bptr, "saveState.bun"

##### Bundle.remove <pointer\_nexp>, <key\_sexp>

Removes the key named by the string expression <key\_sexp>, along with the associated value, from the bundle pointed to by the numeric expression <pointer\_nexp>. If the bundle does not contain the key, nothing happens. If the bundle does not exist, a new one may be created.

\*\*\*\*\*\*

Удаляет ключ, названный строковым выражением <key\_sexp>, вместе со связанным значением из пакета, на который указывает числовое выражение <pointer\_nexp>. Если в комплекте нет ключа, ничего не происходит. Если пакет не существует, может быть создан новый.

\*\*\*\*\*\*

##### Bundle.clear <pointer\_nexp>

The bundle pointed to by <pointer\_nexp> will be cleared of all tags. It will become an empty bundle. If the bundle does not exist, a new one may be created.

\*\*\*\*\*\*

Пакет, на который указывает <pointer\_nexp>, будет очищен от всех тегов. Это станет пустой связкой. Если пакет не существует, может быть создан новый.

\*\*\*\*\*\*

##### Debug.dump.bundle <bundlePtr\_nexp>

Dumps the Bundle pointed to by the Bundle Pointer numeric expression.

If a Bundle contains other types as single numeric values from type Double or String, this value will be converted and printed as a string without quotation marks.

\*\*\*\*\*\*

Дампы Bundle, на которые указывает числовое выражение Bundle Pointer.

Если Bundle содержит другие типы в виде отдельных числовых значений типа Double или String, это значение будет преобразовано и напечатано в виде строки без кавычек.

\*\*\*\*\*\*

Email.send <recipient\_sexp>|Array$[], <subject\_sexp>, <body\_sexp>~

**{{{{,<sendTo\_sexp>},<cC\_sexp>|Array$[]},<bCC\_sexp>|Array$[]},~**

**<attachment\_svar>|Array$[]}**

The email message in the Body string expression will be sent to the named recipient(s) with the named subject heading.

<sendTo\_sexp> specifies an app which should send. (Ex.: Gmail, WhatsApp, DropBox or OneDrive. See also App.installed.) <cC\_sexp> and <bCC\_sexp> place the CC and BCC recipient(s) in the mail header. <attachment\_svar> specifies the attachment(s). For a single file only a single String variable is accepted.

The command expects double quotes for undefined arguments.

Files from the \_Internal file folder have to be copied into an external folder before sending.

\*\*\*\*\*\*

Сообщение электронной почты в строковом выражении будет отправлено указанному получателю (ям) с указанным заголовком.

<sendTo\_sexp> указывает приложение, которое должно быть отправлено. (Например: Gmail, WhatsApp, DropBox или OneDrive. См. Также App.installed.) <CC\_sexp> и <bCC\_sexp> помещают получателей CC и BCC в заголовок письма. <attachment\_svar> определяет вложение (я). Для одного файла допускается только одна строковая переменная.

Команда ожидает двойные кавычки для неопределенных аргументов.

Файлы из папки \_Internal file должны быть скопированы во внешнюю папку перед отправкой.

\*\*\*\*\*\*

Example:

gmail$ = "[c](mailto:harry.c@examples.com)om.google.android.gm"

Array.load cC$[], "**[bob.ex@example.com](mailto:bob.ex@ex.com)**", "**[harry.c@examples.com](mailto:harry.c@examples.com)**" FILE.ROOT dataPath$

Array.load files$[], "file://" + dataPath$ + "/" + "cartman.png",~ "file://" + dataPath$ + "/" + "whee.mp3"

Email.send "**[b.ex@gmail.com](mailto:b.ex@gmail.com)**","Mail Subject","Message", gmail$, cC$[], **""**, files$[]

GitHub#153

##### Pause <ticks\_nexp>

Stops the execution of the BASIC! program for <ticks\_nexp> milliseconds. One millisecond = 1/1000 of a second. Pause 1000 will pause the program for one second. **A pause can not be interrupted.**

An infinite loop can be a very useful construct in your programs. For example, you may use it to wait for the user to tap a control on the screen. A tight spin loop keeps BASIC! very busy doing nothing. A **Pause**, even a short one, reduces the load on the CPU and the drain on the battery. Depending on your application, you may want to add a **Pause** to the loop to conserve battery power:

\*\*\*\*\*\*

Останавливает исполнение бейсика! программа на <ticks\_nexp> миллисекунды. Одна миллисекунда = 1/1000 секунды. Пауза 1000 приостановит программу на одну секунду. Пауза не может быть прервана.

Бесконечный цикл может быть очень полезной конструкцией в ваших программах. Например, вы можете использовать его, чтобы дождаться, когда пользователь коснется элемента управления на экране. Плотная спин-петля держит бейсик! очень занят, ничего не делая. Пауза, даже короткая, снижает нагрузку на процессор и разряд батареи. В зависимости от вашего приложения вы можете добавить паузу в цикл для экономии заряда батареи:

\*\*\*\*\*\*

DO : PAUSE 50 : UNTIL x <> 0

### Interrupt Labels (Event Handlers)

You can perform physical actions that tell your BASIC! program to do something. When you touch the screen or press a key you cause an *event*. These events are *asynchronous*, that is, they happen at times your program cannot predict. BASIC! detects some events so your program can respond to them.

BASIC! handles events as *interrupts*. Each event that BASIC! recognizes has a unique *Interrupt Label*. When an event occurs, BASIC! looks for the Interrupt Label that matches the event.

* If you have not written that Interrupt Label into your program, the event is ignoredand your program goes on running as if nothinghappened.
* If you have included the right Interrupt Label for the event, BASIC! jumps to thatlabel and continues execution at the line after the label. This is called trapping theevent.

BASIC! does not necessarily respond to the event as soon as it occurs. The statement that is executing when the event occurs is allowed to complete, then BASIC! jumps to the Interrupt Label.

When you use an Interrupt Label to trap an event, BASIC! executes instructions until it finds a *Resume* command that matches the Interrupt Label. During that time, it records other events but it does not respond to them. The block of code between the Interrupt Label and the matching Resume may be called an *Interrupt Service Routine (ISR)* or, if you prefer, an *Event Handler*.

**In this version variables in functions** are per **default local** if an **interrupts traps**! If you get in trouble with older code, use in addition GLOBASL.ALL after the colon (on….:) and GLOBALS.NONE before RESUME.

When BASIC! executes the event’s Resume command, it resumes normal execution.

* BASIC! jumps back to where it was running when the interruptoccurred.
* BASIC! again responds to other events, including any that occurred while it was handling an event. \*\*\* In other words events are be put in a waiting queue. Thus many events can be triggered and will be handled on the first in first out principle, but OnError: is always servicedfirst.

For the interrupt handling to be as fast as possible:

* Use a timer interrupt instead of pause in your main loop, that solves a lot ofinterrupt issues, because PAUSE also delays theinterrupts.
* In this version you can use Sched.set as second timer,too.

\*\*\* Need more investigation

An Interrupt Label looks and behaves just like any other label in BASIC!. However, you must not execute any of the Resume commands except to finish an event’s handler.

##### All Interrupt Labels

BASIC! supports trapping of the following events. These interrupt labels and their Resume commands are described in various parts of this manual.

\*\*\*\*\*\*

Вы можете выполнять физические действия, которые говорят вашему бейсику! Программа, чтобы сделать что-то. Когда вы касаетесь экрана или нажимаете клавишу, вы вызываете событие. Эти события асинхронны, то есть они случаются в то время, когда ваша программа не может предсказать. BASIC! обнаруживает некоторые события, чтобы ваша программа могла на них реагировать.

BASIC! обрабатывает события как прерывания. Каждое событие, что Бейсик! распознает имеет уникальную метку прерывания. Когда происходит событие, BASIC! ищет метку прерывания, соответствующую событию.

Если вы не вписали эту метку прерывания в вашу программу, событие игнорируется, и ваша программа продолжает работать, как будто ничего не произошло.

Если вы включили правильную метку прерывания для мероприятия, BASIC! переходит на эту метку и продолжает выполнение на строке после метки. Это называется ловушкой события.

BASIC! не обязательно реагировать на событие, как только оно происходит. Оператор, который выполняется при возникновении события, может завершиться, затем BASIC! прыгает к метке прерывания.

Когда вы используете метку прерывания для захвата события, BASIC! выполняет инструкции, пока не найдет команду Resume, соответствующую метке прерывания. В течение этого времени он записывает другие события, но не отвечает на них. Блок кода между меткой прерывания и соответствующим резюме может называться подпрограммой обработки прерывания (ISR) или, если вы предпочитаете, обработчиком события.

В этой версии переменные в функциях по умолчанию локальные, если прерывание прерывается! Если у вас возникли проблемы со старым кодом, используйте дополнительно GLOBASL.ALL после двоеточия (on…. :) и GLOBALS.NONE перед RESUME.

Когда бейсик! выполняет команду Resume события, она возобновляет нормальное выполнение.

BASIC! возвращается к тому месту, где он работал, когда произошло прерывание.

BASIC! снова реагирует на другие события, в том числе те, которые произошли во время обработки события. \*\*\* Другими словами, события помещаются в очередь ожидания. Таким образом, многие события могут быть инициированы и будут обрабатываться по принципу «первым пришел - первым вышел», но OnError: всегда обслуживается первым.

Чтобы обработка прерываний была максимально быстрой:

Используйте прерывание по таймеру вместо паузы в вашем основном цикле, что решает множество проблем с прерываниями, потому что PAUSE также задерживает прерывания.

В этой версии вы также можете использовать Sched.set в качестве второго таймера.

\*\*\* Нужно больше расследований

Метка прерывания выглядит и ведет себя так же, как и любая другая метка в Бейсике !. Однако вы не должны выполнять какие-либо команды Resume, кроме как для завершения обработчика события.

Все ярлыки прерываний

BASIC! поддерживает отслеживание следующих событий. Эти метки прерываний и их команды возобновления описаны в различных частях данного руководства.

\*\*\*\*\*\*

##### OnBackground: Background.resume

* **OnBackKey: Back.resume**
* **OnBroadcast: Broadcast.resume**
* **OnBtReadReady: Bt.onReadReady.resume**
* **OnConsoleTouch: ConsoleTouch.resume**
* **OnError:** None (not a true event, see **OnError:**,below)

##### OnGrTouch: Gr.onGrTouch.resume

* **OnGrTouchMove: Gr.onGrTouchMove.resume**
* **OnGrTouchUp: Gr.onGrTouchUp.resume**
* **OnGrScreen: Gr.onGrScreen.resume**
* **OnKbChange: Kb.resume**
* **OnKeyDown: KeyDown.resume**
* **OnKeyPress: Key.resume**
* **OnLowMemory: LowMemory.resume**
* **OnMenuKey: MenuKey.resume**
* **OnMenuItem: MenuItem.resume**
* **OnSched: Sched.resume**
* **OnTimer: Timer.resume**
* **OnHtmlReturn: Html.onHtmlReturn.resume**

**OnError:**

Special interrupt label that traps a run-time error as if it were an event, except that:

* **OnError:** has no matching Resume command. You can use **GoTo** to jump anywhere in your program.
* **OnError:** is not locked out byother interrupts.
* **OnError:** does not lock out otherinterrupts.

If a BASIC! program does not have an **OnError:** label and an error occurs while the program is running, an error message is printed to the Output Console and the program stops running.

If the program does have an **OnError:** label, BASIC! does not stop on an error. Instead, it jumps to the **OnError:** label like a GOTO instead of a GOSUB/RETURN (see "Interrupt Labels", above). The error message is not printed, but it can be retrieved by the **GETERROR$()** function. If you jumps from a function its variables are still local in its area also.

Be careful. An infinite loop will occur if a run-time error occurs within the **OnError:** code. You should not place an **OnError:** label into your program until the program is fully debugged. Premature use of **OnError:** will make the program difficult to debug.

\*\*\*\*\*\*

Специальная метка прерывания, которая перехватывает ошибку во время выполнения, как если бы это было событие, за исключением того, что:

OnError: не имеет соответствующей команды Resume. Вы можете использовать GoTo, чтобы перейти в любую точку вашей программы.

OnError: не блокируется другими прерываниями.

OnError: не блокирует другие прерывания.

Если бейсик! Программа не имеет метки OnError: и во время работы программы возникает ошибка, на выходную консоль выводится сообщение об ошибке, и программа останавливается.

Если в программе есть метка OnError: BASIC! не останавливается на ошибке. Вместо этого он переходит к метке OnError: как GOTO вместо GOSUB / RETURN (см. «Метки прерываний» выше). Сообщение об ошибке не печатается, но его можно получить с помощью функции GETERROR $ (). Если вы выпрыгиваете из функции, ее переменные все еще локальны в своей области.

Быть осторожен. Бесконечный цикл произойдет, если в коде OnError: возникнет ошибка во время выполнения. Вы не должны помещать метку OnError: в свою программу, пока программа не будет полностью отлажена. Преждевременное использование OnError: затруднит отладку программы.

Example:

REM Function Exeption Handler FN.DEF myFunc()

func$ = "myFunc"

p = p + 4 : z = 7 : k = 5

p$ = p % Line with Error code. labelGoOn\_myFunc:

PRINT p, z, k FN.END

p =23

z =44

k =99

myFunc() END

ONERROR:

IF func$ = "myFunc" PRINT GETERROR$()

! GLOBALS.FNIMP z % GLOBALS.FNIMP, GLOBALS.ALL and GLOBALS.NONE

! together are not possible in the same function

PRINT p % returns 4, because still in the function myFunc

PRINT z % returns 44, because imported in the area of the function PRINT k % returns 0, because not imported

GLOBALS.ALL % Usefull for exeption handling, the value of the variable is **not** imported! PRINT k % returns 99, because access to the global symbol table

GLOBALS.NONE

GOTO labelGoOn\_myFunc ENDIF

##### OnKeyDown:

Interrupt label that traps a tap\_down on any key except the volume keys. They have to switched on with VolKeys.On. BASIC! executes the statements following the **OnKeyDown:** label until it reaches a **KeyDown.resume**.

\*\*\*\*\*\*

Метка прерывания, которая удерживает нажатие tap\_down на любую клавишу, кроме клавиш регулировки громкости. Они должны быть включены с VolKeys.On. BASIC! выполняет операторы после метки OnKeyDown:, пока не достигнет KeyDown.resume.

\*\*\*\*\*\*

##### KeyDown.resume

Resumes execution at the point in the BASIC! program where the **OnKeyPress:** interrupt occurred.

\*\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, в которой произошло прерывание OnKeyPress:

##### OnKeyPress:

Interrupt label that traps a tap\_up on any key. BASIC! executes the statements following the **OnKeyPress:** label until it reaches a **Key.resume**. (OnKeyUp would describe it better.)

\*\*\*\*\*\*

Метка прерывания, которая захватывает tap\_up на любую клавишу. BASIC! выполняет операторы, следующие за меткой OnKeyPress:, пока не достигнет Key.resume. (OnKeyUp описал бы это лучше.)

\*\*\*\*\*\*

##### Key.resume

Resumes execution at the point in the BASIC! program where the **OnKeyPress:** interrupt occurred.

\*\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, в которой произошло прерывание OnKeyPress:

##### OnMenuItem:

Interrupt label that traps if a menu item is selected. BASIC! executes the statements following the **OnMenuItem:** label until it reaches a **OnMenuItem.resume**.

\*\*\*\*\*\*

Метка прерывания, которая прерывается, если выбран пункт меню. BASIC! выполняет операторы, следующие за меткой OnMenuItem:, пока не достигнет OnMenuItem.resume.

\*\*\*\*\*\*

##### MenuItem.get.datalink <data\_svar>

Returns the data of the last selected menu item in a human readable JSON string.

\*\*\*\*\*\*

Возвращает данные последнего выбранного пункта меню в удобочитаемой строке JSON.

\*\*\*\*\*\*

Begin: "{" +~

"\_MenuFrom:\_Graphic" +~ % Other cases are "\_Console" and "\_HTML" "\_TitleText:MyItemTitle" +~ % If is "\_TitleUp" returned, the Home Button of

% the Title Bar is selected!

"\_ItemId:20" +~ % Returns the itemid

"\_GroupId:1"+~ % Returns the group id "\_Checked:1"+~ % Returns if checked 1 or not 0 "\_Checkable:0"+~ % Returns if it is check-able 1 or not0;

End: "}"

##### MenuItem.resume

Resumes execution at the point in the BASIC! program where the **OnMenuItem:** interrupt occurred.

\*\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, в которой произошло прерывание OnMenuItem:

##### Locals.on

After this command the variables in functions are definitely **local.**

**\*\*\*\*\*\***

**После этой команды переменные в функциях определенно являются локальными.**

**\*\*\*\*\*\***

##### Locals.off

After this command the variables in functions can be **global**, if Globals.all is used.

\*\*\*\*\*\*

После этой команды переменные в функциях могут быть глобальными, если используется Globals.all.

\*\*\*\*\*\*

##### Globals.all

After this command all variables in functions are **global.**

With Locals.on and Locals.off you are able to change variables in parts or in the entire function to local.

**Be very carefully with this command!** With the **Include** command and some libs like GW.bas you will get in trouble if you do not put the lib calls, **Locals.on** and **Locals.off**, in brackets. **The use inside from functions is strongly not recommended, because the results are wrong if the function is a part of a function call chain inside an interrupt routine.**

A better alternative is to use bundles. They are global, thus a global container with all the variable types is available. Take care if you including a library like GW.bas. It uses the 1 as the global bundle pointer. Thus use at the first line in your main program part INCLUDE “GW.bas” and afterwards BUNDLE.CREATE myGlobals. In this case myGlobals is 2.

Setting the global bundle pointer into FN.DEF (myGlobals, …) makes your code readable. See also Globals.fnimp

\*\*\*\*\*\*

После этой команды все переменные в функциях являются глобальными.

С Locals.on и Locals.off вы можете изменить переменные по частям или во всей функции на локальные.

Будьте очень осторожны с этой командой! С командой Include и некоторыми библиотеками, такими как GW.bas, у вас будут проблемы, если вы не заключите в скобки вызовы lib, Locals.on и Locals.off. Использование внутри функции настоятельно не рекомендуется, поскольку результаты неверны, если функция является частью цепочки вызовов функций внутри подпрограммы прерывания.

Лучшая альтернатива - использовать связки. Они являются глобальными, поэтому доступен глобальный контейнер со всеми типами переменных. Будьте осторожны, если вы включите библиотеку, такую ​​как GW.bas. Он использует 1 как глобальный указатель пакета. Таким образом, используйте в первой строке вашей основной части программы INCLUDE «GW.bas», а затем BUNDLE.CREATE myGlobals. В этом случае myGlobals равен 2.

Установка глобального указателя пакета в FN.DEF (myGlobals,…) делает ваш код читабельным. Смотрите также Globals.fnimp

\*\*\*\*\*\*

##### Globals.none

After this command the variables in functions are **local**, except the variables imported by Globals.fnimp.

\*\*\*\*\*\*

После этой команды переменные в функциях являются локальными, за исключением переменных, импортированных Globals.fnimp.

\*\*\*\*\*\*

##### Globals.fnimp <varexp> {… , <varexp>}, ...

Imports variables from the main program area in a function. Thus you have access to the specified variables from the main area.

Credits to Humpty for this enhancement.

GLOBALS.FNIMP, GLOBALS.ALL and GLOBALS.NONE together are not possible in the same function.

\*\*\*\*\*\*

Импортирует переменные из основной области программы в функцию. Таким образом, у вас есть доступ к указанным переменным из основной области.

Кредиты Шалтай для этого улучшения.

GLOBALS.FNIMP, GLOBALS.ALL и GLOBALS.NONE вместе невозможны в одной и той же функции.

### Broadcasts

Broadcasts are messages from the system or other applications (activities). You can send and receive these messages with broadcasts. Before you are able to receive broadcasts, youhavetoinitializeasocalledBroadcastReceiver.TheBroadcastMessageiscalledan Intent. The Intent will be received by any application that has the right IntentFilter.

If you send messages between instances of BASIC! (different package IDs are needed)at runtime, maybe use this actionaddresses:

* Prog 1 use for sending "com.rfo.basicFellow.broadcast.SEND1" andreceiving "com.rfo.basicOli.broadcast.SEND2"
* Prog 2 use for sending "com.rfo.basicOli.broadcast.SEND2" andreceiving "com.rfo.basicFellow.broadcast.SEND1"

The use of APP.Broadcast is not recommended, but APP.SAR can be used to send a Broadcast with much more options.

\*\*\*\*\*\*

Трансляции - это сообщения из системы или других приложений (действий). Вы можете отправлять и получать эти сообщения с помощью трансляций. Прежде чем вы сможете получать широковещательные рассылки, вы должны начать инициализацию BroadcastReceiver.TheBroadcastMessageiscalledan Intent. Намерение будет получено любым приложением, имеющим право IntentFilter.

Если вы отправляете сообщения между экземплярами BASIC! (нужны разные идентификаторы пакетов) во время выполнения, возможно, используйте это actionaddresses:

Prog 1 используется для отправки "com.rfo.basicFellow.broadcast.SEND1" и получения "com.rfo.basicOli.broadcast.SEND2"

Prog 2 используется для отправки "com.rfo.basicOli.broadcast.SEND2" и получения "com.rfo.basicFellow.broadcast.SEND1"

Использование APP.Broadcast не рекомендуется, но APP.SAR можно использовать для отправки широковещательной рассылки с гораздо большим количеством опций.

\*\*\*\*\*\*

Example:

LIST.CREATE S, commandListPointer LIST.ADD commandListPointer~

"new Intent("+CHR$(34)+"com.rfo.basic.broadcast.SEND"+CHR$(34)+");" ~

! → **);" ~ ←** is important "EOCL"

BUNDLE.PL appVarPointer,"\_CommandList",commandListPointer BUNDLE.Put appVarPointer,"\_Broadcast",""

APP.SAR appVarPointer

##### Broadcast.init <action\_sexp> | Array$[]

Initializes a Broadcast Receiver

\*\*\*\*\*\*

Инициализирует широковещательный приемник

##### OnBroadcast:

Interrupt label that traps a received broadcast. BASIC! executes the statements following the **OnBroadcast**: label until it reaches a **Broadcast.resume**.

\*\*\*\*\*\*

Метка прерывания, которая перехватывает полученную трансляцию. BASIC! выполняет операторы после метки OnBroadcast: до тех пор, пока не достигнет Broadcast.resume.

\*\*\*\*\*\*

##### Broadcast.in <recAction\_sexp>, <retData\_svar>, <retBundleIndex\_nvar>

Receives a Broadcast message (Intent) according to the recAction action adress. With retData you get the Data Extra string from the Intent. The retBundleIndex returns a bundle. If no data is broadcasting retData returns "" and retBundleIndex returns an empty Bundle. You have also the option to detect system broadcasts, but in some cases you need also the right permissions. As an example you need for the broadcast filter "android.net.conn.INET\_CONDITION\_ACTION" the new permission ACCESS\_NETWORK\_STATE. Today the two available compilers are still not capable for broadcast permission autodetection. That is your job. A compressed information you can find under: https://android.googlesource.com/platform/frameworks/base/+/master/core/res/ AndroidManifest.xml

\*\*\*\*\*\*

Получает широковещательное сообщение (Intent) в соответствии с адресом действия recAction. С помощью retData вы получаете строку Data Extra из Intent. RetBundleIndex возвращает пакет. Если данные не передаются, retData возвращает "", а retBundleIndex возвращает пустой Bundle. У вас также есть возможность обнаруживать системные трансляции, но в некоторых случаях вам также нужны правильные разрешения. В качестве примера вам понадобится для широковещательного фильтра "android.net.conn.INET\_CONDITION\_ACTION" новое разрешение ACCESS\_NETWORK\_STATE. Сегодня два доступных компилятора по-прежнему не способны к автоопределению разрешений на вещание. Это твоя работа. Сжатую информацию вы можете найти по адресу: https://android.googlesource.com/platform/frameworks/base/+/master/core/res/ AndroidManifest.xml

\*\*\*\*\*\*

##### Broadcast.resume

Resumes execution at the point in the BASIC! program where the **OnBroadcast:** interrupt occurred.

\*\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, в которой произошло прерывание OnBroadcast:

##### Broadcast.close

Closes the Broadcast Receiver

\*\*\*\*\*\*

Закрывает приемник

##### Broadcast.bundle <sndAction\_sexp>, <key\_sexp>, <bundle\_ptr\_nvar>

Sends a Broadcast message as a Bundle with the action address sndAction and the key.

\*\*\*\*\*\*

Посылает широковещательное сообщение как пакет с адресом действия sndAction и ключом.

##### Broadcast.string <sndAction\_sexp>, <key\_sexp>, <msg\_sexp>

Sends a Broadcast message as a String with the action address sndAction and the key.

\*\*\*\*\*\*

Посылает широковещательное сообщение в виде строки с адресом действия sndAction и ключом.

##### KB.send.keyevent <tapType\_nexp>, <actionType\_sexp>, <keyCode\_nexp>

Sends a key event internally as a Broadcast message to the environment.

\*\*\*\*\*\*

Внутренняя отправка ключевого события в виде широковещательного сообщения в среду.

|  |  |
| --- | --- |
| Supported tap types <tapType\_nexp> | |
| 0 | Key down |
| 1 | Key up |
| 4 | Key down and up |

|  |  |
| --- | --- |
| Action type group <actionType\_sexp> | |
| Value | Intent action (informative) |
| \_Global (no permission) | android.intent.action.GLOBAL\_BUTTON |
| \_Call | android.intent.action.CALL\_BUTTON |
| \_Media | android.intent.action.MEDIA\_BUTTON |
| \_Camera | android.intent.action.CAMERA\_BUTTON |

For key events <keyCode\_nexp> see: [https://android.googlesource.com/platform/frameworks/base/+/master/core/java/android/view/KeyEvent.java](https://android.googlesource.com/platform/frameworks/base/%2B/master/core/java/android/view/KeyEvent.java)

Mostly apps ignore these events. Try some media player like VLC, Gplayer or TotalCommander.

\*\*\*\*\*\*

В основном приложения игнорируют эти события. Попробуйте какой-нибудь медиаплеер, например VLC, Player или Total Commander.

Example:

upAndDown = 4 type$ = "\_media"

keyCode = 127 %Media PAUSE button kb.send.keyevent upAndDown, txpe$, keyCode pause 3000

kb.send.keyevent upAndDown, txpe$, keyCode

##### Inkey$ <svar>{{, <rawKeyEvent\_svar>}, <utf-8\_svar>}

Reports key taps for the a-z, 0-9, Space and the D-Pad keys. The key value is returned in

<svar>.

The D-Pad keys are reported as "up", "down", "left", "right" and "go". If any key other than those have been tapped, the string "key nn" will be returned. Where nn will be theAndroid key code for that key.

If no key has been tapped, the "@" character is returned in <svar>.

Keep in mind, that **soft** keyboards send a limited character set. Characters like

"°♤♡ ◇♧《》¡¿äöü" are only supported by **USB** or **Bluetooth** keyboards or other input devices like game pads in the case of this function.

Rapid key taps are buffered in case they come faster than the BASIC! program can process.

With rawKeyEvent you get an optional raw key event description with action,keyCode, scanCode, metaState, flags, repeatCount, eventTime, downTime, deviceId andsource values.

As an option you get with <utf-8\_svar> the UTF-8 character back.

If you want correct results use ONKEY…: interrupt handling instead a DO – UNTIL loop. But do not use the command PAUSE if have a lot keystrokes in conjunction with key event handling.

\*\*\*\*\*\*

Сообщает о нажатиях клавиш для клавиш a-z, 0-9, пробела и D-Pad. Значение ключа возвращается в

<SVAR>.

Клавиши D-Pad обозначаются как «вверх», «вниз», «влево», «вправо» и «идти». Если какая-либо клавиша, кроме этих, была нажата, будет возвращена строка «ключ nn». Где nn будет кодом ключа Android для этого ключа.

Если ни одна клавиша не была нажата, символ «@» возвращается в <svar>.

Имейте в виду, что программные клавиатуры отправляют ограниченный набор символов. Персонажи как

«° ♤ ♡ ◇ ♧ 《》 ¡¿äöü» поддерживаются только клавиатурами USB или Bluetooth или другими устройствами ввода, такими как игровые приставки, в случае этой функции.

Быстрые нажатия клавиш буферизуются в случае, если они идут быстрее, чем BASIC! Программа может обрабатывать.

С rawKeyEvent вы получаете необязательное необработанное описание события ключа с параметрами action, keyCode, scanCode, metaState, flags, repeatCount, eventTime, downTime, deviceId и значениями источника.

Как вариант, вы получаете с <utf-8\_svar> символ UTF-8 обратно.

Если вы хотите получить правильные результаты, используйте ONKEY…: обработку прерываний вместо цикла DO - UNTIL. Но не используйте команду PAUSE, если много нажатий клавиш в сочетании с обработкой событий клавиш.

Example

KEYDOWN.ON % The opposite is KEYDOWN.OFF DO

UNTIL 0

ONKEYDOWN: %A Key Is Down interrupt

! The second string, raw key event parameter INKEY$ mKey$, mKeyEvent$, mUniKeyEvent$

PRINT "Got "; mKey$, " "; mKeyEvent$, mUniKeyEvent$

KEYDOWN.RESUME %Resumes execution at the point BASIC! program where

! the OnKeyDown: interrupt occured.

ONKEYPRESS: %Imo ONKEYUP points the fact better INKEY$ mKey$, mKeyEvent$, mUniKeyEvent$

PRINT "Got "; mKey$, " "; mKeyEvent$, mUniKeyEvent$ KEY.RESUME

##### Keydown.on

Key Down Detection possible.

\*\*\*\*\*\*

Обнаружение ключа вниз возможно.

##### Keydown.off

No Key Down Detection. Default status.

\*\*\*\*\*\*

Обнаружение без нажатия клавиш. Статус по умолчанию.

##### Gr.set.dashpatheffect {<intervals\_list\_ptr\_nvar>{, <phase\_nexp>}}

Path Pattern:

The intervals list must contain an even number of entries (≥ 2), with the even indices specifying the "on" intervals, and the odd indices specifying the "off" intervals.

Phase:

Phase is an offset into the intervals list (mod the sum of all of the intervals).

The intervals list controls the length of the dashes.

The GR.set.stroke controls the thickness of the dashes.

Note: This path effect only affects drawing with the GR.color's style parameter

is set to 0 (STROKE) or 2 (STROKE\_AND\_FILL). It is ignored if the drawing is done with style == 1 (FILL).

A simple Gr.set.dashpatheffect without any arguments clears the effect to a full line.

\*\*\*\*\*\*

Образец пути:

Список интервалов должен содержать четное количество записей (≥ 2), причем четные индексы указывают интервалы включения и нечетные индексы указывают интервалы выключения.

Фаза:

Фаза - это смещение в списке интервалов (мод сумма всех интервалов).

Список интервалов контролирует длину тире.

GR.set.stroke контролирует толщину штрихов.

Примечание: этот эффект пути влияет только на рисование с параметром стиля GR.color

устанавливается в 0 (STROKE) или 2 (STROKE\_AND\_FILL). Он игнорируется, если рисование выполнено со стилем == 1 (FILL).

Простой Gr.set.dashpatheffect без каких-либо аргументов очищает эффект до полной строки.

Example:

scale = 2

LIST.CREATE N, pathPatternType1

LIST.ADD pathPatternType1,10\*scale,7\*scale,3\*scale,7\*scale GR.SET.DASHPATHEFFECT pathPatternType1, 5\*scale

##### Gr.path <obj\_nvar>, <list\_pointer\_nvar> {, <x\_nexp>, <y\_nexp>}

Creates a path object defined by a list of strings with the system values "\_MoveTo", "\_LineTo", "\_QuadTo" (Quadratic Bezier Curve), "\_CubicTo" (Cubic Bezier Curve), "\_ArcTo", "\_Close" and "\_End". The path will be located within the bounds of the parameters. The x and y parameters can be set to move the path. The path will or will not be filled depending upon the **Gr.color** style parameter. The <obj\_nvar> returns the Object List object number for this rectangle. This object will not be visible until the **Gr.render** command is called.

The **Gr.modify** parameters for **Gr.path** are: "x" and "y".

\*\*\*\*\*\*

Создает объект пути, определенный списком строк с системными значениями «\_MoveTo», «\_LineTo», «\_QuadTo» (Quadratic Bezier Curve), «\_CubicTo» (Cubic Bezier Curve), «\_ArcTo», «\_Close» и «\_End». ». Путь будет расположен в пределах параметров. Параметры x и y могут быть установлены для перемещения пути. Путь будет или не будет заполнен в зависимости от параметра стиля Gr.color. <Obj\_nvar> возвращает номер объекта списка объектов для этого прямоугольника. Этот объект не будет виден, пока не будет вызвана команда Gr.render.

Параметры Gr.modify для Gr.path: «x» и «y».

Example:

GR.COLOR 255,255,255,255,0

LIST.CREATE S, pathDraft1 R4 = 230

Angle1 =30

Angle2 =45

Angle3 =60

! The following coordinates are relative values according to the optional placing values. LIST.ADD pathDraft1,~

!"\_MoveTo"~ %The first \_MoveTo call is like GR.poly included. STR$(cPx),STR$(cPy+R4)~ %Point 1

!!b1

Set the beginning of the next contour to the point (x1,y1). Moving like a pen in a pen plotter.

Parameters:

x1 The x-coordinate of the start of a new contour y1 The y-coordinate of the start of a newcontour

!!e1

"\_LineTo"~

STR$(cPx-R4\*COS(Angle2\*PI()/180)),STR$(cPy+R4\*SIN(Angle2\*PI()/180))~ %Point 2

!!b2

Add a line from the last point {(x1,y1)} to the specified point (x2,y2). If no \_MoveTo call

has been made for this contour, the first point is automatically set to (0,0). Parameters:

x2 The x-coordinate of the end of aline y2 The y-coordinate of the end of aline

!!e2

"\_LineTo"~

STR$(cPx-R4),STR$(cPy)~ %Point 2

!!b3

Add a quadratic bezier curve from the last point {(x1,y1)}, approaching control point (x2,y2), and ending at (x3,y3). If no \_MoveTo or \_LineTo call has been made for this contour,

the first point is automatically set to (0,0) relativ to the placing coordinates. Parameters:

x2 The x-coordinate of the control point on a quadratic curve y2 The y-coordinate of the control point on a quadratic curve x3 The x-coordinate of the end point on a quadraticcurve

y3 The y-coordinate of the end point on a quadraticcurve

!!e3

"\_QuadTo"~ %Quadratic Bezier Curve

STR$(cPx-R4\*COS(Angle2\*PI()/180)),STR$(cPy-R4\*SIN(Angle2\*PI()/180))~ %Point 2

STR$(cPx),STR$(cPy-R4)~ %Point 3

!!b4

Add a cubic bezier curve from the last point {(x1,y1)}, approaching control points (x2,y2) and (x3,y3), and ending at (x4,y4). If no \_MoveTo or \_LineTo call has been made for this contour, the first

point is automatically set to (0,0) relativ to the placing coordinates. Parameters:

x2 The x-coordinate of the 1st control point on a cubic curve y2 The y-coordinate of the 1st control point on a cubic curve x3 The x-coordinate of the 2nd control point on a cubiccurve y3 The y-coordinate of the 2nd control point on a cubiccurve x4 The x-coordinate of the end point on a cubiccurve

y4 The y-coordinate of the end point on a cubiccurve

!!e4

"\_CubicTo"~ %Cubic Bezier Curve

STR$(cPx+R4\*COS(Angle3\*PI()/180)),STR$(cPy-R4\*SIN(Angle3\*PI()/180))~ %Point2

STR$(cPx+R4\*COS(Angle1\*PI()/180)),STR$(cPy-R4\*SIN(Angle1\*PI()/180))~ %Point3

STR$(cPx+R4),STR$(cPy)~ %Point 4

!!b5

Append the specified arc to the path as a new contour. If the start of the path is different from the path's current last point, then an automatic \_LineTo is added to connect the current contour to the start of the arc. However, if the path is empty, then we call \_MoveTo with the first point of the arc.

Parameters:

Left, Top, Right, Bottom The bounds of oval defining shape and size of the arc. StartAngle Starting angle (in degrees) where the arc begins SweepAngle Sweep angle (in degrees) measuredclockwise(!).

!!e5

"\_ArcTo"~

STR$(cPx),STR$(cPy-R4/4)~ %Left, Top STR$(cPx+R4),STR$(cPy+R4/4)~ %Right, Bottom STR$(0),STR$(180)~ %StartAngle, SweepAngle

!!b6

Close the current contour. If the current point is not equal to the first point of the contour, a line segment is automatically added.

!!e6

!"\_Close"~

!!b7

After \_End the following items are ignored.

!!e7

"\_End"

GR.SET.DASHPATHEFFECT pathPatternType1, 5\*scale

!!b8

GR.PATH <obj\_nvar>, list\_pointer {,x,y} Parameters:

obj\_nvar Objectnumber

list\_pointer List pointer of the list with curve type calls andcoordinates

{,x,y} Placingcoordinates

!!e8

GR.PATH gfirst, pathDraft1, -50, 0 GR.RENDER

GitHub#56 1., 2., 3., 4.

##### Gr.rect <obj\_nvar>, left, top, right, bottom{{, rx}, ry}

Creates a rectangle object. The rectangle will be located within the bounds of the parameters. The rectangle will or will not be filled depending upon the **Gr.color** style parameter. The <obj\_nvar> returns the Display List object number for this rectangle. This object will not be visible until the **Gr.render** command is called.

The **Gr.modify** parameters for **Gr.rect** are: "left", "top", "right", "bottom", "rx" and "ry".

\*\*\*\*\*\*

Создает прямоугольный объект. Прямоугольник будет расположен в пределах параметров. Прямоугольник будет или не будет заполнен в зависимости от параметра стиля Gr.color. <Obj\_nvar> возвращает номер объекта списка отображения для этого прямоугольника. Этот объект не будет виден, пока не будет вызвана команда Gr.render.

Параметры Gr.modify для Gr.rect: "left", "top", "right", "bottom", "rx" и "ry".

GitHub#56 2.

##### Gr.modify <object\_ptr\_nexp> {, <tag\_sexp>, <value\_nexp | value\_sexp>}...

The value of the parameter named <tag\_sexp> in the Display List object

<object\_ptr\_nvar> is changed to the value of the expression <value\_nexp> or

<value\_sexp>. This command can change only one object at a time, but you may list as many tag/value pairs as you want.

With this command, you can change any of the parameters of any object in the Display List. The parameters you can change are given with the descriptions of the commands in this manual. In addition there are two general purpose parameters, "paint" and "alpha" (see below for details).You must provide parameter names that are valid for the specified object.

The results of **Gr.modify** commands will not be observed until a **Gr.render** command executes.

\*\*\*\*\*\*

Значение параметра с именем <tag\_sexp> в объекте списка отображения

<object\_ptr\_nvar> изменяется на значение выражения <value\_nexp> или

<Value\_sexp>. Эта команда может изменять только один объект за раз, но вы можете перечислить столько пар тегов / значений, сколько хотите.

С помощью этой команды вы можете изменить любой из параметров любого объекта в Списке отображения. Параметры, которые вы можете изменить, приведены в описании команд в этом руководстве. Кроме того, есть два параметра общего назначения, «paint» и «alpha» (подробности см. Ниже). Необходимо указать имена параметров, которые действительны для указанного объекта.

Результаты команд Gr.modify не будут наблюдаться, пока команда Gr.render

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | TYPE | POSITION 1  (numeric) | | POSITION 2  (numeric) | | ANGLE/ RADIUS  (numeric) | UNIQ UE  (variou s) | PAIN T  (list ptr) | ALPH A  (num) |
| SHAPES and OBJECTS | [arc](#_bookmark10) | left | top | right | bottom | start\_angle  sweep\_angle | fill\_mo  de | paint | alpha |
| [bitmap](#_bookmark9) | x | y |  |  |  | bitmap | paint | alpha |
| [circle](#_bookmark8) | x | y |  |  | radius |  | paint | alpha |
| draw-able | left | top | right | bottom |  | draw- able | paint | alpha |
| [line](#_bookmark7) | x1 | y1 | x2 | y2 |  |  | paint | alpha |
| [oval](#_bookmark6) | left | top | right | bottom |  |  | paint | alpha |
| [pixels](#_bookmark5) | x | y |  |  |  |  | paint | alpha |
| [point](#_bookmark4) | x | y |  |  |  |  | paint | alpha |
| [path](#_bookmark3) | x | y |  |  |  | list | paint | alpha |
| [poly](#_bookmark2) | x | y |  |  |  | list | paint | alpha |
| [rect](#_bookmark1) | left | top | right | bottom | rx, ry |  | paint | alpha |
| [text](#_bookmark0) | x | y |  |  |  | text | paint | alpha |
| MODI- FIERS | [cli](#_bookmark11)[p](#_bookmark11) | left | top | right | bottom |  | RO | paint | alpha |
| [cli](#_bookmark12)[p](#_bookmark12)out | left | top | right | bottom |  |  | paint | alpha |
| [grou](#_bookmark13)[p](#_bookmark13) |  |  |  |  |  | list | paint | alpha |
| [rotat](#_bookmark14)[e](#_bookmark14) | x | y |  |  | angle |  | paint | alpha |

##### TABLE NOTES:

* The **TYPE** column shows the string returned by **Gr.get.type** for each graphicalobject type.
* **Gr.get.position** returns the values in the **POSITION 1**columns.
* All table entriesare

**Gr.modify** tags (strings). Values of all the tags are numeric except for **"text"**.

* The values of tags in the **UNIQUE** column are either strings (**"text"**) or numbers with special interpretations. **"fill\_mode"** is a logical value. **"list"** is a pointer to a list of point coordinates. **"RO"** is a Region Operator as explained in**Gr.clip**.
* **"alpha"** is an integer value from 0 to 256, with 256 interpreted specially. See**General Purpose Parameters**,below.
* You can modify the **Gr.set.pixels** point-coordinates array directly. There isno

**Gr.modify** tag.

* Modifiers with an underbar in front and uppercase letters like "\_Paint" are also accepted.

\*\*\*\*\*\*

ТАБЛИЦА ПРИМЕЧАНИЯ:

Столбец TYPE показывает строку, возвращаемую Gr.get.type для каждого типа графического объекта.

Gr.get.position возвращает значения в столбцах POSITION 1.

Все записи таблицы

Gr.modify теги (строки). Значения всех тегов являются числовыми, кроме «текст».

Значения тегов в столбце UNIQUE - это либо строки («текст»), либо числа со специальными интерпретациями. "fill\_mode" является логическим значением. «list» - указатель на список координат точек. «RO» является оператором региона, как объяснено в gr.clip.

«альфа» - это целочисленное значение от 0 до 256, причем 256 интерпретируется специально. См. Общие параметры цели, ниже.

Вы можете напрямую изменить массив координат точек Gr.set.pixels. Здесь нет

Gr.modify тег.

Также допускаются модификаторы с подчеркиванием перед и заглавными буквами, такими как \_Paint.

For example, suppose a bitmap object was created with **Gr.bitmap.draw BM\_ptr, galaxy\_ptr, 400, 120**.

Executing **gr.modify BM\_ptr, "x", 420** would move the bitmap from x =400 to x = 420. Executing **gr.modify BM\_ptr, "y", 200** would move the bitmap from y = 120 to y = 200. Executing **gr.modify BM\_ptr, "x", 420, "y", 200** would change both x and y at the same time. Executing **gr.modify BM\_ptr, "bitmap", Saturn\_ptr** would change the bitmap of an image of a (preloaded) Galaxy to the image of a (preloaded) Saturn.

##### Gr.clip <object\_ptr\_nexp>, <left\_nexp>, <top\_nexp>, <right\_nexp>, <bottom\_nexp>{,

**<RO\_nexp>}**

Objects that are drawn after this command is issued will be drawn only within the bounds (clipped) of the clip rectangle specified by the "left, top, right, bottom" numeric expressions. The final parameter is the Region Operator, <RO\_nexp>. The Region Operator prescribes how this clip will interact with everything else you are drawing on the screen or bitmap. If you issue more than one **Gr.clip** command, the RO prescribes the interaction between the current **Gr.clip** rectangle and the previous one. The RO values are:

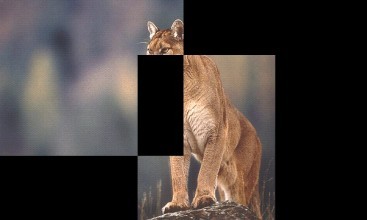
\*\*\*\*\*\*

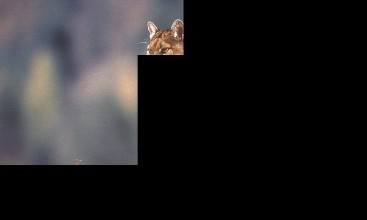
Объекты, которые рисуются после выполнения этой команды, будут отрисовываться только в пределах (обрезанных) прямоугольника клипа, указанного числовыми выражениями «left, top, right, bottom». Последний параметр - это оператор области, <RO\_nexp>. Оператор региона предписывает, как этот клип будет взаимодействовать со всем, что вы рисуете на экране или растровом изображении. Если вы вводите более одной команды Gr.clip, RO предписывает взаимодействие между текущим прямоугольником Gr.clip и предыдущим. Значения RO:

|  |  |
| --- | --- |
| **0** | **Intersect** |
| **1** | **Difference** |
| **~~2~~** | **~~Replace~~** |
| **~~3~~** | **~~Reverse Difference~~** |
| **~~4~~** | **~~Union~~** |
| **~~5~~** | **~~XOR~~** |

The Region Operator parameter is optional. If it is omitted, the default action is **Intersect**. Examples:

##### Original Clip1 Clip 2 Clip 2 applied to Clip 1 with RO parameter on Clip2



##### 0=Intersect 1=Difference 2 =Replace

**3 =ReverseDifference 4=Union 5 =XOR**

**Gr.clip** is a display list object. It can be modified with **Gr.modify**. The modify parameters are "left", "top", "right", "bottom", and "RO".

The **Gr.show** and **Gr.hide** commands can be used with the **Gr.clip** object.

RO values other than INTERSECT and DIFFERENCE have the ability to expand the clip, so values from 2 to 5 are ignored. The graphic mode clipping APIs are intended to only expand the clip as a result of a restore operation. This enables a view parent to clip a canvas to clearly define the maximal drawing area of its children. The recommended alternative calls are Gr.clip with <RO\_nexp> < 2 and Gr.clipOut.

\*\*\*\*\*\*

Значения RO, отличные от INTERSECT и DIFFERENCE, могут расширять клип, поэтому значения от 2 до 5 игнорируются. API-интерфейсы отсечения графического режима предназначены только для расширения клипа в результате операции восстановления. Это позволяет родителю представления обрезать холст, чтобы четко определить максимальную область рисования его дочерних элементов. Рекомендуемые альтернативные вызовы: Gr.clip с <RO\_nexp> <2 и Gr.clipOut.

See also Gr.color with its optional xFermode

##### Gr.clipOut <object\_ptr\_nexp>, <left\_nexp>, <top\_nexp>, <right\_nexp>,

**<bottom\_nexp>**

Sets the clip to the difference of the current clip and the specified rectangle, which is expressed by the "left, top, right, bottom" numeric expressions.

Устанавливает в клипе разницу текущего клипа и указанного прямоугольника, которая выражается числовыми выражениями «слева, сверху, справа, снизу».

### Bitmap CommandsOverview

When a bitmap is created, it is added to a list of bitmaps. Commands that create bitmaps return a pointer to the bitmap. The pointer is an index into the bitmap list. Your program works with the bitmap through the bitmap pointer.

If you want to draw the bitmap on the screen, you must add a graphical object to the Object List. The **Gr.bitmap.draw** command creates a graphical object that holds a pointer to the bitmap. Do not confuse the bitmap with the graphical object. You cannot use the Object Number to access the bitmap, and you cannot use the bitmap pointer to modify the graphical object.

Android devices limit the amount of memory available to your program. Bitmaps may use large blocks of memory, and so may exceed the application memory limit. If a command that creates a bitmap exceeds the limit, the bitmap is not created, and the command returns -1, an invalid bitmap pointer. Your program should test the bitmap pointer to find out if the bitmap was created. If the bitmap pointer is -1, you can call the **GETERROR$()** function to get information about the error.

If a command exceeds the memory limit, but BASIC! does not catch the out-of-memory condition, your program terminates with an error message displayed on the Console screen. If you return the Editor, a line will be highlighted near the one that exceeded the memory limit. It may not be exactly the right line.

Bitmaps use four bytes of memory for each pixel. The amount of memory used depends only on the width and height of the bitmap. The bitmap is not compressed. When you load a bitmap from a file, the file is usually in a compressed format, so the bitmap will usually be larger than the file.

The counting of bitmap pointers was changed, because if it is possible, bitmaps will be overwritten if possible now. If you need automatically a new unique bitmap pointer number the value of the variable have to be 0 at command start. This is important when creating a bitmap for the first time. As a bitmap pointer you can use members of an array also. If a numeric variable or an element of an array has never been used before, the value is 0.

If the value of the bitmap pointer variable is greater than 0 and has been mapped to the internal bitmap list, that bitmap will be overridden.

\*\*\*\*\*\*

Когда растровое изображение создается, оно добавляется в список растровых изображений. Команды, которые создают растровые изображения, возвращают указатель на растровое изображение. Указатель является индексом в списке растровых изображений. Ваша программа работает с растровым изображением через указатель растрового изображения.

Если вы хотите нарисовать растровое изображение на экране, вы должны добавить графический объект в список объектов. Команда Gr.bitmap.draw создает графический объект, который содержит указатель на растровое изображение. Не путайте растровое изображение с графическим объектом. Вы не можете использовать номер объекта для доступа к растровому изображению, и вы не можете использовать указатель растрового изображения для изменения графического объекта.

Устройства Android ограничивают объем памяти, доступный вашей программе. Растровые изображения могут использовать большие блоки памяти и, следовательно, могут превышать лимит памяти приложения. Если команда, которая создает растровое изображение, превышает лимит, растровое изображение не создается, и команда возвращает -1, недопустимый указатель растрового изображения. Ваша программа должна проверить указатель растрового изображения, чтобы выяснить, было ли оно создано. Если указатель растрового изображения равен -1, вы можете вызвать функцию GETERROR $ (), чтобы получить информацию об ошибке.

Если команда превышает предел памяти, но BASIC! не обнаруживает состояние нехватки памяти, ваша программа завершает работу с сообщением об ошибке, отображаемым на экране консоли. Если вы вернете редактор, строка будет выделена рядом с той, которая превысила предел памяти. Это может быть не совсем правильная линия.

Растровые изображения используют четыре байта памяти для каждого пикселя. Объем используемой памяти зависит только от ширины и высоты растрового изображения. Растровое изображение не сжато. Когда вы загружаете растровое изображение из файла, файл обычно находится в сжатом формате, поэтому растровое изображение обычно будет больше, чем файл.

Счетчик растровых указателей был изменен, потому что, если это возможно, растровые изображения будут перезаписаны, если это возможно, сейчас. Если вам нужен автоматически новый уникальный номер указателя растрового изображения, значение переменной должно быть 0 при запуске команды. Это важно при создании растрового изображения в первый раз. В качестве растрового указателя вы также можете использовать члены массива. Если числовая переменная или элемент массива никогда не использовались ранее, значение равно 0.

Если значение переменной указателя растрового изображения больше 0 и сопоставлено с внутренним списком растровых изображений, это растровое изображение будет переопределено

##### Gr.bitmap.load <bitmap\_ptr\_nvar>, <file\_name\_sexp>{{{{{{{, <wB\_nexp>},

**<hB\_nexp>}, <cropX\_nexp>}, <cropY\_nexp>}, <cropW\_nexp>}, <cropH\_nexp>},**

**<bgColor\_sexp>}**

Creates a bitmap from the file specified in the file\_name string expression. Returns a pointer to the created bitmap for use with other **Gr.bitmap** commands. If no bitmap is created, the returned bitmap pointer is -1. Call **GETERROR$()** for information about the failure. Some of the possible causes are:

* + The file or resource does not exist.
  + There is not enough memory available to create thebitmap.

Bitmap image files are assumed to be located as non source and non database files in the "<pref base drive>/rfo-basic/data/" directory by default.

~~Note: You may include path fields in the file name. For example, "../../Cougar.jpg" wouldcause BASIC! to look for Cougar.jpg in the top level directory of the base drive, usually theSD card. "images/Kitty.png" would cause BASIC! to look in the images(d) sub-directory ofthe "/sdcard/rfo-basic/data/" ("/sdcard/rfo-basic/data/images/Kitty.png").~~

Note: Bitmaps loaded with this command cannot be changed with the**Gr.bitmap.drawinto** command. To draw into an image loaded from a file, first create an empty bitmap then draw the loaded bitmap into the emptybitmap.

If a SVG (Scalable Vector Graphic) file is chosen the result based on the measurement in relation to 96 DPI. I.e. 5 cm /(2.54 cm/Inch) \* 96 (Dots/Inch) = 189 Dots

Rendering the SVG file the Android "\_Sans\_Serif" font is the default one. Special fonts like Arial, Verdana etc. are not supported.

The next arguments are optional.

The border arguments <wB\_nexp> and <hB\_nexp> specify the borders of the result. The bitmap is scaled inside these borders, so that a square is a square and not a rectangle.

\*\*\*\*\*\*

Создает растровое изображение из файла, указанного в строковом выражении file\_name. Возвращает указатель на созданное растровое изображение для использования с другими командами Gr.bitmap. Если растровое изображение не создано, возвращаемый указатель растрового изображения равен -1. Вызовите GETERROR $ () для получения информации о сбое. Некоторые из возможных причин:

Файл или ресурс не существует.

Недостаточно памяти для создания карты битов.

Предполагается, что файлы растровых изображений по умолчанию находятся в файлах, не являющихся источником и файлами базы данных, в каталоге «<pref base drive> / rfo-basic / data /».

Примечание. Вы можете включить поля пути в имя файла. Например, "../../Cougar.jpg" может вызвать бейсик! искать Cougar.jpg в каталоге верхнего уровня базового диска, обычно SD-карты. "images / Kitty.png" вызовет бейсик! искать в подкаталоге images (d) каталога "/ sdcard / rfo-basic / data /" ("/sdcard/rfo-basic/data/images/Kitty.png").

Примечание. Растровые изображения, загруженные с помощью этой команды, не могут быть изменены с помощью команды G.r.mapmap.drawinto. Чтобы нарисовать изображение, загруженное из файла, сначала создайте пустое растровое изображение, затем нарисуйте загруженное растровое изображение в пустое растровое изображение.

Если выбран файл SVG (масштабируемая векторная графика), результат будет основан на измерении в отношении 96 DPI. То есть 5 см / (2,54 см / дюйм) \* 96 (точек / дюйм) = 189 точек

Рендеринг SVG-файла - это шрифт Android "\_Sans\_Serif" по умолчанию. Специальные шрифты, такие как Arial, Verdana и т. Д. Не поддерживаются.

Следующие аргументы являются необязательными.

Аргументы границ <wB\_nexp> и <hB\_nexp> определяют границы результата. Растровое изображение масштабируется внутри этих границ, так что квадрат - это квадрат, а не прямоугольник.

|  |  |  |
| --- | --- | --- |
| <wB\_nexp> | <hB\_nexp> | Result |
| 0 (default) | 0 (default) | In the original resolution |
| > 0 | 0 | Scaled to the given width |
| 0 | > 0 | Scaled to the given height |

##### Note that SVG files need to be converted large enough for the best quality.

The arguments <cropX\_nexp>, <cropY\_nexp>, <cropW\_nexp>, <cropH\_nexp> are describing the position of the left (cropX) top (cropY) corner and the size (cropW, cropH) of the part to cut out. The defaults are 0,0,-1,-1. If cropW is -1 the right edge is limiting the cut out. If cropH is -1 the bottom edge is limiting the cut out. So the defaults return the fullsize. To prevent memory faults it is a good idea to crop at loading. Because older Android versions limit the maximum images size for loading to ≤ 2048 x ≤ 2048 pixel ≤ 12 MB RAM.

See also the <file\_name\_sexp> extension of Gr.bitmap.size about this behavior.

Mainly for buttons in conjunction with icons <bgColor\_sexp> sets the background color by the Gr.Paint color notation.

\*\*\*\*\*\*

Обратите внимание, что файлы SVG должны быть конвертированы достаточно большими для лучшего качества.

Аргументы <cropX\_nexp>, <cropY\_nexp>, <cropW\_nexp>, <cropH\_nexp> описывают положение левого верхнего (cropX) верхнего угла (cropY) и размер (cropW, cropH) части, которую нужно вырезать. Значения по умолчанию: 0,0, -1, -1. Если обрезание W равно -1, правый край ограничивает вырез. Если обрезка H равна -1, нижний край ограничивает вырез. Таким образом, значения по умолчанию возвращают полный размер. Во избежание ошибок памяти рекомендуется обрезать при загрузке. Поскольку более старые версии Android ограничивают максимальный размер изображений для загрузки до ≤ 2048 x ≤ 2048 пикселей ≤ 12 МБ ОЗУ.

См. Также расширение <file\_name\_sexp> Gr.bitmap.size об этом поведении.

В основном для кнопок в сочетании с иконками <bgColor\_sexp> устанавливает цвет фона с помощью цветовой нотации Gr.Paint.

Example:

Gr.bitmap.size "blocks.svg", width, height

! Crop the right bottom quarter of "blocks.svg" in its original resolution + a new blue backgr. Gr.bitmap.load bPtr, "blocks.svg",0,0, width/2, height/2, -1, -1, "\_10,Blue"

! Try also "\_Black" and"\_Red"

Seealso Gr.bitmap.crop

##### Gr.bitmap.size <bitmap\_ptr\_nexp>|<file\_name\_sexp>, width, height

Return the pixel width and height of the bitmap pointed to by <bitmap\_ptr\_nexp> or

<file\_name\_sexp> into the width and height variables. SVG (Scalable Vector Graphic) files ending with \*.svg are also supported. In this case the measurement in relation to 96 DPI.

I.e. 5 cm /(2.54 cm/Inch) \* 96 (Dots/Inch) = 189 Dots

The advantage of <file\_name\_sexp> is the detection of the size before loading. This prevents memory faults mostly on older devices.

\*\*\*\*\*\*

Вернуть ширину и высоту пикселя растрового изображения, на которое указывает <bitmap\_ptr\_nexp> или

<file\_name\_sexp> в переменные ширины и высоты. Также поддерживаются файлы SVG (масштабируемая векторная графика), заканчивающиеся \* .svg. В этом случае измерение по отношению к 96 DPI.

То есть 5 см / (2,54 см / дюйм) \* 96 (точек / дюйм) = 189 точек

Преимущество <file\_name\_sexp> заключается в определении размера перед загрузкой. Это предотвращает сбои памяти в основном на старых устройствах.

##### Gr.bitmap.clr <bitmap\_ptr\_nexp> {, <paint\_nexp>}

Fills a bitmap completely with transparency without destroying it. Anything already on the bitmap is cleared. There is no alpha blending. Optional color from a paint.

This command is approximately ten times faster than the combination Gr.bitmap.delete with Gr.bitmap.create.

\*\*\*\*\*\*

Полностью заполняет растровое изображение прозрачностью, не разрушая его. Все, что уже есть на растровом изображении, очищается. Там нет альфа-смешивания. Дополнительный цвет от краски.

Эта команда примерно в десять раз быстрее, чем комбинация Gr.bitmap.delete с Gr.bitmap.create.

##### Gr.bitmap.save <bitmap\_ptr\_nvar>, <filename\_sexp>{, <quality\_nexp>}

Saves the specified bitmap to a file. The default path is "<pref base drive>/rfo-basic/data/". The file will be saved as a JPEG file if the filename ends in ".jpg".

The file will be saved as a PNG file if the filename ends in anything else (including ".png"). To <quality\_nexp>, the possible range is from 0 to 100. Default is 50.

\*\*\*\*\*\*

Сохраняет указанное растровое изображение в файл. Путь по умолчанию: «<pref base drive> / rfo-basic / data /». Файл будет сохранен как файл JPEG, если имя файла заканчивается на «.jpg».

Файл будет сохранен как файл PNG, если имя файла заканчивается чем-либо еще (включая «.png»). Для <quality\_nexp> возможный диапазон от 0 до 100. По умолчанию 50.

##### Gr.bitmap.scale <new\_bitmap\_ptr\_nvar>, <bitmap\_ptr\_nexp>, width, height {,

**<smoothing\_lexp>}**

Scales a previously loaded bitmap (<bitmap\_ptr\_nexp>) to the specified width and height and creates a new bitmap <new\_bitmap\_ptr\_nvar>. The old bitmap still exists, it is not deleted. If there is not enough memory available to create the new bitmap, the returned bitmap pointer is -1. Call **GETERROR$()** for information about the failure.

Negative values for width and height will cause the image to be flipped left to right or upside down.

Neither the width value nor the height value may be zero.

Use the optional smoothing logical expression (<smoothing\_lexp>) to request that the scaled image not be smoothed. If the expression is false (zero) then the image will not be smoothed. If the optional parameter is true (not zero) or not specified then the image will be smoothed.

\*\*\*\*\*\*

Масштабирует ранее загруженное растровое изображение (<bitmap\_ptr\_nexp>) до указанной ширины и высоты и создает новое растровое изображение <new\_bitmap\_ptr\_nvar>. Старый точечный рисунок все еще существует, он не удаляется. Если для создания нового растрового изображения недостаточно памяти, возвращаемый указатель растрового изображения равен -1. Вызовите GETERROR $ () для получения информации о сбое.

Отрицательные значения ширины и высоты приведут к переворачиванию изображения слева направо или вверх ногами.

Ни значение ширины, ни значение высоты не могут быть равны нулю.

Используйте необязательное логическое выражение сглаживания (<smoothing\_lexp>), чтобы запросить сглаживание масштабированного изображения. Если выражение ложно (ноль), изображение не будет сглажено. Если необязательный параметр имеет значение true (не ноль) или не указан, изображение будет сглажено.

##### GR.bitmap.filter <new\_bitmap\_ptr\_nvar>, <bitmap\_ptr\_nexp>, <bundl\_ptr\_nexp>

Processes by filters a previously loaded bitmap (<bitmap\_ptr\_nexp>) specified by the

<bundl\_ptr\_nexp> bundle and creates a new bitmap <new\_bitmap\_ptr\_nvar>. If the old bitmap still exists; it is not deleted.

\*\*\*\*\*\*

Обрабатывает фильтрами ранее загруженное растровое изображение (<bitmap\_ptr\_nexp>), указанное

<bundl\_ptr\_nexp> связывает и создает новое растровое изображение <new\_bitmap\_ptr\_nvar>. Если старое растровое изображение все еще существует; это не удалено.

|  |  |  |
| --- | --- | --- |
| **Table of Bundle Keys** | | |
| **Key** | **Value** | **Description** |
| **\_Filter** | \_Binary  \_BlackFilter  \_Blur  \_Brightness  \_ColorRotation  \_ColorScale  ~~\_EdgeDetection~~  \_Engrave  \_Flip  \_GammaCorrection  \_Hue  \_Invert  \_MaskPoly  \_ModifyOrientation  \_PolyToPoly  \_Rotate  \_RoundCorners  \_Saturation  \_Skew  \_Smooth  \_SnowEffect  \_Sharpen  \_UseColorMatrix  \_UseConvolutionMatrix  \_Watermark (String) | Set the filter type.  **Always needed!** |
| **\_Binary**  The result is a Bitmap with only white and black pixels. | | |
|  |  | Default values. |
| **\_BlackFilter**  Base on randomizing image pixels, enhance the noise of darkness.  The algorithm is to generate a threshold number (0-255), if all R,G,B values of a pixel are less than the threshold, then set the pixel to black. | | |
|  |  | Default values.  It is a pixel by pixel function, so it needs more time. |
| **\_Blur**  Also called Gaussian Blur Effect | | |
| **\_Radius** | 0.1 to 25 (numeric) | Default is 15. |
| **\_Brightness** see \_ContrastBrightness | | |

|  |  |  |
| --- | --- | --- |
| **\_Value** | From -255 to +255 (numeric) | Default is 0.  The result is limited to the possible minimum or maximum. |
| **\_ColorRotation**  Set the rotation on a color axis by the specifiedvalues. | | |
| **\_Axis** | \_Red, \_Green or \_Blue  (String) | Default is \_Red. |
| **\_Degrees** | 0 to 360  (numeric) | Default is 0. |
| **\_ColorScale** | | |
| **\_AlphaScale** | 0 and positive numbers (numeric) | A value of 0 switches the color to off. 1 is identity and default.  The result is limited to the possible maximum. |
| **\_RedScale** | 0 and positive numbers (numeric) | A value of 0 switches the color to off. 1 is identity and default.  The result is limited to the possible maximum. |
| **\_GreenScale** | 0 and positive numbers (numeric) | A value of 0 switches the color to off. 1 is identity and default.  The result is limited to the possible maximum. |
| **\_BlueScale** | 0 and positive numbers (numeric) | A value of 0 switches the color to off. 1 is identity and default.  The result is limited to the possible maximum. |
| **\_Contrast** | | |
| **\_Value** | 0 and positive numbers (numeric) | A value of 0 sets the contrast to minimal. 1 is identity and default. The result is limited to the possible  maximum. |
| **\_ContrastBrightness**  Contrast is the difference in luminance and/or color that makes an object (or its representation in an image or display) distinguishable.  In visual perception of the real world, contrast is determined by the difference in the color and brightness of the object and other objects within the same field of view. The concept of brightness is rather simple, increasing/decreasing value of each R, G, B channel together.  + By increasing: image results brighter.  - By decreasing: image results darker. | | |
| **\_BrightnessValue** | From -255 to +255 (numeric) | Default is 0.  The result is limited to the possible minimum or maximum. |
| **\_ContrastValue** | 0 and positive numbers (numeric) | A value of 0 sets the contrast to minimal. 1 is identity and default.  The result is limited to the possible maximum. |
| **\_EdgeDetection** | | |
| **\_Level** | \_Low, \_Medium or \_High (String) | Default is \_Medium.  The speed could be faster, but the results of the faster Android Render  Script are strange. |

|  |  |  |
| --- | --- | --- |
| **\_Engrave** | | |
|  |  | Default values. |
| **\_Hue**  Translates the colors within the color wheel by the specified angle. White, gray and black are not affected. | | |
| **\_Degrees** | 0 to 360 (numeric) | Default is 0. |
| **\_Flip** | | |
| **\_Horizontal** | 0 or 1 (numeric) | Default is 0. |
| **\_Vertical** | 0 or 1 (numeric) | Default is 0. |
| **\_GammaCorrection** | | |
| **\_Gamma** | 0 to 8 (numeric) | 1 is identity and default. Correction by a 1 / gamma value. Values below 1 provide darker results. Values greater than 1 provide brighter results.  It is a pixel by pixel function, so it needs more time. |
| **\_Invert**  Inverts the bitmap from a positive into a negative or a negative into a positive one. | | |
|  |  | Default values. |
| **\_MaskPoly**  Masks the bitmap with a polygon. | | |
| **\_SourcePoints** | Array (numeric) | Default is {0,0,0,0,0,0,0,0}  Array of polygon nodes positions as x,y pairs. You can use polygons  with three or more nodes. |
| **\_Type** | \_In or \_Out (String) | Default is \_In.  If the \_Type is \_In the area inside the polygon is returned as **transparent**. The argument \_Out  stands for the area outside. |
| **\_ModifyOrientation**  Modifies the orientation of the current bitmap specified by the EXIF of a bitmap given by an URL. | | |
| **\_ImageUrl** | (String) |  |
| **\_PolyToPoly**  Crops from a source bitmap a part defined by a polygon with threeor fournodes. The result is described by a destination polygon with the same number of nodes.  The order of the nodes are beginning at the left top corner in a clockwise turned direction. Sometimes we get **in trouble,** if we use a polygon with **four** nodes and a small source  bitmap. In this case try to scale the source bitmap to double size or more. | | |

|  |  |  |
| --- | --- | --- |
| **\_SourcePoints** | Array (numeric) | Default is {0,0,0,0,0,0,0,0}  Array of polygon nodes positions as x,y pairs. You can use polygons  with three or four nodes. |
| **\_DestinationPoints** | Array (numeric) | Default is {0,0,0,0,0,0,0,0}  Array of polygon nodes positions as x,y pairs. You can use polygons  with three or four nodes. |
| **\_Mask** | 0 or 1 (numeric) | Default = 1  Masks the source bitmap by the  \_SourcePoints to get a transparent area outside the polygon. |
| **\_Debug** | 0 or 1  (numeric) | Debug results will be printed at the  console. |
| **\_Rotate** | | |
| **\_Degrees** | 0 to 360 (numeric) | Default is 0.  Keep in mind, that an angle <> 0, 180 perhaps also 90 and 270 (width equals high) will make the bitmap larger.  Areas of the resulting four triangles are transparent. |
| **\_RoundCorners** | | |
| **\_Radius** | 0 and positive numbers (numeric) | Default is 0. |
| **\_Saturation** | | |
| **\_Value** | 0 and positive numbers (numeric) | A value of 0 maps the color to **gray- scale**. 1 is identity and default.  Values > 1 **boost** the colors.  The result is limited to the possible maximum. |
| **\_Sharpen** | | |
| **\_Level** | \_Low, \_Medium or \_High  (String) | Default is \_Medium. |
| **\_Skew** | | |
| **\_DeltaX** | 0 and positive numbers  (numeric) | Default is 0.  Sets the x size of the skew. |
| **\_DeltaY** | 0 and positive numbers  (numeric) | Default is 0.  Sets the y size of the skew. |
| **\_AtX** | 0 and positive numbers  (numeric) | Default is 0.  Sets the x position of the skew. |
| **\_AtY** | 0 and positive numbers  (numeric) | Default is 0.  Sets the y position of the skew. |
| **\_Smooth** | | |
| **\_Value** |  | Default is 0. |
| **\_SnowEffect**  Opposite of Black Filter  Sets all pixels having R,G,B values to the max of 255 when they are greater than threshold. | | |

|  |  |  |
| --- | --- | --- |
|  |  | Default values.  It is a pixel by pixel function, so it needs more time. |
| **\_UseColorMatrix** | | |
| **\_ColorMatrix** | Example: Array.Load colorMatrix[],~ %[  -1, 0, 0, 0,255,~  0, -1, 0, 0,255,~  0, 0, -1, 0,255,~  0, 0, 0,1,0 %]  (numeric array) |  |
| **\_UseConvolutionMatrix** | | |
| **\_ConvolutionMatrix** | Example:  Array.Load covolutionMatrix[],~ %[  -0.15, -0.15,-0.15,~  -0.15, 2.2,-0.15,~  -0.15, -0.15, -0.15 % ]  (numeric array) |  |
|  |  |  |
| **\_Watermark**  Gives your image a watermark. | | |
| **\_Text** | (String) | Default is "".  Text of the watermark. |
| **\_Color** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Default is "". |
| **\_AtX** | 0 and positive numbers (numeric) | Default is 0.  Sets the x position of the watermark. |
| **\_AtY** | 0 and positive numbers (numeric) | Default is 0.  Sets the y position of the watermark. |
| **\_Size** | 0 and positive numbers (numeric) | Default is 10.  Font size of the watermark. |
| **\_Underline** | 0 or 1 (numeric) | Default is 0.  If > 0 the watermark has a underline. |

Example 1:

GR.OPEN "\_White", 1, 1

GR.BITMAP.LOAD bPtr1, "cartman.png" BUNDLE.PUT bndPtr1, "\_Filter", "\_Hue" counter = 0

DO

BUNDLE.PUT bndPtr1, "\_Degrees", counter GR.BITMAP.SCALE bPtr2, bPtr1, 600,600

GR.BITMAP.FILTER bPtr3, bPtr2, bndPtr1 GR.BITMAP.DRAW oPtr1, bPtr3, 250, 250 GR.RENDER

counter = counter +5 UNTIL counter = 365

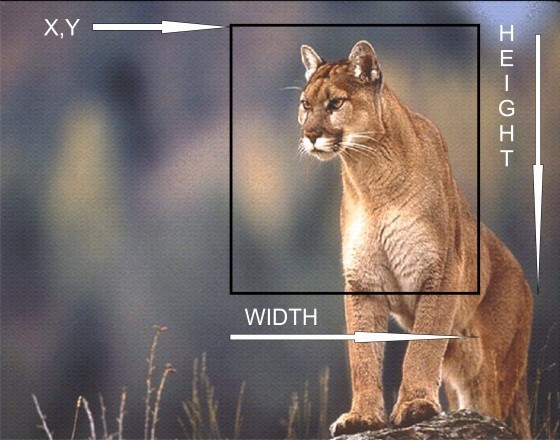
##### Gr.bitmap.crop <new\_bitmap\_ptr\_nvar>, <source\_bitmap\_ptr\_nexp>, <x\_nexp>,

**<y\_nexp>, <width\_nexp>, <height\_nexp>**

Creates a cropped copy of an existing source bitmap specifiedby

<source\_bitmap\_ptr\_nexp>. The source bitmap is unaffected; a rectangular section is copiedintoanewbitmap.Apointertothenewbitmapisreturnedin<new\_bitmap\_nvar>. If there is not enough memory available to create the new bitmap, the returned bitmap pointer is -1. Call **GETERROR$()** for information about thefailure.

The <x\_nexp>, <y\_nexp> pair specifies the point within the source bitmap that the crop is to start at. The <width\_nexp>, <height\_nexp> pair defines the size of the rectangular region to crop.



\*\*\*\*\*\*

Создает обрезанную копию существующего исходного растрового изображения, указанного

<Source\_bitmap\_ptr\_nexp>. Исходное растровое изображение не изменяется; прямоугольный раздел копируется в новую карту битов. Указатель на новую карту битов возвращается в <new\_bitmap\_nvar>. Если для создания нового растрового изображения недостаточно памяти, возвращаемый указатель растрового изображения равен -1. Вызовите GETERROR $ () для получения информации о сбое.

Пара <x\_nexp>, <y\_nexp> указывает точку в исходном растровом изображении, с которой должна начинаться обрезка. Пара <width\_nexp>, <height\_nexp> определяет размер прямоугольной области для обрезки.

To check the bounds, use as an example: GR.BITMAP.SIZE bp, wx, hy % 48, 48 x =10

y =20

width = 10

height = 15

IF x < 0 | (x + width) > wx | width < 1 | y < 0 | (y + height) > hy | height < 1 PRINT "Error: The crop frame is out of bounds"

END END IF

GR.BITMAP.CROP newBp, bp, x, y

##### Gr.bitmap.drawinto.end {<legacy\_mode\_nexp>}

End the draw-into-bitmap mode.Subsequent draw commands will place the objects into the display list for rendering on the screen. If you wish to display the drawn-into bitmap on the screen, issue a **Bitmap.draw** command for that bitmap.

For legacy reasons <legacy\_mode\_nexp> suppresses a runtime error, if Gr.bitmap.drawinto.start is not executed before.

\*\*\*\*\*\*

Завершите режим рисования в растровое изображение. Последующие команды рисования поместят объекты в список отображения для визуализации на экране. Если вы хотите отобразить растровое растровое изображение на экране, введите команду Bitmap.draw для этого растрового изображения.

По устаревшим причинам <legacy\_mode\_nexp> подавляет ошибку времени выполнения, если Gr.bitmap.drawinto.start ранее не выполнялся

##### Gr.bitmap.get.histogram <bitmap\_ptr\_nexp>, alpha[], red[], green[], blue[]

The bitmap pointer is specified by <bitmap\_ptr\_nexp>. The command returns the color channel specific histogram arrays alpha[], red[], green[] and blue[]. Each array contains the sum of pixels with the same intensity. If 47 pixel have the red intensity of 233, red[234] returns 47. 234 because BASIC! arrays are starting with an index of 1 instead of 0.

\*\*\*\*\*\*

Указатель растрового изображения указывается <bitmap\_ptr\_nexp>. Команда возвращает специфичные для цветового канала массивы гистограммы: альфа [], красный [], зеленый [] и синий []. Каждый массив содержит сумму пикселей с одинаковой интенсивностью. Если 47 пикселей имеют интенсивность красного 233, красный [234] возвращает 47. 234, потому что BASIC! массивы начинаются с индекса 1 вместо 0.

##### Gr.bitmap.get.pixarr <bitmap\_ptr\_nexp>, alpha[], red[], green[], blue[]

The bitmap pointer is specified by <bitmap\_ptr\_nexp>. The command returns the color channel specific pixel arrays alpha[], red[], green[] and blue[]. The order is column by column.

\*\*\*\*\*\*

Указатель растрового изображения указывается <bitmap\_ptr\_nexp>. Команда возвращает определенные пиксельные массивы цветового канала: альфа [], красный [], зеленый [] и синий []. Порядок столбец за столбцом.

Example of a bitmap with 5 pixels width and 2 pixels height:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **y\x** | **1** | **2** | **3** | **4** | **5** |
| **1** | *0* | *200* | *40* | *60* | *80* |
| **2** | *100* | *30* | *50* | *70* | *90* |

[3,2] = 50

Example:

GR.BITMAP.LOAD bPtr, "cartman.png" GR.BITMAP.GET.PIXARR bPtr, alpha[], red[], green[], blue[] ARRAY.DIMS alpha[], dims[]

DEBUG.ON DEBUG.DUMP.ARRAY dims[]

See also GR.bitmap.filter with the key \_UseColorMatrix

##### Gr.bitmap.set.pixarr <bitmap\_ptr\_nvar>, alpha[]{, red[], green[], blue[]}

Creates a bitmap by the bitmap pointer <bitmap\_ptr\_nvar> and sets the pixels by the color channel specific pixel arrays. The order is column by column specified by alpha[]. An alpha[] array dimension-ed by [5,2] returns a bitmap with 5 pixels width and 2 pixels height. The defaults of the arrays red[], green[], blue[] are filled with 255. Values < 0 are changed to 0 and values > 255 to 255.

\*\*\*\*\*\*

Создает растровое изображение с помощью указателя растрового изображения <bitmap\_ptr\_nvar> и устанавливает пиксели с помощью определенных пиксельных массивов цветовых каналов. Порядок столбец за столбцом, указанный в альфа []. Размер массива альфа [], отредактированный [5,2], возвращает растровое изображение с шириной 5 пикселей и высотой 2 пикселя. По умолчанию массивы red [], green [], blue [] заполнены 255. Значения <0 изменяются на 0, а значения> 255 на 255.

Example of a bitmap with 5 pixels width and 2 pixels height:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **y\x** | **1** | **2** | **3** | **4** | **5** |
| **1** | *0* | *200* | *40* | *60* | *80* |
| **2** | *100* | *30* | *50* | *70* | *90* |

[3,2] = 50

Example:

! Bitmap 100\*40 with random coloredpixels b = 100 : h = 40 : bS = b \*h

DIM bSArray[bS] ARRAY.FILL bSArray[],255

ARRAY.LOAD d[], b, h % 100\*40 = 4000

ARRAY.TO.DIMS bSArray[], d[], alpha[] % Only alpha[] dims need to be specified ARRAY.RND red[], bS, 0, 255

ARRAY.RND green[], bS, 0, 255

ARRAY.RND blue[], bS, 0, 255

GR.BITMAP.SET.PIXARR nRndPtr, alpha[] , red[], green[], blue[] GR.BITMAP.DRAW rndPtr, nRndPtr, 200, 600

GR.RENDER

See also GR.bitmap.filter with the key \_UseColorMatrix

### Drawable Commands

**Overview**

Beginning with Android **9.0** / Pie API **28** animated drawables are supported.

Unfortunately was the file access to all types of animated drawables not be back-ported for earlier versions until now.

Drawable is a parent type for drawing and animating graphics. Child types are bitmap- or vector graphics are animated or fixed.

Supported image file types are BMP, PNG, JPEG, WEBP, GIF or HEIF.

If the encoded image is an animated GIF or WEBP, the animation has to be started by GR.drawable.start.

Until including Android **8.1** / Oreo API **27** the image file types BMP, PNG, JPEG and GIF are supported.

But it is not possible to convert a simple bitmap to an animated GIF.

When a drawable is created, it is added to a list of drawables. Commands that create drawables return a pointer to the drawable. The pointer is an index into the drawable list. Your program works with the drawable through the drawable pointer.

If you want to draw the drawable on the screen, you must add a graphical object to the Object List. The **Gr.drawable.draw** command creates a graphical object that holds a pointer to the drawable. Do not confuse the drawable with the graphical object. You cannot use the Object Number to access the drawable, and you cannot use the drawable pointer to modify the graphical object.

Android devices limit the amount of memory available to your program. drawables may use large blocks of memory, and so may exceed the application memory limit. If a command that creates a drawable exceeds the limit, the drawable is not created, and the command returns -1, an invalid drawable pointer. Your program should test the drawable pointer to find out if the drawable was created. If the drawable pointer is -1, you can call the **GETERROR$()** function to get information about the error.

If a command exceeds the memory limit, but if BASIC! does not catch the out-of-memory condition, your program terminates with an error message displayed on the Console screen. If you return the Editor, a line will be highlighted near the one that exceeded the memory limit. It may not be exactly the right line.

Drawables can use up to four bytes of memory for each pixel. The amount of memory used depends mainly on the width and height of the drawable. The drawable is not compressed. When you load a drawable from a file, the file is usually in a compressed format, so the drawable will usually be larger than the file.

The counting of drawable pointers was changed, because if it is possible, drawables will be overwritten if possible now. If you need automatically a new unique drawable pointer number the value of the variable have to be 0 at command start. This is important when creating a drawable for the first time. As a drawable pointer you can use members of an array also. If a numeric variable or an element of an array has never been used before,the value is0.

If the value of the drawable pointer variable is greater than 0 and has been mapped to the internal drawable list, that drawable will be overridden.

\*\*\*\*\*\*

Начиная с Android 9.0 / Pie API, поддерживается 28 анимационных рисунков.

К сожалению, доступ к файлам для всех типов анимированных рисунков не был перенесен в более ранние версии до сих пор.

Drawable - это родительский тип для рисования и анимации графики. Дочерние типы - это растровая или векторная графика, анимированная или фиксированная.

Поддерживаемые типы файлов изображений: BMP, PNG, JPEG, WEBP, GIF или HEIF.

Если закодированное изображение представляет собой анимированный GIF или WEBP, анимация должна быть запущена GR.drawable.start.

До включения Android 8.1 / Oreo API 27 поддерживаются типы файлов изображений BMP, PNG, JPEG и GIF.

Но невозможно преобразовать простое растровое изображение в анимированный GIF.

Когда объект рисования создается, он добавляется в список объектов. Команды, которые создают Drawables, возвращают указатель на Drawable. Указатель является указателем в списке для рисования. Ваша программа работает с drawable через drawable указатель.

Если вы хотите нарисовать нарисованное на экране, вы должны добавить графический объект в список объектов. Команда Gr.drawable.draw создает графический объект, который содержит указатель на объект рисования. Не путайте прорисовку с графическим объектом. Вы не можете использовать номер объекта для доступа к объекту рисования, и вы не можете использовать указатель чертежа для изменения графического объекта.

Устройства Android ограничивают объем памяти, доступный вашей программе. Drawables может использовать большие блоки памяти, и поэтому может превышать предел памяти приложения. Если команда, создающая объект рисования, превышает лимит, объект рисования не создается, и команда возвращает -1, недопустимый указатель чертежа. Ваша программа должна проверить указатель для рисования, чтобы выяснить, был ли создан рисованный объект. Если рисуемый указатель равен -1, вы можете вызвать функцию GETERROR $ (), чтобы получить информацию об ошибке.

Если команда превышает предел памяти, но если BASIC! не обнаруживает состояние нехватки памяти, ваша программа завершает работу с сообщением об ошибке, отображаемым на экране консоли. Если вы вернете редактор, строка будет выделена рядом с той, которая превысила предел памяти. Это может быть не совсем правильная линия.

Drawables может использовать до четырех байтов памяти для каждого пикселя. Объем используемой памяти зависит в основном от ширины и высоты вытягиваемого материала. Drawable не сжимается. Когда вы загружаете файл из файла, файл обычно находится в сжатом формате, поэтому размер файла обычно больше файла.

Подсчет указателей для рисования был изменен, потому что, если это возможно, рисованные объекты будут перезаписаны, если это возможно, сейчас. Если вам нужен автоматически новый уникальный номер рисуемого указателя, значение переменной должно быть 0 при запуске команды. Это важно при создании рисования в первый раз. В качестве нарисованного указателя вы также можете использовать элементы массива. Если числовая переменная или элемент массива никогда не использовались ранее, значение равно 0.

Если значение переменной указателя рисования больше 0 и сопоставлено с внутренним списком рисования, то это рисование будет переопределено.

##### Gr.drawable.load <drawable\_ptr\_nvar>, <file\_name\_sexp>

Creates a drawable from the file specified in the file\_name string expression. Returns a pointer to the created drawable for use with other **Gr.drawable** commands. If no drawable is created, the returned drawable pointer is -1. Call **GETERROR$()** for information about the failure. Some of the possible causes are:

* + The file or resource does not exist.
  + There is not enough memory available to create thedrawable.

drawable image files are assumed to be located in the "<pref base drive>/rfo-basic/data/" directory.

Note: Drawables loaded with this command cannot be changed directly. To draw into an image loaded from a file, first create an empty bitmap then draw the loaded drawable into the empty bitmap by Gr.bitmap.drawinto.start.

\*\*\*\*\*\*

Создает отрисовку из файла, указанного в строковом выражении file\_name. Возвращает указатель на созданный чертеж для использования с другими командами Gr.drawable. Если ничья не создана, возвращаемый указатель рисования равен -1. Вызовите GETERROR $ () для получения информации о сбое. Некоторые из возможных причин:

Файл или ресурс не существует.

Недостаточно памяти для создания разборного.

Предполагается, что файлы для рисования изображений находятся в каталоге «<pref base drive> / rfo-basic / data /».

Примечание. Drawables, загруженные с помощью этой команды, не могут быть изменены напрямую. Чтобы нарисовать изображение, загруженное из файла, сначала создайте пустое растровое изображение, а затем нарисуйте загруженный чертеж в пустое растровое изображение с помощью Gr.bitmap.drawinto.start.

See alsoGr.bitmap.load

##### Gr.drawable.fromBitmap <drawable\_ptr\_nvar>,<bitmap\_ptr\_nexp>

Creates a drawable from a given bitmap specified by the bitmappointer

<bitmap\_ptr\_nexp>. Returns a pointer to the created drawable for use with other

**Gr.drawable**commands.

Note: The bitmap referenced by <bitmap\_ptr\_nexp> must not be deleted.

\*\*\*\*\*\*

Создает отрисовку из заданного растрового изображения, указанного растровым указателем

<Bitmap\_ptr\_nexp>. Возвращает указатель на созданный чертеж для использования с другими

Gr.drawablecommands.

Примечание. Растровое изображение, на которое ссылается <bitmap\_ptr\_nexp>, не должно быть удалено

See alsoGr.bitmap.load

**Gr.drawable.draw <object\_ptr\_nvar>, <drawable\_ptr\_nexp>, left, top, right, bottom** Creates a graphical object that contains a drawable and inserts the object into the Object List. The drawable is specified by the drawable pointer <drawable\_ptr\_nexp>. The drawable will be drawn within the bounds of the parameters.. The command returns the Object List object number of the graphical object in the <object\_ptr\_nvar> variable. This object will not be visible until the **Gr.render** command is called.

The alpha value of the latest **Gr.color** will determine the transparency of the drawable. The **Gr.modify** parameters for **Gr.drawable.draw** are: "drawable", "left", "top", "right" and "bottom".

Using bounds instead of coordinates is due to the animated images. But unfortunately the animated drawable is placed at the upper left corner in the original size until now. It seems the Android developers have forgotten to implement the function setBorder() for animated ones.

\*\*\*\*\*\*

графический объект, который содержит объект рисования и вставляет объект в список объектов. Drawable указывается указателем drawable <drawable\_ptr\_nexp>. Чертеж будет нарисован в пределах параметров. Команда возвращает номер объекта списка объектов графического объекта в переменной <object\_ptr\_nvar>. Этот объект не будет виден, пока не будет вызвана команда Gr.render.

Альфа-значение последней версии Gr.color будет определять прозрачность рисунка. Параметры Gr.modify для Gr.drawable.draw: «drawable», «left», «top», «right» и «bottom».

Использование границ вместо координат связано с анимированными изображениями. Но, к сожалению, до сих пор анимированный чертеж находится в верхнем левом углу в исходном размере. Похоже, разработчики Android забыли реализовать функцию setBorder () для анимированных.

See alsoGr.bitmap.draw

##### Gr.drawable.start <drawable\_ptr\_nvar>

Starts the animation of the drawable <drawable\_ptr\_nvar> if possible.

\*\*\*\*\*\*

Запускает анимацию отрисовки <drawable\_ptr\_nvar>, если это возможно.

##### Gr.drawable.stop <drawable\_ptr\_nvar>

Stops the animation of the drawable <drawable\_ptr\_nvar> if possible.

\*\*\*\*\*\*

Останавливает анимацию рисованного <drawable\_ptr\_nvar>, если это возможно

##### Gr.drawable.delete <drawable\_ptr\_nexp>

Deletes an existing drawable. The drawable's memory is returned to the system. This does not destroy any graphical object that points to the drawable. If you do not

**Gr.hide** such objects, or remove them from the Display List, you will get a run-time error from the next **Gr.render** command.

\*\*\*\*\*\*

Удаляет существующий чертеж. Память отрисовки возвращается в систему. Это не разрушает любой графический объект, который указывает на рисование. Если вы этого не сделаете

Собрав такие объекты или убрав их из списка отображения, вы получите ошибку во время выполнения следующей команды Gr.render.

See alsoGr.bitmap.delete

##### Gr.cls {<clear\_bitmaps/drawables\_nexp>}

Clears the graphics screen. Deletes all previously drawn objects; all existing object references are invalid. Deletes all existing Paints and resets all **Gr.color** or **Gr.text** {size| align|bold|strike|underline|skew} settings. Disposes of the current Object List and Display List and creates a new Initial Display List.

Note: bitmaps and drawables are not deleted. They will not be drawn because no graphical objects point to them, but the bitmaps or drawables still exist. Variables that point to them remainvalid.

If the optional <clear\_bitmaps/drawables\_nexp> is > 0, then all bitmaps and drawables are deleted also.

In this case all variables that point to them are not valid.

The **Gr.render** command must be called to make the cleared screen visible to the user.

\*\*\*\*\*\*

Очищает графический экран. Удаляет все ранее нарисованные объекты; все существующие ссылки на объекты недействительны. Удаляет все существующие краски и сбрасывает все Gr.color или Gr.text {size | выровнять | полужирный | удар | подчеркивание | перекос} настройки. Удаляет текущий список объектов и список отображения и создает новый начальный список отображения.

Примечание: растровые изображения и элементы рисования не удаляются. Они не будут нарисованы, потому что на них не указывают графические объекты, но растровые изображения или графические объекты все еще существуют. Переменные, которые указывают на них, остаются действительными.

Если необязательный <clear\_bitmaps / drawables\_nexp> равен> 0, то все растровые изображения и рисованные объекты также удаляются.

В этом случае все переменные, которые на них указывают, недопустимы.

Команда Gr.render должна быть вызвана, чтобы сделать очищенный экран видимым для пользователя.

##### Gr.statusbar <height\_nvar>} {, showing\_lvar}

Returns information about the Status Bar. If the **height** variable <height\_nvar> is present, it is set to the nominal height of the Status Bar. If the **showing** flag <showing\_lvar> is present, it is set to **0** (false, not showing) or **1** (true, showing) based on on how Graphics Mode was opened.

The parameters are both optional. If you omit the first parameter but use the second, you must keep the comma.

\*\*\*\*\*\*

Возвращает информацию о строке состояния. Если переменная высоты <height\_nvar> присутствует, она устанавливается на номинальную высоту строки состояния. Если присутствует флаг показа <show\_lvar>, он устанавливается в 0 (false, не отображается) или 1 (true, показывается) в зависимости от того, как был открыт графический режим.

Параметры являются необязательными. Если вы пропустите первый параметр, но используете второй, вы должны оставить запятую.

##### Gr.screen <width\_nvar>, <height\_nvar>{{{{{, density\_nvar> }, isRound\_lvar> },

**<layout[ ]>}, <insets[ ]>}, <bounds[ ]>}**

Returns the screen's width and height, and optionally its density, in the numeric variables. The density, in dots per inch (dpi), is a standardized Android density value (usually 120, 160, 240 or 480 dpi), and not necessarily the real physical density of the screen.

If a **Gr.orientation** command changes the orientation, the width and height values from a previous **Gr.screen** command are invalid.

Android’s orientation-change animation takes time. You may need to wait for a secondor so after **Gr.open** or **Gr.orientation** before executing **Gr.screen**, otherwise the width and height values may be set before the orientation change iscomplete.

Placing this command behind OnGrScreen: should solve thisissue.

**Gr.screen** returns a subset of the information returned by the newer **Screen**command. If the display is round isRound returns1.

The <layout[ ]> array returns the bounds of the current display view. [Left, Top, Right, Bottom].

# It is strongly recommended, to use the <layout[ ]> array for graphic- screen size-calculation!

If the device has one or two notches and Android ≥ 9:

The <insets[ ]> array returns the insets of the current display view. [Left, Top, Right, Bottom].

The <bounds[ ]> array returns the bounds of the notch(es). [Left, Top, Right, Bottom{,Left, Top, Right, Bottom} ].

\*\*\*\*\*\*

Возвращает ширину и высоту экрана и, при необходимости, его плотность, в числовых переменных. Плотность в точках на дюйм (dpi) является стандартизированным значением плотности Android (обычно 120, 160, 240 или 480 dpi) и не обязательно является реальной физической плотностью экрана.

Если команда Gr.orientation меняет ориентацию, значения ширины и высоты из предыдущей команды Gr.screen являются недействительными.

Анимация изменения ориентации Android требует времени. Возможно, вам придется подождать второй или после Gr.open или Gr.orientation перед выполнением Gr.screen, в противном случае значения ширины и высоты могут быть установлены до того, как изменение ориентации будет завершено.

Размещение этой команды за OnGrScreen: должно решить эту проблему.

Gr.screen возвращает подмножество информации, возвращаемой более новой командой Screen. Если дисплей круглый, isRound возвращает1.

Массив <layout []> возвращает границы текущего представления экрана. [Слева, сверху, справа, снизу].

Настоятельно рекомендуется использовать массив <layout []> для расчета размера графического экрана!

Если устройство имеет одну или две метки и Android ≥ 9:

Массив <insets []> возвращает вставки текущего отображаемого вида. [Слева, сверху, справа, снизу].

Массив <bounds []> возвращает границы меток. [Слева, сверху, справа, снизу {, слева, сверху, справа, снизу}].

Example:

If a nutch with the dimensions 150 x 100 is in the upper left corner insets[] returns [150, 100, 0, 0]

and

bounds[] returns [0, 0, 150, 100].

See also OnGrScreen:, Screen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table of Decor Options | | | | |
| Decor =  [ 1 ] [ ] [ ][ ] |  | 0 | 1 | 2 |
| [ 1 ] | Place holder |  | In any case |  |
| [ ] | Use the place beside the cut out | No | Yes |  |
| [ ] | Navigation bar | Immersive Sticky Mode (Android 7+, Translucent is the fall back.) | Yes | Translucent |
| [] | Status bar | No | Yes |  |
|  | | | | |
| Decor = [] | Status bar | No | Yes |  |



Decor = **1** % The same as **1011 / Status Bar**

GR.open 255, 100, 200, 50, Decor, 1 % **Returns:**

SCREEN.size, w, 1312.0, h, 2279.0, rW, 1312.0, rH,

2560.0, den, 480.0

GR.Statusbar, 144.0, 1.0

GR.Screen, 1312.0, 2279.0, 480.0, 137.0, 586.0

GR.Screen → LayoutIsRound, 0.0

GR.Screen → Layout, 0.0, 0.0, 1312.0, **2272**.0

GR.Screen → Insets, 0.0, 137.0, 0.0, 0.0

GR.Screen → Bounds, 586.0, 0.0, 726.0, 137.0

Decor = **1101** % **Immersive Sticky Navigation Bar**

GR.open 255, 100, 200, 50, Decor, 1 **% Returns:**

SCREEN.size, w, 1312.0, h, 2279.0, rW, 1312.0, rH,

2560.0, den, 480.0

GR.Statusbar, 144.0, 1.0

GR.Screen, 1312.0, 2279.0, 480.0, 137.0, 586.0

GR.Screen → LayoutIsRound, 0.0

GR.Screen → Layout, 0.0, 0.0, 1312.0, **2560**.0

GR.Screen → Insets, 0.0, 137.0, 0.0, 0.0

GR.Screen → Bounds, 586.0, 0.0, 726.0, 137.0

Decor = **1110** % **Use the place beside thenotch.**

GR.open 255, 100, 200, 50, Decor, 1 %**Returns:**

SCREEN.size, w, 1312.0, h, 2279.0, rW, 1312.0, rH,

2560.0, den, 480.0

GR.Statusbar, 144.0, 0.0

GR.Screen, 1312.0, 2279.0, 480.0, 137.0, 586.0

GR.Screen → LayoutIsRound, 0.0

GR.Screen → Layout, 0.0, 0.0, 1312.0, **2416**.0

GR.Screen → Insets, 0.0, 137.0, 0.0, 0.0

GR.Screen → Bounds, 586.0, 0.0, 726.0, 137.0

##### OnGrScreen:

Interrupt label that traps on reseive a screen layout change. BASIC! executes the statements following the **OnGrScreen:** label until it reaches a **GR.ongerscreen.resume**.

\*\*\*\*\*\*

Метка прерывания, которая перехватывает повторное изменение макета экрана. BASIC! выполняет операторы, следующие за меткой OnGrScreen:, пока не достигнет GR.ongerscreen.resume

##### GR.ongrscreen.resume

Resumes execution at the point in the BASIC! program where the **OnGrScreen:** interrupt occurred.

\*\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, в которой произошло прерывание OnGrScreen:

##### Gr.scale x\_factor, y\_factor{{, x\_distance{, y\_distance}

Scale all drawing commands by the numeric x and y scale factors and translates optional by the numeric x and y distance. This command is provided to allow you to draw in a device-independent manner and then scale and translate the drawing to the actual size of the screen that your program is running on.

The translation will be executed before scaling. If you need translation after scaling multiply factor with distance.

\*\*\*\*\*\*

Масштабируйте все команды рисования с помощью числовых масштабных коэффициентов x и y и переводите необязательно на числовые расстояния x и y. Эта команда позволяет рисовать независимо от устройства, а затем масштабировать и преобразовывать чертеж в фактический размер экрана, на котором работает ваша программа.

Перевод будет выполнен до масштабирования. Если вам нужен перевод после масштабирования, умножьте коэффициент на расстояние.

For example:

! Set the device independent sizes di\_height = 480

di\_width = 800

! Get the actual width and height

gr.open % defaults: white, no status bar, landscape gr.screen actual\_w,actual\_h

! Calculate the scale factors scale\_width = actual\_w /di\_width scale\_height = actual\_h /di\_height

! Set the scale

gr.scale scale\_width, scale\_height

Now, start drawing based upon di\_height and di\_width. The drawings will be scaled to fit the device running the program.

\*\*\*\*\*\*

Теперь начните рисовать на основе di\_height и di\_width. Чертежи будут масштабированы, чтобы соответствовать устройству, на котором запущена программа.

**Gr.scale.touch x\_factor, y\_factor{{{, x\_distance}, y\_distance}, <reverse\_nexp>}**Scale all touch commands by the numeric x and y scale factors and translates optional by the numeric x and y distance. This command is provided to allow you to touch in a device- independent manner, scaling and translating the touches to the actual size of the screen that your program is running on. If <reverse\_nexp> set to 1 you can insert the samevalues used in Gr.scale to compensate its scale and translation values. Default is 0, in this case you have to compute the valuesyourself.

\*\*\*\*\*\*

сенсорные команды с помощью числовых масштабных коэффициентов x и y и необязательные переводы с помощью числового расстояния x и y. Эта команда предназначена для того, чтобы вы могли касаться независимо от устройства, масштабируя и переводя касания в фактический размер экрана, на котором работает ваша программа. Если <reverse\_nexp> установлен в 1, вы можете вставить те же значения, которые используются в Gr.scale, чтобы компенсировать его масштаб и значения перевода. По умолчанию 0, в этом случае вы должны вычислить значения самостоятельно.

See also Gr.scale, Gr.touch, Gr.last.touch

##### Gr.array.touch Array[], <count\_nvar>

Returns all available touched points as an array of pairs (x, y), independent if these are touched or moved. The variable <count\_nvar> returns the number of points. This command should be placed behind onTimer: and before Timer.resume, for best stability. Set.Timer is recommended with 10 or more.

\*\*\*\*\*\*

Возвращает все доступные точки касания в виде массива пар (x, y), независимо от того, были ли они затронуты или перемещены. Переменная <count\_nvar> возвращает количество точек. Эта команда должна быть помещена позади onTimer: и перед Timer.resume, для лучшей стабильности. Set.Timer рекомендуется с 10 или более.

See also Gr.scale, Gr.touch, Gr.last.touch, Gr.scale.touch, Gr.list.touch

##### Gr.list.touch <listX(Y)\_pointer\_nexp>, {<listY\_pointer\_nexp>}, <count\_nvar>

Returns all available touched points, independent if these are touched or moved and as one list of pairs (x, y) if <listY\_pointer\_nexp> is not defined. When <listY\_pointer\_nexp> is used it’s list contains the Y value of the point and the list <listX(Y)\_pointer\_nexp> the X value. The variable <count\_nvar> returns the number of points. This command should be placed behind onTimer: and before Timer resume, for best stability. Set.Timer is recommended with 10 or more.

\*\*\*\*\*\*

Возвращает все доступные точки касания, независимо, если они были затронуты или перемещены, и как один список пар (x, y), если <listY\_pointer\_nexp> не определено. Когда используется <listY\_pointer\_nexp>, его список содержит значение Y точки и список <listX (Y) \_pointer\_nexp> значения X. Переменная <count\_nvar> возвращает количество точек. Эта команда должна быть помещена за onTimer: и перед возобновлением работы таймера, для лучшей стабильности. Set.Timer рекомендуется с 10 или более.

Seealso Gr.scale, Gr.touch, Gr.last.touch, Gr.scale.touch,Gr.array.touch, Timer.set, Sched.set (If you need a secondtimer.)

##### Gr.last.touch <last\_index\_nvar>, <x\_nvar>, <y\_nvar>

Returns the last touched index with <last\_index\_nvar> with the (x, y) coordinates of the touch. If the screen is not currently touched, Touched returns false (0) with the (x,y) coordinates of the last previous touch. If the screen has never been touched, the x and y variables are left unchanged.

The returned values are relative to the actual screen size. Thus if you scaled the screen, you need to scale the returned parameters in the opposite direction.

\*\*\*\*\*\*

Возвращает индекс последнего касания с <last\_index\_nvar> с координатами (x, y) касания. Если к экрану в данный момент не прикасались, Touched возвращает false (0) с координатами (x, y) последнего предыдущего касания. Если к экрану никогда не прикасались, переменные x и y остаются без изменений.

Возвращаемые значения относятся к фактическому размеру экрана. Таким образом, если вы масштабировали экран, вам нужно масштабировать возвращаемые параметры в обратном направлении.

See also Gr.scale, Gr.touch, Gr.scale.touch

##### Gr.touch touched, x, y

Tests for a touch on the graphics screen. If the screen is being touched, Touched is returned as true (not 0) with the (x,y) coordinates of the touch. If the screen is not currently touched, Touched returns false (0) with the (x,y) coordinates of the last previous touch. If the screen has never been touched, the x and y variables are left unchanged. The command continues to return true as long as the screen remains touched.

If you want to detect a single short tap, after detecting the touch, you should loop until touched is false.

\*\*\*\*\*\*

Тесты на прикосновение к графическому экрану. Если к экрану прикасаться, Touched возвращается как true (не 0) с координатами (x, y) касания. Если к экрану в данный момент не прикасались, Touched возвращает false (0) с координатами (x, y) последнего предыдущего касания. Если к экрану никогда не прикасались, переменные x и y остаются без изменений. Команда продолжает возвращать истину до тех пор, пока экран остается на месте.

Если вы хотите обнаружить одно короткое нажатие, после обнаружения касания вы должны выполнить цикл, пока касание не станет ложным

DO

GR.TOUCH touched, x, y UNTIL touched

! Touch detected, now wait for

! finger lifted DO

GR.TOUCH touched, x, y UNTIL !touched

The returned values are relative to the actual screen size. If you have scaled the screen then you need to similarly scale the returned parameters. If the parameters that you used in **Gr.scale** were scale\_x and scale\_y (**Gr.scale scale\_x, scale\_y**) then divide the returned x and y by those same values.

\*\*\*\*\*\*

Возвращаемые значения относятся к фактическому размеру экрана. Если вы масштабировали экран, вам нужно аналогичным образом масштабировать возвращаемые параметры. Если параметры, которые вы использовали в Gr.scale, были scale\_x и scale\_y (Gr.scale scale\_x, scale\_y), то разделите возвращаемые x и y на те же значения.

GR.TOUCH touched, x, y Xscaled = x / scale\_x Yscaled = y / scale\_y

##### Gr.touch2 touched, x, y

The same as **Gr.touch** except that it reports on second simultaneous touch of the screen. WARNING

Use this command with caution, as the event handler works like a tennis player batting against a much too fast set ball-machine. If possible, use Gr.array.touch or Gr.list.touch, as these commands recognize the x-y-status of one or more fingers at the **same** time.

\*\*\*\*\*\*

Так же, как Gr.touch, за исключением того, что он сообщает о втором одновременном касании экрана. ПРЕДУПРЕЖДЕНИЕ

Используйте эту команду с осторожностью, так как обработчик событий работает как теннисист, бьющийся против слишком быстрой установленной шариковой машины. Если возможно, используйте Gr.array.touch или Gr.list.touch, так как эти команды распознают состояние x-y одного или нескольких пальцев одновременно.

Use instead if possible Gr.array.touch or Gr.list.touch!

##### Gr.bounded.touch2 touched, left, top, right, bottom

The same as **Gr.bounded.touch** except that it reports on second simultaneous touch of the screen.

\*\*\*\*\*\*

То же, что и Gr.bounded.touch, за исключением того, что он сообщает о втором одновременном касании экрана.

See also the warning at Gr.touch2

Use instead of Gr.bounded.touch and Gr.bounded.touch2 if possible Gr.list.touch and Within()!

##### OnGrTouch:

Interrupt label that traps any touch **down** on the Graphics screen (see "Interrupt Labels"). BASIC! executes the statements following the **OnGrTouch:** label until it reaches a **Gr.onGrTouch.resume** command.

To detect touches on the Output Console (not in Graphics mode), use **OnConsoleTouch:**.

\*\*\*\*\*\*

Метка прерывания, которая задерживает любое касание на графическом экране (см. «Метки прерывания»). BASIC! выполняет операторы, следующие за меткой OnGrTouch:, пока не достигнет команды Gr.onGrTouch.resume.

Чтобы обнаружить прикосновения к консоли вывода (не в графическом режиме), используйте OnConsoleTouch :.

If you want to detect a double tap use: grTapCounter = 0

PRINT "GR Double Tap Example" DO

UNTIL 0 END"Bye...!"

OnTimer:

IF grTapCounter = 1 PRINT "Only One Tap!" TIMER.CLEAR

grTapCounter = 0 ENDIF TIMER.RESUME

OnGrTouch: grTapCounter++

IF grTapCounter = 1 TIMER.CLEAR TIMER.SET 500

ENDIF

IF grTapCounter = 2 PRINT "Double Tap!"

TIMER.CLEAR %Needed, if you will not end execution. grTapCounter = 0 %Needed, if you will not end execution. PAUSE 2000

END "Bye...!" ENDIF

Gr.onGrTouch.resume

Note: OliBasic's SCHED.SET command can work as a timer, too.

##### OnGrTouchMove:

Interrupt label that traps any touch **move** on the Graphics screen (see "Interrupt Labels"). BASIC! executes the statements following the **OnGrTouchMove:** label until it reaches a **Gr.onGrTouchMove.resume** command.

\*\*\*\*\*\*

Метка прерывания, которая задерживает любое касание, перемещается на графический экран (см. «Метки прерывания»). BASIC! выполняет операторы после метки OnGrTouchMove: до достижения команды Gr.onGrTouchMove.resume.

##### Gr.onGrTouchMove.resume

Resumes execution at the point in the BASIC! program where the **OnGrTouchMove:**

interrupt occurred.

\*\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, где OnGrTouchMove:

произошло прерывание.

##### OnGrTouchUp:

Interrupt label that traps any touch **up** on the Graphics screen (see "Interrupt Labels"). BASIC! executes the statements following the **OnGrTouchUp:** label until it reaches a **Gr.onGrTouchUp.resume** command.

\*\*\*\*\*\*

Метка прерывания, которая задерживает любое прикосновение на графическом экране (см. «Метки прерывания»). BASIC! выполняет операторы после метки OnGrTouchUp: до достижения команды Gr.onGrTouchUp.resume

##### Gr.onGrTouchUp.resume

Resumes execution at the point in the BASIC! program where the **OnGrTouchUp:**

interrupt occurred.

\*\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, где OnGrTouchUp:

произошло прерывание.

**Graphics Setup Commands**

##### Gr.open {{alpha}{, red}{, green}{, blue}{, <Decors\_nexp>}{, <Orientation\_nexp>}}

**{,<Camera\_nexp>}**

Opens the Graphics Screen and puts BASIC! into Graphics Mode. The color values become the background color of the graphics screen. The default color is opaque white (255,255,255,255).

All parameters are optional; use commas to indicate omitted parameters (see Optional Parameters).

Each of the four color components is a numeric expression with a value from 0through

255. If a value is outside of this range, only the last eight bits of the value are used; for example, 257 and 1025 are the same as 1. If any color parameter is omitted, it is setto

255. You are able to use color definitions like "\_100,Green", "\_Blue", "#ff008080" etc. also. See the color definition pages at the End of this appendix.

Beginning with Android 7 and 9 some things have changed around screen decors like the status bar.

Starting with Android 7 the background of the status bar is the same as the graphic screen. If you switch the status bar to on and the background is white you see no text or icon.

So you should darken the background in this case in the bounds of the status bar. See also Gr.Screen.

Starting with Android 9 the operating system supports display cutouts also called notches. The Status Bar Text will be shown on the graphics screen if the <Decors\_nexp> is 1 or 1XX[1]. If the <Decors\_nexp> is 0, 1XX[0] or the <Decors\_nexp> is not present, the Status Bar Text will not be shown.

Ending with Android 6 the background of the status bar is darken to Black by default.

But keep in mind, that the coordinate system starts still in left, top display corner. Thus you have to move your graphic objects down if needed.

Begining with Android 9 you can deal with the bounds of the cut outs.

If <Decors\_nexp> 1X[0]X the navigation bar will be in the Immersive Sticky Mode, thus the navigation bar is only visible, if you stroke a touch from the outside into the screen.

If <Decors\_nexp> 1X[1]X the navigation bar will be displayed in the normal mode.

If <Decors\_nexp> 1X[2]X the navigation bar will be in the Translucent Mode, thus the navigation bar and the content behind are visible.

If <Decors\_nexp> 1[0]XX the drawing layout will be shrinked by the insets of the cut out(s). In this case the coordinate system starts in left, top **shrinked layout** corner.

If <Decors\_nexp> is 1[1]XX the drawing layout is like the display layout. If a cut out or round display edges are within your graphic, it will not be displayed (because no display area) but this area will still be saved in a screenshot. See also Gr.Screen.

If <Decors\_nexp> is **negative**, Gr.open expects a layout bundle defined by items of the second following table like Gr.open "\_Blue", - myLayouBundle, -1.

##### For graphic-screen size-calculation and layout dimensions refer to the command Gr.screen!

The orientation upon opening graphics will be determined by the <Orientation\_nexp> value. <Orientation\_nexp> values are the same as values for the **Gr.orientation**command (see below). If the <Orientation\_nexp> is not present, the default orientation is Landscape. The <Camera\_nexp> parameter sets an optional camera view behind the graphicsscreen.

* 0 No camera view in background(default)
* 1 Camera view with camera onback
* 2 Camera view with camera infront
* If > as number of existing cameras, the camera with the highest id isused. Keep in mind, that Android’s numbering starts with 0 BASIC! starts with1.

Today only Android 6+ is supported.

If no CAMERA permission is granted, <Camera\_nexp> falls back to 0.

##### ~~Note, if <Camera\_nexp> is > 0 and some commands like select, dialog.select, dialogmessage and text.input are used at running in graphics mode, the~~ *~~program will behalted.~~*So use this commands outside the graphic mode if a camera is turned to on. Keep in mind, that Orientation -1 is not allowed if Camera > 0.

If the app goes in background in this GR.Open camera mode, you have to use GR.close after detecting with OnBackground: and Background().

\*\*\*\*\*\*

Открывает графический экран и ставит бейсик! в графический режим. Значения цвета становятся цветом фона графического экрана. Цвет по умолчанию - непрозрачный белый (255,255,255,255).

Все параметры являются необязательными; используйте запятые для обозначения пропущенных параметров (см. Дополнительные параметры).

Каждый из четырех цветовых компонентов представляет собой числовое выражение со значением от 0 до

255. Если значение находится за пределами этого диапазона, используются только последние восемь битов значения; например, 257 и 1025 - это то же самое, что и 1. Если какой-либо параметр цвета пропущен, устанавливается значение

255. Вы также можете использовать определения цвета, такие как "\_100, Зеленый", "\_Blue", "# ff008080" и т. Д. См. Страницы с описанием цветов в конце этого приложения.

Начиная с Android 7 и 9, некоторые вещи изменились в отношении декоров экрана, таких как строка состояния.

Начиная с Android 7 фон строки состояния совпадает с графическим экраном. Если вы включите строку состояния, а фон будет белым, вы не увидите текст или значок.

Таким образом, вы должны затемнить фон в этом случае в границах строки состояния. Смотрите также Gr.Screen.

Начиная с Android 9, операционная система поддерживает вырезы на дисплее, также называемые выемками. Текст строки состояния будет отображаться на графическом экране, если <Decors\_nexp> равен 1 или 1XX [1]. Если <Decors\_nexp> равен 0, 1XX [0] или <Decors\_nexp> отсутствует, текст строки состояния не будет отображаться.

В Android 6 фон строки состояния по умолчанию становится черным.

Но имейте в виду, что система координат начинается еще в левом верхнем углу экрана. Таким образом, вы должны переместить ваши графические объекты вниз, если это необходимо.

Начиная с Android 9 вы можете разобраться с границами вырезов.

Если <Decors\_nexp> 1X [0] X, панель навигации будет находиться в режиме иммерсивного залипания, поэтому панель навигации будет видна, только если вы прикоснетесь к экрану снаружи.

Если <Decors\_nexp> 1X [1] X, панель навигации будет отображаться в обычном режиме.

Если <Decors\_nexp> 1X [2] X, панель навигации будет находиться в полупрозрачном режиме, поэтому панель навигации и содержимое позади будут видны.

Если <Decors\_nexp> 1 [0] XX, компоновка чертежа будет уменьшена вставками вырезов. В этом случае система координат начинается в левом верхнем суженном углу макета.

Если <Decors\_nexp> равно 1 [1] XX, компоновка чертежа аналогична компоновке дисплея. Если вырезанные или закругленные края дисплея находятся внутри вашего графика, он не будет отображаться (потому что нет области отображения), но эта область все равно будет сохранена на скриншоте. Смотрите также Gr.Screen.

Если <Decors\_nexp> отрицательно, Gr.open ожидает пакет макета, определенный элементами второй следующей таблицы, такими как Gr.open "\_Blue", - myLayouBundle, -1.

Для расчета размеров графического экрана и размеров макета см. Команду Gr.screen!

Ориентация при открытии графики будет определяться значением <Orientation\_nexp>. Значения <Orientation\_nexp> совпадают со значениями для команды Gr.orientation (см. Ниже). Если <Orientation\_nexp> отсутствует, ориентацией по умолчанию является альбомная. Параметр <Camera\_nexp> устанавливает дополнительный вид камеры за графическим экраном.

0 Нет камеры в фоновом режиме (по умолчанию)

1 камера с обратной стороны камеры

2 Вид камеры с камерой перед

Если> в качестве числа существующих камер, используется камера с наибольшим идентификатором. Помните, что нумерация Android начинается с 0 BASIC! начинается с 1.

На сегодняшний день поддерживается только Android 6+.

Если разрешение CAMERA не предоставлено, <Camera\_nexp> возвращается к 0.

Обратите внимание, что если <Camera\_nexp> равен> 0, а некоторые команды, такие как select, dialog.select, dialogmessage и text.input, используются при работе в графическом режиме, программа будет отключена. Поэтому использовать эти команды вне графического режима можно, если повернуть камеру вкл. Имейте в виду, что ориентация -1 не допускается, если камера> 0.

Если приложение работает в фоновом режиме в этом режиме камеры GR.Open, вы должны использовать GR.close после обнаружения с помощью OnBackground: и Background ().

See also Gr.screen, Gr.statusbar

Example:

FN.DEF OpenGraphicDisplay() sBr = 1101

SCREEN.SIZE size[], realsize[], density ori = 0

IF size[1] < size[2] THEN ori = 1 cam = 1

GR.OPEN 0, 10, 0, 0, sBr, ori , cam

GR.CAMERA.GETPARAM p$ p$ = REPLACE$(p$, ";",";\n")

PRINT p$

s$ = "effect=mono" GR.CAMERA.SETPARAM s$,1

fl = 3 GR.CAMERA.FLASH fl

GR.STATUSBAR sHeight, sE GR.SCREEN w, h GR.COLOR 255,250,0,0,2

yFromTop = sHeight + 50

GR.RECT rc1, 50, (sHeight\*sE)+ 50, w-50, (sHeight\*sE) + 150, 10 GR.RENDER

FN.RTN 1 FN.END

OpenGraphicDisplay() mGr =1

bg = 0

zoomFactor = 100 GR.CAMERA.ZOOM zoomFactor zoomFactorMax = zoomFactor zoomDirection = 1

DO UNTIL 0

ONGRTOUCH:

IF zoomFactor > (zoomFactorMax-0.1) THEN zoomDirection = -1 zoomFactor = zoomFactor + 0.2 \* zoomDirection

IF zoomFactor < 1 THEN zoomDirection = 1 : zoomFactor = 1 GR.CAMERA.ZOOM zoomFactor

BIGD.ROUND zoomFactor$, STR$(zoomFactor), 1, "HU" sel = -400

DIALOG.MESSAGE "Zoom Factor", zoomFactor$, sel GR.ONGRTOUCH.RESUME

ONBACKGROUND: IF BACKGROUND() IF mGr = 1 GR.CLOSE

mGr=0 ENDIF

bg = 1 ELSE

IF bg = 1 OpenGraphicDisplay() mGr = 1

bg = 0 ENDIF ENDIF

BACKGROUND.RESUME

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table of Decor Options | | | | |
| Decor =  [ 1 ] [ ] [ ][ ] |  | 0 | 1 | 2 |
| [ 1 ] | Place holder |  | In any case |  |
| [ ] | Use the place beside the cut out | No | Yes |  |
| [ ] | Navigation bar | Immersive Sticky Mode (Android 7+. Translucent is the fall back.) | Yes | Translucent |
| [] | Status bar | No | Yes |  |
|  | | | | |
| Decor = [] | Status bar | No | Yes |  |

The optional options bundle (-) <Decors\_nexp> controls the layouts of the Action and Navigation bars. That means, a negative <Decors\_nexp> is interpreted as a bundle pointer.

|  |  |  |
| --- | --- | --- |
| **Table of Layout Control Options** | | |
| **Key** | **Value** | **Description** |
| **\_Title** | String | Set the action bar's title. |
| **\_Subtitle** | String | Set the action bar's subtitle. |
| **\_TitleShow** | 0 or 1 (numeric) | If 1 (default)  Show the Action bar if it is not currently showing. It will resize application content to fit the new space available.  If 0  Hide the Actionbar if it is currently showing. It will resize application content to fit the new space available. |
| **\_TitleIcon** | Icon file path | Add a large icon to the notification content view.<http://romannurik.github.io/>  AndroidAssetStudio/index.html |
| **\_TitleHomeEnabled** | 0 or 1 (numeric) | Set whether to include the application home accordance in the action bar. Home is presented as an activityicon.  Have to be 1 if you want to show the icon.  Have to be 0 if you want to hide the icon.  The default setting is API dependent. |
| **\_TitleBackground** | Background file path |  |

|  |  |  |
| --- | --- | --- |
| **\_TitleHtml** | 0 or 1 (numeric) | Returns displayable styled text from the provided HTML string. But not all tags are supported.    Uses parts of TagSoup library to handle real HTML, including all of the brokenness found in the wild.  <b>  <big>  <font size="..." color="..." face="...">  <h1>, <h2>, <h3>, <h4>, <h5>,  <h6>  <i>  <small>  <strike>? < A.7  <strong>  <sub>  <sup>  <tt>?  <u>  **Replace Space with &#160,**  **& with &amp,**  **< with &lt,**  **> with &gt, " with &quot if necessary.**  Usable for Title andSubtitle.  Keep in mind that the Actionbar height will not beexpanded. |
| **\_ShowStatusbar** | 0, 1 or 2 (numeric) | If 1 (default)  The Status bar will be displayed. If 2  The Status bar will be transparent displayed.  Min. Lollipop 5.0 (API 21)  If 0  The Status bar will be hidden to the background.  Min. Nougat 7.0 (API 24)  Will be switched to option 2 or 1 if the current API level is lower. |
| **\_StatusbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |

|  |  |  |
| --- | --- | --- |
| **\_StatusbarLight** | 0 or 1 (numeric) | If 0 (default)  The Status bar background is dark. In this case the **bar content** will be **light**.  If 1  The Status bar background is light. In this case the **bar content** will be **dark**.  Min. Lollipop 5.0 (API 21) |
| **\_ShowNavigationbar** | 0, 1 or 2 (numeric) | If 1 (default)  The Navigation bar will be displayed.  If 2  The Navigation bar will be transparent displayed.  Min. Lollipop 5.0 (API 21)  If 0  The Navigation bar will be hidden to the background.  Min. Nougat 7.0 (API 24)  Will be switched to option 2 or 1 if the current API level is lower. |
| **\_NavigationbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |
| **\_NavigationbarLight** | 0 or 1 (numeric) | If 0 (default)  The Navigation bar background is dark.  In this case the **bar content** will be  **light**. If 1  The Navigation bar background is light.  In this case the **bar content** will be  **dark**.  Min. Lollipop 5.0 (API 21) |
| **\_Menu** | Menu Bundle Pointer | Creates menu entries.  A successful selection will be returned as a human readable JSONstring.  See the example at Console.title for moredetails. |
| **\_ExtendBesideNotch** | 0 or 1 (numeric) | If 0 (default)  The space beside the notch is not used for graphics.  If 1  The space beside the notch is used also.  Min. Pie 9.0 (API 28) |

|  |  |  |
| --- | --- | --- |
| **\_CameraViewBounds** | left**,**top**,**right**,**bottom  (comma delimited string) | Defines the bounds of a  background camera view in pixels. |

##### IS\_GR <status\_nvar>

IS\_GR returns the graphic mode status. If <status\_nvar> is 1 the graphic mode is enabled. Is 0 returned the graphic mode is not open.

\*\*\*\*\*\*

IS\_GR возвращает статус графического режима. Если <status\_nvar> равен 1, графический режим включен. Возвращено 0, графический режим не открыт.

See also Console.Save

##### GR.set.acceleration <mode\_nvar>

This command overwrites the preference settings. Use this command with care. For starters is a good place directly behind the Gr.open command.

If <mode\_nvar> = 0 (SOFTWARE): The graphic view is rendered in software into a bitmap. If <mode\_nvar> = 1 (HARDWARE): The view is rendered in hardware into a hardware texture if the application is hardware accelerated.

If <mode\_nvar> = 2 (NONE): The view is rendered normally and is not backed by an off- screen buffer.

The default behavior is the mode gotten from the APK-xml or set in the preferences.

The advantage of hardware acceleration is not only the speed, the power consumption is much lower too.

If you cannot use the hardware acceleration, because some details are not displayed, proceed as follows. Create your graphic, if nothing happened in one or two seconds, take a screenshot and display the saved bitmap on top if necessary. Switch to hardware acceleration and wait for an event by OnGrTouch:. Now switch back to software rendering.

\*\*\*\*\*

Эта команда перезаписывает настройки предпочтений. Используйте эту команду с осторожностью. Для начала это хорошее место прямо за командой Gr.open.

Если <mode\_nvar> = 0 (ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ): Графическое представление отображается в программном обеспечении в растровое изображение. Если <mode\_nvar> = 1 (HARDWARE): представление отображается на аппаратном уровне в аппаратную текстуру, если приложение работает с аппаратным ускорением.

Если <mode\_nvar> = 2 (NONE): представление отображается нормально и не поддерживается закадровым буфером.

Поведение по умолчанию - режим, полученный из APK-xml или установленный в настройках.

Преимущество аппаратного ускорения заключается не только в скорости, но и в гораздо меньшем энергопотреблении.

Если вы не можете использовать аппаратное ускорение, поскольку некоторые детали не отображаются, выполните следующие действия. Создайте свою графику, если в течение одной или двух секунд ничего не произошло, сделайте снимок экрана и при необходимости отобразите сохраненное растровое изображение. Переключитесь на аппаратное ускорение и дождитесь события по OnGrTouch :. Теперь вернитесь к программному рендерингу.

For more information consult also: <https://developer.android.com/guide/topics/graphics/hardware-accel#java>Gr.render equals the Java function invalidate().

Look also under Unsupported Drawing Operations. For Android 6+

See also Timer.Set, onTimer:, Sched.Set, onSched:, Gr.ScreenToBitmap, Gr.hide

##### Gr.set.cap {{<cap\_nexp>}{,<paint\_nexp>}}

Sets the line caps of objects drawn after this command is issued. The opportunities of <cap\_nexp> are shown in this table:

\*\*\*\*\*\*

Устанавливает заглавные буквы объектов, нарисованных после выполнения этой команды. Возможности <cap\_nexp> показаны в этой таблице:

|  |  |  |
| --- | --- | --- |
| **Value** | **Meaning** | **DescriptionThe stroke ends with the path, and does not project beyond it.** |
| 0 | BUTT | The stroke ends with the path, and does not project beyond it. |
| 1 | ROUND | The stroke projects out as a semicircle, with the center at the end of the path. |
| 2 | SQUARE | The stroke projects out as a square, with the center at the end of the path. |

Example:

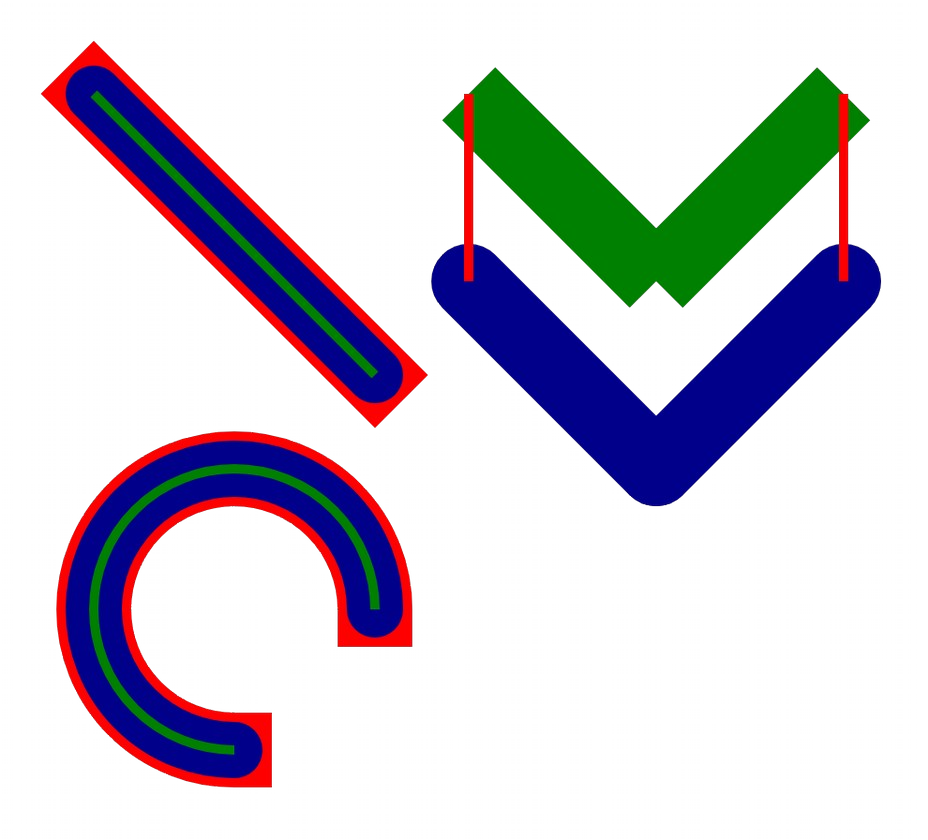
GR.OPEN 80,0,0,0,0,1

GR.ARC GR.SET.CAP 2

GR.SET.STROKE 80

GR.COLOR 255,255,0,0 GR.LINE nn,100,100,400,400

GR.COLOR 255,255,0,0,0 % Red with fill mode = 0, now the default one GR.ARC cc,100,500,400,800, 0, -270, 0

GR.SET.STROKE 60

GR.SET.CAP 1

GR.COLOR "\_DarkBlue" GR.LINE nn,100,100,400,400

GR.ARC cc,100,500,400,800, 0, -270, 0

GR.SET.STROKE 10

GR.COLOR "\_Green" GR.SET.CAP 0

GR.LINE nn,100,100,400,400

GR.ARC cc,100,500,400,800, 0, -270, 0

GR.SET.STROKE 80 GR.LINE nn,500,100,700,300 GR.LINE nn,700,300,900,100

GR.SET.CAP 1

GR.COLOR "\_DarkBlue" GR.LINE nn,500,300,700,500 GR.LINE nn,700,500,900,300

GR.SET.STROKE 10

GR.SET.CAP 0 GR.COLOR "\_Red"

GR.LINE nn,500,100,500,300 GR.LINE nn,900,100,900,300

GR.RENDER DO:UNTIL 0

##### Gr.color {{alpha}{, red}{, green}{, blue}{, style}{, paint}{, xFermode}}

Sets the color and style for drawing objects. There are two ways to use this command.

* *Basicusage*:ignoretheoptional<paint>parameter.Thenewcolorandstylewillbe used for whatever graphical objects are subsequently drawn until the next **Gr.color** command isexecuted.
* *Advanced usage*: The "basic usage" of this command always creates a new Paint. If you prefer, you can use the <paint> parameter to specify an existing Paint. The **Gr.color** command sets the color and style of that Paint, changing the appearance of any graphical object to which it is attached. The current Paint is not changed. See "Paints *Advanced Usage*" above and the examplebelow.

All of the parameters are optional. If a color component or the style is omitted, that component is left unchanged. For example, **Gr.color ,,0** sets only green to 0, leaving alpha, red, blue, and style as they were. Use commas to indicate omitted parameters (see Optional Parameters).

Each of the four color components (alpha, red, green, blue) is a numeric expression with a value from 0 through 255. If a value is outside of this range, only the last eight bits of the value are used; for example, 257 and 1025 are both the same as 1.

The style parameter, is a numeric expression that determines the stroking and filling of objects. The effect of this parameter is explained in detail in the "Style" sections, see above. The possible values for <style\_nexp> are shown in this table:

\*\*\*\*\*\*

Устанавливает цвет и стиль для рисования объектов. Есть два способа использовать эту команду.

Базовое использование: игнорировать необязательный параметр <paint>. Новый цвет и стиль будут использоваться для любых графических объектов, которые будут нарисованы до тех пор, пока не будет выполнена следующая команда Gr.color.

Расширенное использование: «Основное использование» этой команды всегда создает новый Paint. Если вы предпочитаете, вы можете использовать параметр <paint>, чтобы указать существующий Paint. Команда Gr.color устанавливает цвет и стиль этой краски, изменяя внешний вид любого графического объекта, к которому она прикреплена. Текущая краска не изменяется. См. «Расширенное использование красок» выше и пример ниже.

Все параметры являются необязательными. Если компонент цвета или стиль опущен, этот компонент остается неизменным. Например, Gr.color ,, 0 устанавливает только зеленый в 0, оставляя альфа, красный, синий и стиль такими, какими они были. Используйте запятые для обозначения пропущенных параметров (см. Дополнительные параметры).

Каждый из четырех цветовых компонентов (альфа, красный, зеленый, синий) является числовым выражением со значением от 0 до 255. Если значение находится за пределами этого диапазона, используются только последние восемь битов значения; например, 257 и 1025 оба равны 1.

Параметр style - это числовое выражение, которое определяет обводку и заливку объектов. Эффект этого параметра подробно объясняется в разделах «Стиль», см. Выше. Возможные значения для <style\_nexp> показаны в этой таблице:

|  |  |  |
| --- | --- | --- |
| **Value** | **Meaning** | **Description** |
| 0 | STROKE | Geometry and text drawn with this style will be stroked (outlined), respecting the stroke-related fields on the paint. |
| 1 | FILL | Geometry and text drawn with this style will be filled, ignoring all stroke-related settings in the paint. |
| 2 | STROKE\_AN D\_FILL | Geometry and text drawn with this style will be filled and stroked at the same time, respecting the stroke-related fields on the paint. |

If you specify a value other than -1, 0, 1, or 2, then the style is set to 2. If you specify a style of -1, the style is left unchanged, just as if the style parameter were omitted. If you never set a style, the **default** value is **1, FILL**.

You can change the stroke weight with commands such as **Gr.set.stroke** (see below) and the various text style commands.

\*\*\*\*\*\*

Если вы укажете значение, отличное от -1, 0, 1 или 2, тогда стиль будет установлен на 2. Если вы укажете стиль -1, стиль останется неизменным, как если бы параметр стиля был опущен. Если вы никогда не устанавливаете стиль, значением по умолчанию является 1, FILL.

Вы можете изменить вес штриха с помощью таких команд, как Gr.set.stroke (см. Ниже) и различных команд стиля текста.

Example:

GR.OPEN

! basic usage

GR.COLOR,0,0,255,2 % opaque blue, stroke andfill

GR.RECTr1,50,50,100,100 % draw two squares GR.RECT r2,100,100,150,150

GR.COLOR128,255,0,0 % half-transparent red GR.RECTr3,75,75,125,125 % draw an overlapping square GR.RENDER : PAUSE 2000

! advanced usage

GR.GET.VALUE r1,"paint", p % get index of firstPaint

GR.COLOR255,0,255,0,,p % change that Paint’s color to opaquegreen GR.RENDER :PAUSE2000 % both r1 and r2change

GR.RECTr4,125,125,175,175 % use current Paint, unchanged GR.RENDER :PAUSE2000 % still draws half-transparent red GR.CLOSE :END

The argument xFermode sets the Porter-Duff Compositing and Blend Modes.

If you want to use it directly, the Background on Gr.open the alpha has to be set to 0. For a mask use Gr.bitmap.drawinto.start with Gr.bitmap.drawinto.end.

\*\*\*\*\*\*

Аргумент xFermode устанавливает режимы композитинга и смешивания Портера-Даффа.

Если вы хотите использовать его напрямую, фон для Gr.open alpha должен быть установлен на 0. Для маски используйте Gr.bitmap.drawinto.start с Gr.bitmap.drawinto.end.

|  |  |  |
| --- | --- | --- |
| **Index** | **Mode** | **Formula** |
| -1 | NORMAL | **Default** |
| 0 | CLEAR | [Sa, Sc] |
| 1 | SRC | [Da, Dc] |
| 2 | DST | [Sa + (1 - Sa)\*Da, Rc = Sc + (1 - Sa)\*Dc] |
| 3 | SRC\_OVER | [Sa + (1 - Sa)\*Da, Rc = Dc + (1 - Da)\*Sc] |
| 4 | DST\_OVER | [Sa \* Da, Sc \* Da] |
| 5 | SRC\_IN | [Sa \* Da, Sa \* Dc] |
| 6 | DST\_IN | [Sa \* (1 - Da), Sc \* (1 - Da)] |
| 7 | SRC\_OUT | [Da \* (1 - Sa), Dc \* (1 - Sa)] |
| 8 | DST\_OUT | [Da, Sc \* Da + (1 - Sa) \* Dc] |
| 9 | SRC\_ATOP | [Sa, Sa \* Dc + Sc \* (1 - Da)] |
| 10 | DST\_ATOP | [Sa + Da - 2 \* Sa \* Da, Sc \* (1 - Da) + (1 - Sa) \* Dc] |
| 11 | XOR | [Sa + Da – Sa\*Da, Sc\*(1 - Da) + Dc\*(1 - Sa) + min(Sc, Dc)] |
| 12 | DARKEN | [Sa + Da – Sa\*Da, Sc\*(1 - Da) + Dc\*(1 - Sa) + max(Sc, Dc)] |
| 13 | LIGHTEN | [Sa \* Da, Sc \* Dc] |
| 14 | MULTIPLY | [Sa + Da - Sa \* Da, Sc + Dc - Sc \* Dc] |
| 15 | SCREEN | Saturate(S + D) |
| 16 | ADD |  |
| 17 | OVERLAY |  |

For more information see: <https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html>Example:

GR.OPEN 0

GR.COLOR 150, 0, 155,0,1, , -1

GR.RECT destinationR, 20,80,400,600

GR.COLOR 255, 255, 0,0,1, , 0

GR.CIRCLE sourceC, 300, 300, 200 GR.RENDER

DO UNTIL 0

##### You are able to use color definitions like "\_100,Green", "\_Blue", "#ff008080" etc. also. See the color definition pages at the end of this appendix.

**Gr.orientation <nexp> depriciate**

The value of the <nexp> sets the orientation of screen as follows:

-1 = Orientation depends upon the sensors. 0 = Orientation is forced to Landscape.

1 = Orientation is forced to Portrait.

2 = Orientation is forced to Reverse Landscape. 3 = Orientation is forced to Reverse Portrait.

You can monitor changes in orientation by reading the screen width and height using the the **Gr.screen** or **Screen** commands.

Keep in mind, that the camera orientation in the GR.Open camera mode is not changed.

\*\*\*\*\*\*

Значение <nexp> устанавливает ориентацию экрана следующим образом:

-1 = ориентация зависит от датчиков. 0 = Ориентация принудительно настроена на Пейзаж.

1 = ориентация принудительно установлена ​​в портрет.

2 = Ориентация вынуждена изменить ландшафт. 3 = Ориентация вынуждена изменить портрет.

Вы можете отслеживать изменения в ориентации, считывая ширину и высоту экрана с помощью команд Gr.screen или Screen.

Имейте в виду, что ориентация камеры в режиме GR. Открытая камера не изменяется.

### Paint Commands

##### Gr.paint.set <bundle\_nexp>{, <paint\_nexp>}

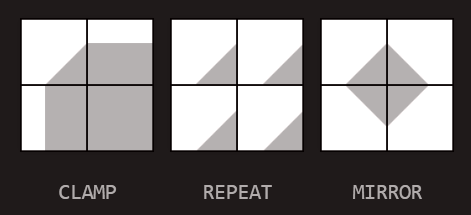
Creates a new Paint with arguments provided by the given bundle <bundle\_nexp>. The optional <paint\_nexp> selects the Paint definition to overwrite. If<paint\_nexp> is -1 (default) a new Paint definition iscreated.

\*\*\*\*\*\*

Создает новый Paint с аргументами, предоставленными данным пакетом <bundle\_nexp>. Необязательный <paint\_nexp> выбирает определение Paint для перезаписи. Если <paint\_nexp> равен -1 (по умолчанию), создается новое определение Paint.

|  |  |  |
| --- | --- | --- |
| **Table of Bundle Keys** | | |
| **Key** | **Value** | **Description** |
| **\_Shader** | \_LinearGradient,  \_Pattern,  \_RadialGradient or  \_SweepGradient (String) | Creates a Shader |
| **\_LinearGradient** | | |
| **\_X0** | (numeric) | Default is 0.  Upper left corner in X direction |
| **\_Y0** | (numeric) | Default is 0.  Upper left corner in Y direction |
| **\_X1** | (numeric) | Default is 0.  Lower right corner in X direction |
| **\_Y1** | (numeric) | Default is 0.  Lower right corner in Y direction |
| **\_ColorArray** | Array of  {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Default is {"\_Black", "\_White"}  An array of colors that  creates the gradient from the top to the bottom. |
| **\_PositionArray** | Array  (numeric) | Default is none.  Array of positions |
| **\_TileMode** | \_Clamp, \_Mirror or  \_Repeat (String) | Default is "\_Mirror". Sets the tile mode |
| **\_Degree** | (numeric) | Default is 0.  Turns the Shader clockwise. |
| **\_Pattern** (Bitmap Shader) | | |
| **\_Bitmap** | Bitmap object number (numeric) | Source bitmap |
| **\_Degree** | (numeric) | Default is 0.  Turns the Shader clockwise. |
| **\_TileModeX** | \_Clamp, \_Mirror or  \_Repeat (String) | Default is "\_Repeat".  Sets the tile mode in X direction |
| **\_TileModeY** | \_Clamp, \_Mirror or  \_Repeat (String) | Default is "\_Repeat".  Sets the tile mode in Y direction |
| **\_RadialGradient** | | |

|  |  |  |
| --- | --- | --- |
| **\_CenterX** | (numeric) | Default is 0.  Center point in X direction |
| **\_CenterY** | (numeric) | Default is 0.  Center point in Y direction |
| **\_Radius** | > 0  (numeric) |  |
| **\_ColorArray** | Array of  {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Default is {"\_Black", "\_Withe"} |
| **\_StopArray** | Array (numeric) | Default is none. Array of stop points |
| **\_TileMode** | \_Clamp, \_Mirror or  \_Repeat (String) | Default is "\_Mirror". Sets the tile mode |
| **\_Degree** | (numeric) | Default is 0.  Turns the Shader clockwise. |
| **\_SweepGradient** | | |
| **\_CenterX** | (numeric) | Default is 0.  Center point in X direction |
| **\_CenterY** | (numeric) | Default is 0.  Center point in Y direction |
| **\_ColorArray** | Array of  {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Default is {"\_Black", "\_Withe"} |
| **\_StopArray** | Array (numeric) | Default is none. Array of stop points |
| **\_Degree** | (numeric) | Default is 0.  Turns the Shader clockwise. |



**Tile Modes (Source: [http://chiuki.github.io/android-shaders-filters/#/8)](http://chiuki.github.io/android-shaders-filters/%23/8))**

Example:

BUNDLE.PUT bundlePtr2, "\_Shader", "\_Pattern" GR.BITMAP.LOAD sp1, "cartman.png" GR.BITMAP.PUT bundlePtr2, "\_Bitmap", sp1 BUNDLE.PUT bundlePtr2, "\_Degree", 30 GR.PAINT.SET bundlePtr2

### Gr.paint.reset {<nexp>}

Force the specified Paint to default settings:

\*\*\*\*\*\*

Принудительно назначить указанным Paint настройки по умолчанию:

Color opaque black (255, 0, 0, 0) Antialias ON

Style FILL (**1**)

Minimum stroke width (0.0) Stroke Cap (0) → "\_Butt" Xfermode (-1)

The parameter is optional. If the parameter is omitted or set to -1, a new current Paintis created with defaultsettings.

\*\*\*\*\*\*

Параметр является необязательным. Если параметр опущен или установлен на -1, новый текущий Paintis создан с настройками по умолчанию.

##### Gr.camera.flash <Flash\_mode\_nval>

Sets the Flash mode of the chosen camera.

The Flash\_mode numeric value specifies the flash operation:

\*\*\*\*\*\*

Устанавливает режим вспышки выбранной камеры.

Числовое значение Flash\_mode указывает операцию flash:

|  |  |
| --- | --- |
| **0** | **Auto Flash** |
| **1** | **Flash On** |
| **2** | **Flash Off** |
| **3** | **Torch** |
| **4** | **Red-eye** |

The default, if no parameter is given, is (2) Flash Off.

If a failure occurs, <Flash\_mode\_nval> returns -1.

##### Available if the camera is opened with Gr.open.

If no camera is running, you can use the flash like a torch. But this is only available in Graphics Mode.

\*\*\*\*\*\*

По умолчанию, если параметры не заданы, это (2) Flash Off.

Если происходит сбой, <Flash\_mode\_nval> возвращает -1.

Доступно, если камера открыта с помощью Gr.open.

Если камера не работает, вы можете использовать вспышку как фонарик. Но это доступно только в графическом режиме.

Torch Example:

Gr.open 255,80,80,80, 1, -1, 1

on = 3 Gr.camera.flash on Pause 5000

off = 2 Gr.camera.flash off

##### Gr.camera.focus <Focus\_mode\_nval>

Sets the focus mode of the chosen camera.

The Focus\_mode numeric value specifies the camera focus:

\*\*\*\*\*\*

Устанавливает режим фокусировки выбранной камеры.

Числовое значение Focus\_mode указывает фокус камеры:

|  |  |
| --- | --- |
| **0** | **Auto Focus** |
| **1** | **Fixed Focus** |
| **2** | **Focus at Infinity** |
| **3** | **Macro Focus (close-up)** |
| **4** | **Continuous Picture** |
| **5** | **Continuous Video** |

The default, if no parameter is given, is Auto Focus and Continuous Picture.

If a failure occurs, <Focus\_mode\_nval> returns-1.

Available if the camera is opened withGr.open.

\*\*\*\*\*\*

По умолчанию, если параметры не заданы, это Автофокус и Непрерывное изображение.

Если происходит сбой, <Focus\_mode\_nval> возвращает -1.

Доступно, если камера открыта с помощью G.open.

**Gr.camera.zoom <Zoom\_factor\_nval>**

Sets the zoom factor of the chosen camera.

If the zoom factor is larger as the camera’s zoom maximum,<Zoom\_factor\_nval> returns the maximum zoom factor.

If a failure occurs, <Zoom\_factor\_nval> returns -1.

##### Available if the camera is opened with Gr.open.

##### \*\*\*\*\*\*

##### Устанавливает коэффициент увеличения выбранной камеры.

##### Если коэффициент увеличения больше максимума увеличения камеры, <Zoom\_factor\_nval> возвращает максимальный коэффициент увеличения.

##### Если происходит сбой, <Zoom\_factor\_nval> возвращает -1.

##### Доступно, если камера открыта с помощью Gr.open.

**Gr.camera.getparam<param\_svar>**

Returns the actual cameraparameter.

##### Can be used with Android 5+ and with Android 4 with limits.

##### \*\*\*\*\*\*

##### Возвращает фактический параметр камеры.

##### Может использоваться с Android 5+ и с Android 4 с ограничениями.

**Gr.camera.setparam<param\_sexp>** Sets one ore more cameraparameter. **Can be used with Android6+.**

**\*\*\*\*\*\***

**Устанавливает один или несколько параметров камеры. Может использоваться с Android 6.**

Example:

GR.CAMERA.GETPARAM p$ p$ = REPLACE$(p$, ";",";\n")

PRINT p$

s$ = "effect=mono;iso=ISO200" % The ; is the delimiter. GR.CAMERA.SETPARAM s$,1

##### Gr.camera.directshoot <bm\_ptr\_nvar>{{,<file\_name\_sexp>} <size\_sexp>}

An image is captured as soon as the command is executed. The optional <file\_name\_sexp> takes the wished file name.

The optional argument <size\_sexp> can be "\_Max"~~, ("\_Screen")~~or a direct size like "1920x1080".

<bm\_ptr\_nvar> returns a bitmap. Is it smaller as the screen, it is not scaled. In the opposite case the bitmap is scaled in the manner that it only covers the screen in size, thus this is in the width or the height it is a little larger.

##### Available if the camera is opened with Gr.open.

Some Android devices are failing until now. Maybe use a Gr.camera.\*\*\*Shoot command instead.

See also File.root

\*\*\*\*\*\*

Изображение захватывается, как только команда выполнена. Необязательный <file\_name\_sexp> принимает желаемое имя файла.

Необязательный аргумент <size\_sexp> может быть «\_Max», («\_Screen») или прямой размер, например «1920x1080».

<bm\_ptr\_nvar> возвращает растровое изображение. Это меньше, как экран, он не масштабируется. В противоположном случае растровое изображение масштабируется таким образом, чтобы оно охватывало только экран по размеру, поэтому оно по ширине или высоте немного больше.

Доступно, если камера открыта с помощью Gr.open.

Некоторые устройства Android отказывают до сих пор. Возможно, вместо этого используйте команду Gr.camera. \*\*\* Shoot.

Смотрите также File.root

##### Gr.camera.select 1|2|...

Selects the Back (1) or Front(2) camera in devices with two cameras. The default camera is the back (opposite the screen) camera. Today a device can have more than two cameras.

If only one exists camera exists, then the default will be that camera. For example, if the device (such as the Nexus 7) only has a Front Camera then it will be the default camera. If the device does not have any installed camera apps, then there will be a run-time error message, "This device does not have a camera." In addition, a run-time error message will be shown if the device does not have the type of camera (front or back) selected.

\*\*\*\*\*\*

Выбирает заднюю (1) или переднюю (2) камеру в устройствах с двумя камерами. Камера по умолчанию - это задняя (напротив экрана) камера. Сегодня устройство может иметь более двух камер.

Если существует только одна камера, существует камера по умолчанию. Например, если устройство (например, Nexus 7) имеет только переднюю камеру, то это будет камера по умолчанию. Если на устройстве нет установленных приложений камеры, появится сообщение об ошибке во время выполнения: «На этом устройстве нет камеры». Кроме того, будет отображаться сообщение об ошибке во время выполнения, если на устройстве не выбран тип камеры (передняя или задняя).

##### Gr.camera.shoot <bm\_ptr\_nvar>{, 0, 0, 0, 0, <file\_name\_sexp>}

The command calls the device’s built in camera user interface to take a picture. The image is returned to BASIC! as a bitmap pointed to by the bm\_ptr numeric variable. If the camera interface does not, for some reason, take a picture, bm\_ptr will be returned with a zero value.

The command also stores the captured image into the file, "<pref base drive>/rfo-basic/data/image.jpg(png)" as default if possible. If the optional

<file\_name\_sexp> is specified with "", the default file path is used. <bm\_ptr\_nvar> returns a bitmap. Is it smaller as the screen, it is not scaled. In the opposite case the bitmap is

scaled in the manner that it only covers the screen in size, thus this is in the widght or the height it is a little larger.

Many of the device camera interfaces will also store the captured images somewhere else in memory with a date coded filename. These images can be found with the gallery application. BASIC! is not able to prevent these extraneous files from being created.

Note: Some devices like the Nexus 7 do not come with a built in camera interface. If you have installed an aftermarket camera application then it will be called when executing this command. You can take pictures with the Nexus 7 (or similar devices) using the other commands even if you do not have camera application installed. If the device does not have any installed camera apps, then there will be a run-time error message, "This device does not have a camera."

\*\*\*\*\*\*

Команда вызывает встроенный пользовательский интерфейс камеры, чтобы сделать снимок. Изображение возвращается на бейсик! как растровое изображение, на которое указывает числовая переменная bm\_ptr. Если интерфейс камеры по какой-то причине не делает снимок, bm\_ptr будет возвращено с нулевым значением.

Команда также сохраняет захваченное изображение в файл, «<pref base drive> /rfo-basic/data/image.jpg (png)» по умолчанию, если это возможно. Если опционально

<file\_name\_sexp> указывается с "", используется путь к файлу по умолчанию. <bm\_ptr\_nvar> возвращает растровое изображение. Это меньше, как экран, он не масштабируется. В противоположном случае растровое изображение

масштабируется таким образом, что он покрывает только экран по размеру, таким образом, это в ширину или высоту немного больше.

Многие из интерфейсов камеры устройства также будут хранить захваченные изображения в другом месте в памяти с именем, закодированным в дату. Эти изображения можно найти с помощью приложения галереи. BASIC! не может предотвратить создание этих посторонних файлов.

Примечание. Некоторые устройства, такие как Nexus 7, не оснащены встроенным интерфейсом камеры. Если вы установили приложение камеры вторичного рынка, оно будет вызываться при выполнении этой команды. Вы можете делать снимки с Nexus 7 (или аналогичными устройствами), используя другие команды, даже если у вас не установлено приложение камеры. Если на устройстве нет установленных приложений камеры, появится сообщение об ошибке во время выполнения: «На этом устройстве нет камеры».

##### Gr.camera.autoshoot <bm\_ptr\_nvar>{{{{{, <flash\_ mode\_nexp> },

**<focus\_mode\_nexp>} , <orientation\_nexp>} , <take\_params\_nexp>} ,**

**<file\_name\_sexp>}**

An image is captured as soon as the command is executed. No user interaction is required. This command can be used for untended, time-sequence image captures. The optional flash\_mode numeric expression specifies the flash operation:

\*\*\*\*\*\*

Изображение захватывается, как только команда выполнена. Взаимодействие с пользователем не требуется. Эта команда может использоваться для непреднамеренных захватов изображений с временной последовательностью. Необязательное числовое выражение flash\_mode определяет операцию flash:

|  |  |
| --- | --- |
| **0** | **Auto Flash** |
| **1** | **Flash On** |
| **2** | **Flash Off** |
| **3** | **Torch** |
| **4** | **Red-eye** |

The default, if no parameter is given, is Auto Flash.

The optional focus\_mode numeric expression specifies the camera focus:

|  |  |
| --- | --- |
| **0** | **Auto Focus** |
| **1** | **Fixed Focus** |
| **2** | **Focus at Infinity** |
| **3** | **Macro Focus (close-up)** |

The default, if no parameter is given, is Auto Focus.

If you want to specify a focus mode, you must also specify a flash mode.

If the <orientation\_nexp> is not present or -1, the default orientation is Landscape. If <take\_params\_nexp> is greater than 0 camera parameters are taken over. :

|  |  |
| --- | --- |
| **0** | **No take over** |
| **1** | **parameters created with Gr.camera.setparam in the simple GR.open mode** |
| **2** | **parameters created with Gr.camera.setparam in the GR.open mode with hidden camera preview** |

<flash\_ mode\_nexp>, <focus\_mode\_nexp>, <orientation\_nexp> and picture size are overwritten by the command settings and default settings. Today only Android 6+!

The command also stores the captured image into the file, "<pref base

drive>/rfo-basic/data/image.jpg(png)" as default. If the optional <file\_name\_sexp> is specified with "", the default file path is used. <bm\_ptr\_nvar> returns a bitmap, that only covers the screen in size, thus this is in the widght or the height it is a little larger.

##### Keep in mind, that the time gap between two Gr.camera.\*\*\*Shoot commands should be round about 500 milliseconds. If a user input is necessary, a separate Pause command is not needed.

##### \*\*\*\*\*\*

##### Имейте в виду, что промежуток времени между двумя Gr.camera. \*\*\* Команды съемки должен составлять около 500 миллисекунд. Если пользовательский ввод необходим, отдельная команда Пауза не требуется.

**Gr.camera.manualShoot <bm\_ptr\_nvar>{{{{{, <flash\_ mode\_nexp> },**

**<focus\_mode\_nexp>} , <orientation\_nexp>} , <take\_params\_nexp>} ,**

**<file\_name\_sexp>}**

This command is much like **Gr.camera.autoshoot** except that a live preview is shown on the screen. The image will not be captured until the user taps the screen.

\*\*\*\*\*\*

Эта команда очень похожа на Gr.camera.autoshoot за исключением того, что на экране отображается предварительный просмотр в реальном времени. Изображение не будет снято, пока пользователь не коснется экрана.

##### Gr.camera.takeVideo <file\_name\_sexp> {{, <duration\_limit\_nexp> },

**<size\_limit\_nexp>}**

The command calls the device’s built in camera user interface to take a video. The argument <file\_name\_sexp> sets the file name to store the video.

The file name should end with ".mp4".

If the file name is "", the file should be saved at the standard location, unfortunately, this is not always guaranteed.

The duration limit can be set in seconds with time <duration\_limit\_nexp>. This is not always guaranteed also.

The maximal file size can be set with <size\_limit\_nexp>. And the same restriction as before.

\*\*\*\*\*\*

Команда вызывает встроенный пользовательский интерфейс камеры, чтобы снять видео. Аргумент <file\_name\_sexp> устанавливает имя файла для хранения видео.

Имя файла должно заканчиваться на «.mp4».

Если имя файла "", файл должен быть сохранен в стандартном месте, к сожалению, это не всегда гарантируется.

Предел продолжительности может быть установлен в секундах со временем <duration\_limit\_nexp>. Это не всегда гарантировано.

Максимальный размер файла может быть установлен с помощью <size\_limit\_nexp>. И то же ограничение, что и раньше.

Example:

Gr.camera.takeVideo "video.mp4", 5, 12\*1048\*1048 %=12MB File.root path$

!vv Shows the video based on an application to be selected

Browse "file://" + path$ + "/" + "video.mp4" % Absolute file path needed!

### Other Graphics Commands

##### Gr.screen.to\_bitmap <bm\_ptr\_nvar>

The current contents of the screen will be placed into a bitmap. The pointer to the bitmap will be returned in the bm\_ptr variable. If there is not enough memory available to create the bitmap, the returned bitmap pointer is -1. Call **GETERROR$()** for information about the failure.

Please note the idiosyncratic underscore in the command.

If a camera preview is behind the graphic screen, this preview is not visible.

So take a photo, convert (scale, crop) its size fitted to the screen, put the result in the graphic screen background, start Gr.screen.to\_bitmap and delete the background.

\*\*\*\*\*\*

Текущее содержимое экрана будет помещено в растровое изображение. Указатель на растровое изображение будет возвращен в переменной bm\_ptr. Если для создания растрового изображения недостаточно памяти, возвращаемый указатель растрового изображения равен -1. Вызовите GETERROR $ () для получения информации о сбое.

Пожалуйста, обратите внимание на своеобразное подчеркивание в команде.

Если предварительный просмотр камеры находится за графическим экраном, этот предварительный просмотр не отображается.

Поэтому сделайте фотографию, преобразуйте (масштабируйте, обрежьте) ее размер, соответствующий размеру экрана, поместите результат в фон графического экрана, запустите Gr.screen.to\_bitmap и удалите фон.

##### Audio.info <aft\_nvar>, <bundle\_pointer\_nvar>

Returns a bundle with the system value keys:

\_Album, \_Artist, \_Title, \_BitRate, \_Duration, \_LocationPath, \_StreamMetadata

\*\*\*\*\*\*

Возвращает пакет с ключами системного значения:

\_Album, \_Artist, \_Title, \_BitRate, \_Duration, \_LocationPath, \_StreamMetadata

##### Audio.load <aft\_nvar>, <filename\_sexp>|<http\_stream\_sexp>

Loads a music file or stream into the Audio File Table. The AFT index is returned in

<aft\_nvar>. If the file or stream can’t be loaded, the <aft\_nvar> is set to 0. Your program should test the AFT index to find out if the file or stream was loaded. If the AFT index is 0, you can call the **GETERROR$()** function to get information about the error. If you use index 0 in another **Audio** command you will get a run-time error.

The file must be in the "<pref base drive>/ref-basic/data/" directories or one of its subdirectories.

You can reach outside the "<pref base drive>/ref-basic/data/" by using path fields in the filename. For example, "../../Music/Blue Danube Waltz.mp3" would access "<pref base drive>/Music/Blue Danube Waltz.mp3" or take a look at **File.root**.

\*\*\*\*\*\*

Загружает музыкальный файл или поток в таблицу аудиофайлов. Индекс AFT возвращается в

<Aft\_nvar>. Если файл или поток не могут быть загружены, <aft\_nvar> имеет значение 0. Ваша программа должна проверить индекс AFT, чтобы выяснить, был ли загружен файл или поток. Если индекс AFT равен 0, вы можете вызвать функцию GETERROR $ (), чтобы получить информацию об ошибке. Если вы используете индекс 0 в другой аудио команде, вы получите ошибку во время выполнения.

Файл должен находиться в каталогах «<pref base drive> / ref-basic / data /» или в одном из его подкаталогов.

Вы можете выйти за пределы «<pref base drive> / ref-basic / data /», используя поля пути в имени файла. Например, "../../Music/Blue Danube Waltz.mp3" может открыть "<pref base drive> / Music / Blue Danube Waltz.mp3" или взглянуть на File.root.

Example:

FILE.ROOT dataPath$, "\_Music"

fn$ = "file://" + dataPath$ + "/" + "Blue Danube Waltz.mp3" FILE.EXISTS ok, fn$

IF ok != 0

AUDIO.LOAD aft, fn$ IF aft != 0

AUDIO.STOP AUDIO.PLAY aft

ENDIF ENDIF

…

AUDIO.LOAD aft, "http://amp1.cesnet.cz:8000/cro1.ogg"

##### Audio.play <aft\_nexp>{, <output\_nexp>}

Selects the file from the Audio File Table pointed to by <aft\_nexp> and begins to play it. There must not be an audio file already playing when this command is executed. If there is a file playing, execute audio.stop first.

Using <output\_nexp> you get control over the output channels. For the loud speakers **or** the ear phones use the 1, level-controlled by the Music mode (**♪**). If the ear phones **and** the loud speakers should play use the 2, level-controlled by the Alarm mode ().

At this moment the current specific outer volume level is saved.

The music stops playing when the program stops running. To simply start a music file playing and keep it playing, keep the program running. This infinite loop will accomplish that:

\*\*\*\*\*\*

Выбирает файл из таблицы аудиофайлов, на который указывает <aft\_nexp>, и начинает его воспроизведение. При выполнении этой команды не должно быть аудиофайлов, которые уже воспроизводятся. Если файл воспроизводится, сначала выполните audio.stop.

Используя <output\_nexp>, вы получаете контроль над выходными каналами. Для громкоговорителей или наушников используйте 1, регулируемый по уровню в режиме музыки (♪). Если наушники и громкоговорители должны воспроизводиться, используйте 2, контролируемые по уровню в режиме будильника ().

В данный момент текущий конкретный внешний уровень громкости сохраняется.

Музыка перестает играть, когда программа перестает работать. Чтобы просто запустить воспроизведение музыкального файла и продолжить его воспроизведение, продолжайте работу программы. Этот бесконечный цикл выполнит это:

Audio.load ptr, "my\_music.mp3" Audio.play ptr

Do

Pause 5000

Until 0

##### Audio.stop { <reset\_vol\_nexp>}

Audio.stop terminates the currently-playing music file. The command will ignored is no file is playing. It is best to precede each Audio.play command with an Audio.stop command.

If the <reset\_vol\_nexp> is set to 1 instead of 0 the outer volume levels will be reset tothe saved one at Audio.play. Default <reset\_vol\_nexp> is set to1.

\*\*\*\*\*\*

Audio.stop завершает воспроизведение текущего музыкального файла. Команда игнорируется, если файл не воспроизводится. Перед каждой командой Audio.play лучше всего предшествовать команде Audio.stop.

Если для <reset\_vol\_nexp> установлено значение 1 вместо 0, внешние уровни громкости будут сброшены до сохраненного уровня в Audio.play. По умолчанию <reset\_vol\_nexp> установлен в 1.

##### Audio.volume <left\_nexp>, <right\_nexp>{,<outer\_nexp>}

Android devices have an inner and an outer volume control. The outer one is controlled by the device’s volume control buttons (key 24 and 25). Changes the inner volume of the left and right stereo channels. There must be a currently playing stream when this command is executed.

The values should range between 0.0 (lowest) to 1.0 (highest). But the human ear perceives the level of sound changes on a logarithmic scale.

The inner volume can only change in the given range of the outer volume control. Optionally with <outer\_nexp> the outer volume level can be controlled in a range between

* 1. (lowest) to 1.0 (highest). But in this case the logarithmic scale is already taken into account.OnAudio.playthecurrentoutervolumelevelaresaved.OnAudio.stoporthe program stops, the outer volume level will be reset to the savedlevels.

Btw. instead of "volume", "sound pressure level" <SPL> is the correct name.

~~The ear perceives a 10db change as twice as loud. A 20db change would be four times asloud.~~

~~A 1 db change would be about 0.89. One way to implement a volume control would be setup a volume table with 1db level changes. The following code creates a 16 step table.~~

\*\*\*\*\*\*

Устройства Android имеют внутренний и внешний регулятор громкости. Внешний управляется кнопками регулировки громкости устройства (клавиши 24 и 25). Изменяет внутреннюю громкость левого и правого стереоканалов. При выполнении этой команды должен быть текущий воспроизводимый поток.

Значения должны находиться в диапазоне от 0,0 (самое низкое) до 1,0 (самое высокое). Но человеческое ухо воспринимает уровень звуковых изменений в логарифмическом масштабе.

Внутренняя громкость может изменяться только в заданном диапазоне внешнего регулятора громкости. Опционально с помощью <external\_nexp> можно регулировать уровень внешней громкости в диапазоне от

(самый низкий) до 1,0 (самый высокий). Но в этом случае логарифмическая шкала уже учтена. OnAudio.playthecurrentoutervolumelevelaresaved.OnAudio.stoport, программа остановится, уровень внешней громкости будет сброшен до сохраненных уровней.

Btw. вместо «громкость» правильное название - «уровень звукового давления» <SPL>.

Ухо воспринимает изменение на 10 дБ вдвое громче. Изменение в 20 дБ будет в четыре раза громче.

Изменение на 1 дБ составило бы около 0,89. Одним из способов реализации регулятора громкости будет настройка таблицы громкости с изменениями уровня 1 дБ. Следующий код создает таблицу из 16 шагов.

~~dim volume[16]x =1~~

~~volume [1] = xfor i = 2 to 16~~

~~x = x \* 0.89volume [i] = x~~

~~next I~~

maxVolume = 10 % Wished steps FOR vol = 0 TO maxVolume

myVolume = (1 - (LOG(maxVolume- vol) / LOG(maxVolume)))

!PRINT mVolume

AUDIO.VOLUME myVolume/maxVolume, myVolume/maxVolume, 1

! AUDIO.VOLUME 1, 1, vol/ maxVolume

PAUSE 1000

NEXT

~~Your code can select volume values from the table for use in the audio.volume command.The loudest volume would be volume[1].~~

\*\*\*\*\*\*

Ваш код может выбрать значения громкости из таблицы для использования в команде audio.volume. Самым громким будет громкость [1].

##### Audio.record.peak <level\_nvar>

With this PEAK command you get with <level\_nvar> the raw max. sound pressure level amplitude value between to calls.

The Audio.record.START command initializes the PEAK process for the first time. If no current recording process available, PEAK returns -1.

\*\*\*\*\*\*

С помощью этой команды PEAK вы получите <level\_nvar> raw max. значение амплитуды уровня звукового давления между вызовами.

Команда Audio.record.START инициализирует процесс PEAK в первый раз. Если текущий процесс записи недоступен, PEAK возвращает -1.

##### Audio.record.start <fn\_sexp>{{{{{{{{{, <so\_nexp>}, <oF\_nexp>}, <eC\_nexp>},

**<sR\_nexp>}, <eBR\_nexp>}, <aC\_nexp>}, <mFS\_nexp>}, <lat\_nexp>}, <lon\_nexp>}** Start audio recording using the microphone as the audio source. The recording will be saved to the specified file. Recording will continue until the Audio.record.stop command is issued.

~~Some sources, codecs and bit-rates need a higher API level.~~

~~In this case your user should get a warning or better, your program handles the currentAndroid API level.~~

\*\*\*\*\*\*

Начните запись звука, используя микрофон в качестве источника звука. Запись будет сохранена в указанном файле. Запись будет продолжаться до тех пор, пока не будет введена команда Audio.record.stop.

Некоторые источники, кодеки и скорости передачи данных требуют более высокого уровня API.

В этом случае ваш пользователь должен получить предупреждение или лучше, ваша программа обрабатывает текущий уровень API Android.

Command OPTIONS:

|  |  |  |
| --- | --- | --- |
| Argument | Description | Defaults |
| <fn\_sexp> | File Name |  |
| <so\_nexp> | Source  1\* MICROPHONE;   1. CAMCORDER; 2. VOICE\_RECOGNITION; 3. VOICE\_COMMUNICATION~~(API level11)~~ | 0 |
| <oF\_nexp> | Output Format  1\* THREE\_GPP(.g3p);  2 MPEG\_4 (.mp4, .m4a) | 1 |
| <eC\_nexp> | Encoder  1\* AMR\_NB;   1. AMR\_WB~~(API level10)~~; 2. AAC~~(API level10)~~; 3. HE\_AAC~~(API level16)~~ 4. AAC\_ELD~~(API level16)~~ | 1 |
| <sR\_nexp> | Sampling Rate  in samples per second ~~(API level 8)~~ | 44100 |
| <eBR\_nexp> | Encoding Bit Rate  in bits per second ~~(API level 8)~~ | 96000 |
| <aC\_nexp> | Audio Channels  1 or 2\* ~~(API level 8)~~For one channel you get a mono output in a stereo file! | 2 |
| <mFS\_nexp> | Maximal File Size  in bytes, but the real size is more less! If the max. file size is reached, the recordingstops sooner orlater! | -1 |
| <lat\_nexp> | Latitude  -90 to 90 degree ~~(API level 14)~~ |  |
| <lon\_nexp> | Longitude  -180 to 180 degree~~(API level 14)~~ |  |

Attention:

Some results are device, sampling rate, encoding bit rate and/or codec depended.

\*\*\*\*\*\*

Внимание:

Некоторые результаты зависят от устройства, частоты дискретизации, скорости кодирования и / или кодека.

Example:

REM Start of BASIC! Program audio\_recording.bas Device dbp

Bundle.get dbp, "OS", myOs$ Print myOs$

DEBUG.ON

filename$ = "NewSound.g3p" filename$ = "NewSound.m4a"

! file.delete done, filename$ %If you use the RFO-BASIC you have to delete first!!!

! Command OPTIONS:

so = 1 %Source 0 Default; 1\* Mic; 5 CAMCORDER; 6 VOICE\_RECOGNITION;~

! (7 VOICE\_COMMUNICATION)

oF = 2 %OutputFormat 1\* THREE\_GPP(.g3p); 2 MPEG\_4 (.mp4, .m4a)

eC = 3 %Encoder 1\* AMR\_NB; (2 AMR\_WB; 3 AAC); (4 HE\_AAC; 5 AAC\_ELD)

sR = 48000 %(SamplingRate in samples per second) 44100\* eBR = 80000 %(EncodingBitRate in bits per second) 96000\*

aC = 2 %(AudioChannels 1 or 2\*) For one channel you get a mono output in a stereo file! mFS = 280000 %MaxFileSize in bytes, but the real size is more less! -1\*

! If the max. file size is reached, the recording stops sooner or later! lat = 50 %(Latitude -90 to 90 degree)

lon = 150 %(Longitude -180 to 180 degree)

AUDIO.RECORD.START filename$,so,oF,eC,sR,eBR,aC,mFS,lat,lon FOR i = 1 TO 20

PAUSE 1000

! With this PEAK command you get the max. amplitude between to calls.

! The START command init the PEAK process for the first time.

! If no current recording process available, PEAK returns -1. AUDIO.RECORD.PEAK m

FILE.SIZE FS, filename$

PRINT "Peak "; m; " File size "; fS NEXT i

FILE.SIZE fS,filename$

PRINT " Sound file size in byte:";fS AUDIO.RECORD.STOP

AUDIO.LOAD ptr, filename$ AUDIO.PLAY ptr

DO UNTIL 0 END

See also:

https://developer.android.com/guide/topics/media/media-formats.html

##### STT.listen {{<prompt\_sexp>}, <extras\_bundle\_nexp>}

Start the voice recognize process by displaying a "Speak Now" dialog box. The optional prompt string expression <prompt\_sexp> sets the dialog box’s prompt. If you do not provide the prompt parameter, the default prompt "BASIC! Speech To Text" is used.

Begin speaking when the dialog box appers.

The recognition will stop when there is a pause in the speaking.

**STT.results** should be executed next.

Note: **STT.listen** is *not*to be used in HTML mode.

The extras bundle <extras\_bundle\_nexp> controls more options:

\*\*\*\*\*\*

Запустите процесс распознавания голоса, открыв диалоговое окно «Говорить сейчас». Необязательное строковое выражение подсказки <prompt\_sexp> устанавливает подсказку диалогового окна. Если вы не укажете параметр подсказки, будет использоваться подсказка по умолчанию «BASIC! Speech To Text».

Начните говорить, когда появится диалоговое окно.

Распознавание прекратится, когда произойдет пауза в речи.

STT.results должен быть выполнен следующим.

Примечание. STT.listen нельзя использовать в режиме HTML.

Пакет extras <extras\_bundle\_nexp> контролирует дополнительные параметры:

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** |  |
| **\_MaxResults** | numeric | Optional limit on the maximum number of results to return. If omitted the recognizer will choose  how many results to return. |
| **\_Language** | String | Optional IETF language tag, for  example "en-US". This tag informs the recognizer. |
| **\_MinimumLength** | numeric | The minimum length of an utterance. We will not stop recording before this amount of time. Note that it is extremely rare you'd want to specify this value. If you don't have a very good reason to change these, you should leave them as they are per default. Note also that certain values may cause undesired or unexpected results - use judiciously! Additionally, depending on the recognizer  implementation, these values may have no effect. |
| **\_CompletSilence** | numeric | The amount of time that it should take after we stop hearing speech to consider the input complete.Note that it is extremely rare you'd want to specify this value. If you don't have a very good reason to change these, you should leave them as they are per default. Note also that certain values may cause undesired or unexpected results - use judiciously! Additionally, depending on the recognizerimplementation,  these values may have no effect. |

|  |  |  |
| --- | --- | --- |
| **\_PossiblyCompleteSilence** | numeric | The amount of time that it should take after we stop hearing speech to consider the input possibly complete. This is used to prevent the end pointer cutting off during very short mid-speech pauses. Note that it is extremely rare you'd want to specify this value. If you don't have a very good reason to change these, you should leave them as they are per default. Note also that certain values may cause undesired or unexpected results - use judiciously! Additionally, depending on the recognizerimplementation,  these values may have no effect. |
| **\_WebSearch** | 0\*   1. Web search and others 2. Only web search(numeric)   [Does not work properly with Google Assistant] | ~~Min. Jelly Bean 4.1 (API 16)~~Prompts the user for speech, send it through a speech recognizer, and either display a web search result or trigger another type of action based on the user'sspeech.  For security reasons the mode falls back to normal recognition, if the screen is off or the device is locked. If Google Assistant installed, it will be used. The problem is, Google  Assistant can not be closed by voice. |
| **\_HandsFree** | 0\* or 1 (numeric) | ~~Min. Jelly Bean 4.1 (API 16)~~Launches a start of a HandsFreeApp in a mode that will prompt the user for speech without requiring the user's visual attention or touch input. This activity may be launched while device is locked in a secure mode. Special care is be taken to ensure that the voice actions that are performed while hands free cannot compromise the device's security.  The called application can return a list, a String or a Number from type Double in the returned intent **extra** bundle. But your result is in each case only a list from type Sting.  If you create with Bundle.out your  result, you can use **only** a string or a number. |
| **\_Package** | String | ~~Min. Jelly Bean 4.1 (API 16)~~Only in conjunction with  \_HandsFree.  Package name of the whished App. Example: "com.rfo.basicOli" |

|  |  |  |
| --- | --- | --- |
| **\_Component** | String | ~~Min. Jelly Bean 4.1 (API 16)~~Only in conjunction with  \_HandsFree and \_Package. Example: "com.rfo.basicOli.Basic" |
| **\_Data** | String as URL | ~~Min. Jelly Bean 4.1 (API 16)~~Only in conjunction with  \_HandsFree and \_Package. Example:  FILE.ROOT fp$, "\_External" filePath$ = "file://" + fp$ + "rfo-basic/ source"  mUrl$ = filePath$ + "/" + "freeHandsDemo.bas" |

If the following paragraph in the AndroidManifest.xml is accessible, the program is also a possible hands free receiver like Google Assistant. But the advantage is, the program can be closed by voice. If you create a normal APK from your program, the paragraph should be deleted.

\*\*\*\*\*\*

Если доступен следующий абзац в AndroidManifest.xml, программа также является возможным приемником громкой связи, таким как Google Assistant. Но преимущество в том, что программу можно закрыть голосом. Если вы создаете нормальный APK из своей программы, абзац должен быть удален.

<!-- vv 2017-11-25gt Comment it out if it should **not** be a hands free receiver, too. -->

<intent-filter >

<action android:name="android.speech.action.VOICE\_SEARCH\_HANDS\_FREE" />

<category android:name="android.intent.category.DEFAULT" />

</intent-filter>

<!-- ^^ 2017-11-25gt -->

##### TTS.kill

TTS.kill stops the TTS process without waiting for finish. With this command you have to code carefully! You have to be guaranteed, that all following commands including until TTS.STOP will not be executed. Following **TTS.kill**, if you want to run **TTS.speak** or **TTS.speak.toFile** again, you will have to run **TTS.init** again.

\*\*\*\*\*\*

TTS.kill останавливает процесс TTS, не дожидаясь завершения. С помощью этой команды вы должны тщательно кодировать! Вы должны быть гарантированы, что все последующие команды, в том числе и до TTS.STOP, не будут выполнены. После TTS.kill, если вы хотите снова запустить TTS.speak или TTS.speak.toFile, вам придется снова запустить TTS.init.

# Information About Your Android Device

## Device Command Overview

You can get information about your Android device with the **Device** command:

* + - The Device Brand, Device Model, Device Type, andOS
    - The Language andLocale
    - The PhoneType, PhoneNumber, andDeviceID
    - The SN, MCC/MNC, and Network Provider stored on the SIM, if there isone.

The**Device**commandhastwoformsthatdifferonlyinthetypeoftheparameter,whichdetermines the format of the returned data. Both forms return the same information, as shown in thistable:

|  |  |  |  |
| --- | --- | --- | --- |
| **Key** | **Values** | **Meaning** | **Example**  (from emulator) |
| **Brand** | Any string | Brand name assigned by device manufacturer | generic |
| **Model** | Any string | Model identifier assigned by device manufacturer | sdk |
| **Device** | Any string | Device identifier assigned by device manufacturer | generic |
| **Product** | Any string | Product identifier assigned by device manufacturer | sdk |
| **OS** | OS Version | Android operating system version number | 4.1.2 |
| **Language** | Language name | Default language of this device | English |
| **Locale** | Locale code | Default locale code, typically language and country | en\_US |
| **PhoneType** | **GSM**,**CDMA**,  **SIP**, or**None**  **or Not available** | Type of phone radio in this device | GSM |
| **PhoneNumber** | String of digits or **Not**  **available** | Phone number registered to this device, if any | 15555215554 |
| **DeviceID** | String of digits  or **Not available** | The unique device ID, such as the IMEI until Android 8.1 | 000000000000000 |
| **SIM SN** | String of digits or **Not**  **available** | Serial number of the SIM card,  if one is present and it is accessible | 890141032111185  10720 |
| **SIM MCC/MNC** | String of digits  or **Not available** | The "numeric name" of the provider of the SIM, if present and accessible | 310260 |
| **SIM Provider** | Name string or  **Not available** | The name of the provider of the SIM, if present and accessible | Android |

The last six items access your device’s telephony system and SIM card. If your device has no telephone, or BASIC! does not have permission to access them, the fields are set to neutral values: "None", "Not available", or a string of "0" characters.

In addition, there are convenience commands to retrieve only the Locale or the Language. The information returned by Device is static. To get dynamic information, use the Phone.Info command.

##### Device.os <os\_nvar>{{{{{, <release\_svar>} , <codename\_svar>} ,

**<incremental\_svar>} , <security\_svar>} ,<base\_os\_svar>}**

Returns the os as a number. For this command you need **no Phone permissions**. Options:

<release\_svar> returns

the user-visible version string. E.g., "1.0" or "3.4b5".

<codename\_svar> returns

the current development codename, or the string "REL" if this is a release build.

<incremental\_svar> returns

the internal value used by the underlying source control to represent this build. E.g., a perforce changelist number or a git hash.

<security\_svar> returns

the user-visible security patch level. (Needs API level 23.)

<base\_os\_svar> returns

the base OS build the product is based on. (Needs API level 23.)

##### Device.USB <bundlePointer\_nvar>

Returns the parameters of plugged in USB devices.

Following keys are supported "\_DeviceId", "\_VendorId", "\_ProductId", "\_DeviceClass", "\_InterfaceClass0", "\_DeviceSubclassId", "\_DeviceName", "\_SerialNumber", "\_ManufacturerName", "\_ProductName" and "\_AllAsString".

The keys "\_ManufacturerName" and "\_ProductName" are available on Android 5+. Example:

DEVICE.USB bp BUNDLE.KEYS bp, lp LIST.SIZE lp,n

FOR i = 1 TOn

BUNDLE.GB bp, int$(i), pUSB BUNDLE.GET pUSB, "\_AllAsString", aAS$ PRINT aAS$

NEXT

##### Device.get.brightness <brigth\_nvar>

Returns the brightness in the range from 0 to 255. Needs WRITE\_SETTINGS permission.

\*\*\*\*\*\*

Возвращает яркость в диапазоне от 0 до 255. Требуется разрешение WRITE\_SETTINGS.

##### Device.set.brightness <brigth\_nvar>

Sets brightness mode to manual and sets the brightness in the range from 0 to 255. Needs WRITE\_SETTINGS permission.

\*\*\*\*\*\*

Устанавливает ручной режим яркости и устанавливает яркость в диапазоне от 0 до 255. Требуется разрешение WRITE\_SETTINGS.

##### Device.auto.brightness <bright\_nexp>

Sets brightness mode to automatic. Needs WRITE\_SETTINGS permission.

\*\*\*\*\*\*

Устанавливает автоматический режим яркости. Требуется разрешение WRITE\_SETTINGS.

**HTML Commands**

##### Html.open {<Title\_nexp> {, <Orientation\_nexp>}}

This command must be executed before using the HTML interface.

The Statusbar will be shown on the Web Screen if the <Title\_nexp> is 1 ~~true (not zero)~~. If the <Title\_nexp> is not present, the Status bar will not be shown.

If <Title\_nexp> is **negative**, Html.open expects a layout bundle defined by items of the following table.

The orientation upon opening the HTML screen will be determined by the

<Orientation\_nexp> value. <Orientation\_nexp> values are the same as values for the Html.orientation command (see below). If the <Orientation\_nexp> is not present, the default orientation is determined by the orientation of the device.

Both <Title\_nexp> and <Orientation\_nexp> are optional; however, a <Title\_nexp> must be present in order to specify an <Orientation\_nexp>.

Executing a second HTML.OPEN before executing HTML.CLOSE will generate a run-time error.

\*\*\*\*\*\*

Эта команда должна быть выполнена перед использованием интерфейса HTML.

Строка состояния будет отображаться на веб-экране, если <Title\_nexp> равно 1 true (не ноль). Если <Title\_nexp> отсутствует, строка состояния не будет отображаться.

Если <Title\_nexp> отрицательно, Html.open ожидает пакет макета, определенный элементами следующей таблицы.

Ориентация при открытии экрана HTML будет определяться

Значение <Orientation\_nexp>. Значения <Orientation\_nexp> совпадают со значениями для команды Html.orientation (см. Ниже). Если <Orientation\_nexp> отсутствует, ориентация по умолчанию определяется ориентацией устройства.

И <Title\_nexp>, и <Orientation\_nexp> являются необязательными; однако для указания <Orientation\_nexp> должен присутствовать <Title\_nexp>.

Выполнение второго HTML.OPEN перед выполнением HTML.CLOSE приведет к ошибке во время выполнения.

The optional options bundle (-) <Title\_nexp> controls the layouts of theAction and Navigation bars:

|  |  |  |
| --- | --- | --- |
| **Table of Layout Control Options** | | |
| **Key** | **Value** | **Description** |
| **\_Title** | String by default | Set the action bar's title. |
| **\_Subtitle** | String by default | Set the action bar's subtitle. |
| **\_TitleShow** | 0 or 1 (numeric) | If 1 (default)  Show the Action bar if it is not currently showing. It will resize application content to fit the new space available.  If 0  Hide the Actionbar if it is currently showing. It will resize application content to fit the new space available. |
| **\_TitleIcon** | Icon file path | Add a large icon to the notification content view.  <http://romannurik.github.io/>AndroidAssetStudio/index.html |
| **\_TitleHomeEnabled** | 0 or 1 (numeric) | Set whether to include the application home accordance in the action bar. Home is presented as an activityicon.  Have to be 1 if you want to show the icon.  Have to be 0 if you want to hide the icon.  The default setting is API dependent. |
| **\_TitleBackground** | Background file path |  |

|  |  |  |
| --- | --- | --- |
| **\_TitleHtml** | 0 or 1 (numeric) | Returns displayable styled text from the provided HTML string. But not all tags are supported.    Uses parts of TagSoup library to handle real HTML, including all of the brokenness found in the wild.  <b>  <big>  <font size="..." color="..." face="...">  <h1>, <h2>, <h3>, <h4>, <h5>,  <h6>  <i>  <small>  <strike>? < A.7  <strong>  <sub>  <sup>  <tt>?  <u>  **Replace Space with &#160,**  **& with &amp,**  **< with &lt,**  **> with &gt, " with &quot if necessary.**  Usable for Title andSubtitle.  Keep in mind that theActionbar height will not beexpanded. |
| **\_ShowStatusbar** | 0, 1 or 2 (numeric) | If 1 (default)  The Status bar will be displayed. If 2  The Status bar will be transparent displayed.  Min. Lollipop 5.0 (API 21)  If 0  The Status bar will be hidden to the background.  Min. Nougat 7.0 (API 24)  Will be switched to option 2 or 1 if the current API level is lower. |
| **\_StatusbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |

|  |  |  |
| --- | --- | --- |
| **\_StatusbarLight** | 0 or 1 (numeric) | If 0 (default)  The Status bar background is dark. In this case the **bar content** will be **light**.  If 1  The Status bar background is light. In this case the **bar content** will be **dark**.  Min. Lollipop 5.0 (API 21) |
| **\_ShowNavigationbar** | 0, 1 or 2 (numeric) | If 1 (default)  The Navigation bar will be displayed.  If 2  The Navigation bar will be transparent displayed.  Min. Lollipop 5.0 (API 21)  If 0  The Navigation bar will be hidden to the background.  Min. Nougat 7.0 (API 24)  Will be switched to option 2 or 1 if the current API level is lower. |
| **\_NavigationbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |
| **\_NavigationbarLight** | 0 or 1 (numeric) | If 0 (default)  The Navigation bar background is dark.  In this case the **bar content** will be  **light**. If 1  The Navigation bar background is light.  In this case the **bar content** will be  **dark**.  Min. Lollipop 5.0 (API 21) |
| **\_Menu** | Menu Bundle Pointer | Creates menu entries.  A successful selection will be returned as a human readable JSONstring.  See the example at Console.title for moredetails. |

See also: GR.open, Console.title, Select

##### Html.screenshot <filename\_sexp>

##### Html.to.pdf <filename\_sexp> {{{{, <paperformat\_sexp>}, <orientation\_sexp>},

**<resolution\_nexp>}, <color\_sexp>}**

Saves the current html content to a file. The default path is "<pref base drive>/rfo-basic/data/".

The chosen paper format will be set by <paperformat\_sexp> default is "ISO\_A4".

The optional <orientation\_sexp> defines the sheet orientation. "L" is landscape and "P" the default portrait. The resolution can be set by <resolution\_nexp> in dots per inch. The default setting is 600 dpi. By default <color\_sexp> is set to "C" for colored output. "M" saves the content monochrome.

##### Note that to override, you have to delete the old PDF file first.

##### \*\*\*\*\*

##### Сохраняет текущее HTML-содержимое в файл. Путь по умолчанию: «<pref base drive> / rfo-basic / data /».

##### Выбранный формат бумаги будет установлен <paperformat\_sexp> по умолчанию «ISO\_A4».

##### Необязательный <direction\_sexp> определяет ориентацию листа. «L» - пейзаж, а «P» - портрет по умолчанию. Разрешение может быть установлено с помощью <resolution\_nexp> в точках на дюйм. Значение по умолчанию составляет 600 точек на дюйм. По умолчанию для <color\_sexp> установлено значение «C» для цветного вывода. «М» сохраняет содержимое в монохромном режиме.

##### Обратите внимание, что для переопределения вы должны сначала удалить старый файл PDF.

Example:

HTML.OPEN 0, -1

!html.load.url "htmlDemo1.html" mM$ = "Hello<br>World" HTML.LOAD.STRING mM$ HTML.PAPERFORMATS pfb

BUNDLE.KEYS pfb, pfl

Dialog.select sel, pfl, "Choose a paper format"

List.get pfl, sel, pf$ fileName$ = "html.pdf" FILE.EXISTS fOk,fileName$

IF fOk THEN FILE.DELETE dOk, fileName$ HTML.TO.PDF fileName$, pf$ , "P" HTML.CLOSE

##### Html.paperformats <bundlePointer\_nexp>

Returns possible paper formats as a bundle. The bundle key is also the paper format key name. The bundle entry returns more information about the specified format as a delimited string.

\*\*\*\*\*

Возвращает возможные форматы бумаги в виде пакета. Ключ пакета также является именем ключа формата бумаги. Запись пакета возвращает дополнительную информацию об указанном формате в виде строки с разделителями.

##### OnHtmlReturn:

Interrupt label that traps if HTML data arrived. BASIC! executes the statements following the **OnHtmlReturn:** label until it reaches a **Html.onHtmlReturn.resume**.

Метка прерывания, которая перехватывает при поступлении HTML-данных. BASIC! выполняет операторы после метки OnHtmlReturn: до тех пор, пока не достигнет Html.onHtmlReturn.resume.

See also Html.get.datalink, MenuItem.get.datalink

##### Html.onHtmlReturn.resume

Resumes execution at the point in the BASIC! program where the **OnHtmlReturn:**

interrupt occurred.

\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, где OnHtmlReturn:

произошло прерывание.

##### IS\_HTML <status\_nvar>

IS\_HTML returns the HTML mode status. If <status\_nvar> is 1 the HTML mode is enabled. Is 0 returned the HTML mode is not open.

\*\*\*\*\*

IS\_HTML возвращает статус режима HTML. Если <status\_nvar> равен 1, режим HTML включен. Возвращено 0, режим HTML не открыт.

See also Console.Save

### BACKGROUND()

A running BASIC! program continues to run when the HOME key is tapped. This is called running in the Background. When not in the Background mode, BASIC! is in the Foreground mode. BASIC! exits the Background mode and enters the Foreground mode when the BASIC! icon on the home screen is tapped.

Sometimes a BASIC! programmer wants to know if the program is running in the Background. One reason for this might be to stop music playing while in the Background mode.

The **BACKGROUND()** function returns true (1) if the program is running in the background. It returns false (0) if the program is not running in the background. It returns (3) if the display screen is **off**.

~~Min. Jelly Bean 4.1 (API16):~~

It returns (4) if the device is **locked**.

It returns (5) if the display screen is **on** and the device is **locked**.

If you want to be able to detect Background mode while Graphics is open, you must not call **Gr.render** while in the Background mode. Doing so will cause the program to stop running until the Foreground mode is re-entered. Use the following code line for all **Gr.render** commands:

Беговая бейсик! Программа продолжает работать, когда нажата клавиша HOME. Это называется бегом в фоновом режиме. Когда не в фоновом режиме, бейсик! находится в режиме переднего плана. BASIC! Выход из фонового режима и переход в режим переднего плана, когда BASIC! значок на главном экране коснулся.

Иногда бейсик! программист хочет знать, работает ли программа в фоновом режиме. Одной из причин этого может быть прекращение воспроизведения музыки в фоновом режиме.

Функция BACKGROUND () возвращает true (1), если программа работает в фоновом режиме. Возвращает false (0), если программа не работает в фоновом режиме. Возвращает (3), если экран дисплея выключен.

Минимум Jelly Bean 4.1 (API16):

Возвращает (4), если устройство заблокировано.

Возвращает (5), если экран дисплея включен и устройство заблокировано.

Если вы хотите иметь возможность определять фоновый режим, когда графика открыта, вы не должны вызывать Gr.render в фоновом режиме. Это приведет к остановке работы программы до повторного входа в режим переднего плана. Используйте следующую строку кода для всех команд Gr.render:

IF !BACKGROUND() THEN GR.RENDER

### HYPOT(<nexp\_x>, <nexp\_y>)

Returns SQR(*x*2+*y*2) without intermediate overflow or underflow.

### ATAN(<nexp>)

Returns the arc tangent of the angle <nexp>, in the range of -pi/2 through pi/2. The units of the angle are radians.

Returns the closest double approximation of the arc tangent of the argument within the range [-pi/2..pi/2].

Special cases:

•atan(+0.0) = +0.0

•atan(-0.0) = -0.0

•atan(+infinity) = +pi/2

•atan(-infinity) = -pi/2

•atan(NaN) = NaN

### ATAN2(<nexp\_y>, <nexp\_x>)

Returns the angle *theta* from the conversion of rectangular coordinates (x, y) to polar coordinates (r,*theta*), in the range of -pi through pi. (Please **note the order** of the parameters in this function.)

Special cases:

•atan2((anything), NaN ) = NaN;

•atan2(NaN , (anything) ) = NaN;

•atan2(+0.0, +(anything but NaN)) = +0.0

•atan2(-0.0, +(anything but NaN)) = -0.0

•atan2(+0.0, -(anything but NaN)) = +pi

•atan2(-0.0, -(anything but NaN)) = -pi

•atan2(+(anything but 0 and NaN), 0) = +pi/2

•atan2(-(anything but 0 and NaN), 0) = -pi/2

•atan2(+(anything but infinity and NaN), +infinity) = +0.0

•atan2(-(anything but infinity and NaN), +infinity) = -0.0

•atan2(+(anything but infinity and NaN), -infinity) = +pi

•atan2(-(anything but infinity and NaN), -infinity) = -pi

•atan2(+infinity, +infinity )=+pi/4 **+ 45° 1.Quadrant**

•atan2(-infinity, +infinity )=-pi/4 **- 45° 4.Quadrant**

•atan2(+infinity, -infinity )=+3pi/4 **+ 135° 2.Quadrant**

•atan2(-infinity, -infinity )=-3pi/4 **- 135° 3.Quadrant**

•atan2(+infinity, (anything but,0, NaN, and infinity)) =+pi/2

•atan2(-infinity, (anything but,0, NaN, and infinity)) = -pi/2

### CLAMP(<value\_nexp>, <min\_nexp>, <max\_nexp>)

This method takes the numerical value <value\_nexp> and ensures it fits in a given numerical range. If the number is smaller than the minimum <min\_nexp> required by the range, then the minimum <min\_nexp> of the range will be returned. If the number is higher than the maximum <max\_nexp> allowed by the range then the maximum <max\_nexp> of the range will be returned.

It is also a one liner, max(mMin, min(mMax, mVal))

If you have a lot of values use List.join, like the line above. It is much faster.

\*\*\*\*\*

Этот метод принимает числовое значение <value\_nexp> и обеспечивает его соответствие заданному числовому диапазону. Если число меньше минимального <min\_nexp>, требуемого диапазоном, то будет возвращено минимальное <min\_nexp> диапазона. Если число превышает максимальный <max\_nexp>, разрешенный диапазоном, то будет возвращено максимальное значение <max\_nexp> диапазона.

Это также один вкладыш, макс (мМин, мин (мМакс, мВал))

Если у вас много значений, используйте List.join, как в строке выше. Это намного быстрее.

Example:

x = 20

grid\_size = 100

grid\_width = 1000

gx = CLAMP(x/grid\_size, 1, grid\_width)

! gx = MAX(mMin, MIN(mMax, mVal))

gx = MAX(1, MIN(grid\_width, x/grid\_size))

LIST.CREATE n, mListX

LIST.ADD mListX, ***x***

LIST.ADD mListX, 40, 60, 80, 100, 120 LIST.CREATE n, mListR1

LIST.JOIN mListR1, mListX, STR$(***grid\_size***) , "", "\_/8"

! DEBUG.DUMP.LIST mListR1 LIST.CREATE n, mListR2

LIST.JOIN mListR2, STR$(***grid\_width***), mListR1, "", "\_min"

! DEBUG.DUMP.LIST mListR2 LIST.CREATE n, mListR3

LIST.JOIN mListR3, STR$(***1***), mListR2, "", "\_max" DEBUG.DUMP.LIST mListR3

See also List.Join, Array.Math

##### USING$({<locale\_sexp>} , <format\_sexp> { , <exp>}...)

Returns a string, using the locale and format expressions to format the expression list. This function gives BASIC! programs access to the Formatter class of the Android platform. You can find full documentation here: <http://developer.android.com/reference/java/util/Formatter.html>.

The <locale\_sexp> is a string that tells the formatter to use the formatting conventions of a specific language and region or country. For example, "en\_US" specifies AmericanEnglish conventions.

The <format\_sexp> is a string that contains *format specifiers*, like "%d" or "%7.2f," that tell the formatter what to do with the expressions that follow.

The format string is followed by a list of zero or more expressions. Most format specifiers take one argument from the list, in order. If you don’t provide as many arguments as your format string needs, you will get a detailed Java error message.

Each expression must also match the type of the corresponding format specifier. If youtry to apply a string format specifier, like "%-7s", to a number, or a floating point specifier, like "%5.2f" to a string, you will get a Java errormessage.

##### Locale expression

The **USING$()** function can localize the output string based on language and region. The locale specifies the language and region with standardized codes. The <locale\_sexp> is a string containing zero or more codes separated by underscores.

The function accepts up to three codes. The first must be a language code, such as "en", "de", or "ja". The second must be a region or country code, such as "FR", "US", or "IN".

Some language and country combinations can accept a third code, called the "variant code".

The function also accepts the standard three-letter codes and numeric codes for country or region. For example, "fr\_FR", "fr\_FRA", and "fr\_250" are all equivalent.

If you want to use the default locale of your Android device, make the <locale\_exp> an empty string (""), or leave it out altogether. If you leave it out, you must keep the comma: **USING$(, "%f", x)**

If you make a mistake in the <locale\_sexp>, you may get an obscure Java error message, but more likely your locale will be ignored, and your string will be formatted using the default locale of your device.

Android devices do not support all possible locales. If you specify a valid locale that your device does not understand, your string will be formatted using the default locale.

##### Format expression

If you are familiar with the *printf* functions of C/C++, Perl, or other languages, you will recognize most of format specifiers of this function. The format expression is exactly the same as format string of the Java *Formatter* class, or the *format(String, Object…)* method of the Java *String*, with two exceptions: Boolean format specifiers are not supported, and hex hash specifiers are limited to numeric and string types.

If you have not programmed in one of those other languages, this will be your introduction to a powerful tool for formatting text.

A format expression is a string with embedded format specifiers. Anything that is not a format specifier is copied literally to the function output string. Each embedded format specifier is replaced with the value of an expression from the list, formatted according to the specifier. For example:

PRINT USING$("","Pi is approximately%f.",PI()) functioncall

Pi isapproximately3.141593. printed output for Englishlocale

The <locale\_exp> is "", meaning "use my default locale".

The <format\_exp>, "Pi is approximately %f", has one format specifier, "%f". "%f" means, "use the default decimal floating point output format".

The expression list has one item, the math function **PI()**.

In the output, "%f" is replaced by the value of the ~~the~~**PI()** function. Your output may be different if your locale language is not English.

###### *Format Specifiers*

Here is a brief summary of the available format specifiers:

|  |  |  |
| --- | --- | --- |
| For this type of data | Use these formats | Comments |
| String | %s %S | %S forces output to upper-case |
| Number | %f %e %E %g %G  %a %A | Standard BASIC! numbers are floating point Use %f for decimal output: "1234.567"  Use %e or %E for exponential notation: "1.234e+03"  %E writes upper-case: "1.234E+03"  %g (%G) lets the system choose %f or %e (%E)  %a and %A are "hexadecimal floating point" |
| Integer | %d %o %x %X | USING$ can use some math functions as integers  Use %d for decimal, %o for octal, %x %X for hex  %x writes lower-case abcdef, %X writes upper-case |
| Special integer | %c %C %t | These specifiers can operate on an integer  %c %C output a character, %C writes upper- case  %t represents a family of time format specifiers |
| None | %% %n | These specifiers do not read the expression list  %% writes a single "%" to the output  %n writes a newline, exactly the same as \n |

For more information about %a and %A, see the Android documentation linked above.

Android’s %b and %B are not supported because BASIC! has no Boolean type. Android’s %h and %H hash code specifiers are limited to strings and numbers in BASIC!.

For an explanation of **USING$()** with integer format specifiers, see below.

There is a whole family of time format specifiers: **%t<x>** where **<x>** is another letter. They operate on an integer, which they interpret as the number of milliseconds since the beginning of January 1, 1970, UTC (the "epoch"). You can apply time format specifiers to the output of the **TIME()** functions. Note, however, that the **%t** time specifiers use your local timezone, not the **TimeZone.set** value.

There are more than 30 time format specifiers. Afew examples appear below, but to get the full list you should read the Android documentation linked above.

PRINT USING$("", "The time is: %tI:%<tM:%<tS %<Tp", time()) % the hard way PRINT USING$("", "The time is:%tr",time()) % same thing! 02:27:16PM example of printedoutput

t = TIME(2001, 2, 3, 4,5,6) % set 2001/02/03 04:05:06, localtimezone PRINT USING$("sv","%tA",t) % day in Swedish, prints"lördag"

PRINT USING$("es","%tB",t) % month in Spanish, prints "febrero" PRINT USING$("", "%tY/%tm/%td", t ,t,t) % prints"2001/02/03"

PRINT USING$( ,"%tY/%<tm/%<td",t) % prints "2001/02/03" PRINT USING$("en\_GB","%tH:%<tM:%<tS",t) % prints "04:05:06" PRINT USING$("in\_IN","%tT",t) % prints"04:05:06"

Note: Date and time are printed for your local timezone, regardless of either the TIMEZONE.SET setting or the locale parameter. Try the same set of examples with **TIMEZONE.SET "UTC"**. Unless that is your local timezone, a different hour and perhaps even a different day will be displayed.

###### *Optional Modifiers*

The format specifiers can be used exactly as shown in the table. They have default settings that control how they behave. You can control the settings yourself, fine-tuning the behavior to suit your needs.

You can modify the format specifiers with *index*, *flags*, *width*, and *precision*, as shown in this example:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| "%3$-,15.4f" | | | | | | | |
| "  % | 3$ | -, | 15 | . | 4 | f | " |
|  | <inde x> | <flags  > | <width  > |  | <precisio n> | <specifie r> |  |

###### *Index*

Normally the format specifiers are applied to the arguments in order. You can change the order with an argument index. An index a number followed by a **$** character. The argument index **3$** specifies the third argument in the list.

PRINT USING$("", "%3$s %s %s", "a", "b", "c") % prints "c ab"

The special argument index "<" lets you reuse anargument.

PRINT USING$("", "%o %<d %<h",~~int(~~64)) % prints "100 6440"

In the last example, there is only one argument, but three format specifiers. This is not an error because the argument is reused.

###### *Flags*

There are six flags:

|  |  |
| --- | --- |
| - | left-justify; if no flag, right-justify |
| + | always show sign; if no flag, show "-" but do not show "+" |
| 0 | pad numbers with leading zeros; if no flag pad with spaces |
| , | use grouping separators for large numbers |
| ( | put parentheses around negative values |
| # | alternate notation (leading 0 for octal, leading 0x for hexadecimal) |

###### *Width*

The **width** control sets the minimum number of characters to print. If the value to format is longer than the width setting, the entire value is printed (unless it would exceed the **precision** setting). If the value to format is shorter than the **width** setting, it is padded to fill the width. By default, it is padded with spaces on the left, but you change this with the "-" and "0" flags.

###### *Precision*

The **precision** control means different things to different data types.

For floating point numbers, **precision** specifies the number of characters to print after the decimal point, padding with trailing zeros if needed.

For string values, it specifies the maximum number of characters to print. If **precision** is less than **width**, only **precision** characters are printed.

"%4s","foo"  "foo"

"%-4s","foo"  "foo"

"%4.2s","foo"  "fo"

The **precision** control is not valid for other types. In the example above, **%-,15.4f**:

The **flags** "-" and "," mean "left-justify the output" and "use a thousands separator". The **width** is 15, meaning the output is to be at least 15 characters wide.

The **precision** is 4, so there will be exactly four digits after the decimal point.

The whole format specifier means, "format a floating point number (%f) left-justified ("-") in a space 15 characters wide, with 4 characters after the decimal point, with a thousands separator (",")".

The characters used for the decimal point and the thousands separator depend on the locale:

"1,234.5678 " for locale "en" "1234,5678 " for locale "fr" "1.234,5678 " for locale"it"

##### Integer values

BASIC! has only double-precision floating point numbers. It does not have an integer type. The **USING$()** function supports format specifiers ("%d", "%t", "%x", "%X") that apply only to integer values. Is converting from type Double necessary, it will be done automatically. **~~USING$()~~** ~~has a special relationship with the math functions that intrinsically produceinteger results. BASIC! converts the output of these functions to floating point, for storagein numeric variables, but~~ **~~USING$()~~** ~~can get the original integer values. For example:~~

~~PRINT USING$("","%d",123) % ERROR!PRINT USING$("","%d",INT(123)) % Noerror~~

The functions that can produce integer values for **USING$()** are:

INT() BIN() OCT() HEX() CEIL()FLOOR()

ASCII() UCODE() BAND() BOR() BXOR() SHIFT()TIME()

### FORMAT\_USING$(<locale\_sexp>, <format\_sexp> { , <exp>}...)

Alias for **USING$()**. You can use the two equivalent functions to make your code easier to read. For example:

string$ = FORMAT\_USING("", "pi is not %d", pi()) Print USING$("en\_US", "Balance: $%8.2f", balance)

### HEX$(<nexp>|<color\_sexp>)

Returns a string representing the hexadecimal representation of the numeric expression.

If the alternative <color\_sexp> is specified, the function returns the color by the (#) hex color notation like (#)ff00ff00 (\_Lime). The source specified by <color\_sexp> can be one of these options:

{Alpha,}Red,Green,Blue % All these members in a range from 0 to 255 (comma delimited string)

or

\_{Alpha,}ColorName ({comma delim.} string) or

#{hn}hnhnhn (hex. String)

Example:

hColor$ = **"#" +** HEX$("0,0,255") % (\_Blue) returns "#ff0000ff"

### SPC$(<nexp>{, <sexp>})

Returns a string with <nexp> spaces.

If <sexp> is set, the function returns <nexp> times the into <sexp> defined string. For example, CHR$(9) can insert tabs instead of spaces.

\*\*\*\*\*

Возвращает строку с пробелами <nexp>.

Если установлено значение <sexp>, функция возвращает <nexp> раз в определенную строку <sexp>. Например, CHR $ (9) может вставлять табуляции вместо пробелов.

### REVERSE$(<sexp>)

Returns a copy of <sexp> with the order of the characters reversed. Example:

\*\*\*\*\*

Возвращает копию <sexp> с обратным порядком символов. Пример:

PRINT Reverse$("Was it a cat I saw?") % Returns "?was I tac a ti saW"

### REPLACE$(<sexp>, <find\_sexp>, <replace\_sexp>{, <mode\_sexp>} )

Returns <sexp> with all instances of <find\_sexp> replaced with <replace\_sexp>. Options of <mode\_sexp>:

"\_Default"

replace **without** regular expressions how described above(default) "\_RegEx\_All"

replace **with** regular expressions **all** instances of <argument\_sexp> "\_RegEx\_First"

replace **with** regular expressions **first** instance of <argument\_sexp> "\_RegEx\_All\_IgnoreCase"

replace **with** regular expressions **all** instances of <argument\_sexp>, but ignore case "\_RegEx\_First\_IgnoreCase"

replace **with** regular expressions **first** instance of <argument\_sexp>, but ignore case

\*\*\*\*\*

Возвращает <sexp> со всеми экземплярами <find\_sexp>, замененными на <replace\_sexp>. Опции <mode\_sexp>:

"\_По умолчанию"

заменить без регулярных выражений, как описано выше (по умолчанию) "\_RegEx\_All"

заменить регулярными выражениями все экземпляры <arguments\_sexp> "\_RegEx\_First"

заменить регулярным выражением первый экземпляр <аргумент\_sexp> "\_RegEx\_All\_IgnoreCase"

заменить регулярными выражениями все экземпляры <arguments\_sexp>, но игнорировать регистр "\_RegEx\_First\_IgnoreCase"

заменить регулярными выражениями первый экземпляр <аргумент\_sexp>, но игнорировать регистр

Examples:

Result$ = Replace$("abcdefghijklmnabc", "abc", "rst", "\_RegEx\_All")

!-- returns "rstdefghijklmnrst" PRINT Result$

Result$ = Replace$("abc:defghi,jklmn;abc", "[,;:]", "|", "\_RegEx\_All")

!-- returns "abc|defghi|jklmn|abc" PRINT Result$

Result$ = Replace$("abcdefghijklmnabc", "abc", "rst", "\_RegEx\_First")

!-- returns "rstdefghijklmnabc" PRINT Result$

### Split <result\_array$[]>, <sexp> {, <test\_sexp>}Split.all <result\_array$[]>, <sexp> {, <test\_sexp>}

Splits the source string <sexp> into multiple strings and place them into <result\_array$[]>. The array is specified without an index. If the array exists, it is overwritten. Otherwise a new array is created. The result is always a one-dimensional array.

The string is split at each location where <test\_sexp> occurs. The <test\_sexp> occurrences are removed from the result strings. The <text\_sexp> parameter is optional; if it is not given, the string is split on whitespace. Omitting the parameter is equivalent to specifying "\\s+".

\*\*\*\*\*

Разбивает исходную строку <sexp> на несколько строк и помещает их в <result\_array $ []>. Массив указывается без индекса. Если массив существует, он перезаписывается. В противном случае создается новый массив. Результатом всегда является одномерный массив.

Строка разбивается в каждом месте, где находится <test\_sexp>. Вхождения <test\_sexp> удаляются из строк результата. Параметр <text\_sexp> является необязательным; если он не задан, строка разбивается на пробелы. Пропуск параметра эквивалентен указанию "\\ s +"

If the beginning of the source string matches the test string, the first element of the result array will be an empty string. This differs from the **WORD$()** function, which strips leading and trailing occurrences of the test string from the source string before splitting.

Two adjacent occurrences of the test expression in the source expression result in an empty element somewhere in the result array. The **Split** command discards these empty strings if they occur at the end of the result array. To keep these trailing empty strings, use the **Split.all** command.

\*\*\*\*\*

Если начало исходной строки совпадает с тестовой строкой, первый элемент массива результатов будет пустой строкой. Это отличается от функции WORD $ (), которая удаляет начальные и конечные вхождения тестовой строки из исходной строки перед разбиением.

Два соседних вхождения тестового выражения в исходном выражении приводят к пустому элементу где-то в массиве результатов. Команда Split отбрасывает эти пустые строки, если они встречаются в конце массива результатов. Чтобы сохранить эти завершающие пустые строки, используйте команду Split.all.

Example:

string$ = "a:b:c:d" delimiter$ = ":"

SPLIT result$[], string$, delimiter$

ARRAY.LENGTH length, result$[] FOR i = 1 TO length

PRINT result$[i] + " "; NEXT i

PRINT ""

Prints: a b c d

Note: The <test\_sexp> is actually a Regular Expression. If you are not getting the results that you expect from the <test\_sexp> then you should examine the rules for Regular Expressions at:

\*\*\*\*\*

Примечание. <Test\_sexp> на самом деле является регулярным выражением. Если вы не получаете ожидаемых результатов от <test\_sexp>, вам следует изучить правила для регулярных выражений по адресу

<http://developer.android.com/reference/java/util/regex/Pattern.html>For the Regular Expression "\[afl\]" you need a (Java) string "\\[afl\\]"

A good address to verify is: <http://www.regexplanet.com/advanced/java/index.html>

If you miss a **TALLY** command, do so:

FN.DEF TALLY( mString$, regExp$)

SPLIT.ALL result\_array$[], mString$, regExp$ ARRAY.LENGTH count, result\_array$[] ARRAY.DELETE result\_array$[]

FN.RTN count - 1 FN.END

Count = TALLY("abcdefghijklmnabc", "abc") %-- returns 2 PRINTCount

Count = TALLY("abcdefghijklmnabc", "\\[afl\\]") %-- returns 0 PRINTCount

Count = TALLY("abcdefghijklmnabc", "[afl]") %-- returns 4 PRINTCount

Count = TALLY("abc Defghi Jklmn Abc", "[:upper:]") %-- returns3 PRINTCount

Count = TALLY("abc Defghi Jklmn Abc", "[:lower:]") %-- returns14 PRINTCount

Count = TALLY("abc Defghi Jklmn Abc", "[:space:]") %-- returns3 PRINTCount

### Array.sort Array[{<start>,<length>}]~~}~~

Sorts the values of the specified array (Array[] or Array$[]) or array segment (Array[start,length] or Array$[start,length]) in ascending order.

\*\*\*\*\*

Сортирует значения указанного массива (Array [] или Array $ []) или сегмента массива (Array [start, length] или Array $ [start, length]) в порядке возрастания.

##### Array.search Array[{<start>,<length>}], <value\_exp>, <result\_nvar>{,<start\_nexp>}

**Array.search Array$[{<start>,<length>}], <value\_exp>, <result\_nvar>{,<start\_nexp>}**

Searches in the numeric or string array (Array[] or Array$[]) or array segment (Array[start,length] or Array$[start,length]) for the specified numeric or string value, which may be an expression. If the value is found in the array, its position will be returned in the result numeric variable <result\_nvar>. If the value is not found the result will be zero.

If only a segment of an array is used, the result is in conjunction to the start point.

\*\*\*\*\*

Поиск в числовом или строковом массиве (Array [] или Array $ []) или сегменте массива (Array [start, length] или Array $ [start, length]) для указанного числового или строкового значения, которое может быть выражением. Если значение найдено в массиве, его позиция будет возвращена в результирующей числовой переменной <result\_nvar>. Если значение не найдено, результат будет нулевым.

Если используется только сегмент массива, результат связан с начальной точкой.

Example:

Array.load ar[],10, 8,12, 20, 6, 7, 88, 11

Array.search ar[4, 3], 6, pos % pos returns 2 instead of 5

Array.search ar[4, 3], 88, pos % pos returns 0, because 88 is not part of the segment

### Array.binary.search Array[]|Array$[], <value\_exp>, <result\_nvar>

Searches in the numeric or string array (Array[] or Array$[])for the specified numeric or string value, which may be an expression. If the value is found in the array, its position will be returned in the result numeric variable <result\_nvar>. If the value is not found the result will be zero.

This command use the very fast binary-search-method. **Keep in mind, that you have to sort the array before to prevent unexpected results.**

**\*\*\*\*\***

**Ищет в числовом или строковом массиве (Array [] или Array $ []) указанное числовое или строковое значение, которое может быть выражением. Если значение найдено в массиве, его позиция будет возвращена в результирующей числовой переменной <result\_nvar>. Если значение не найдено, результат будет нулевым.**

**Эта команда использует очень быстрый двоичный метод поиска. Имейте в виду, что вы должны отсортировать массив прежде, чтобы предотвратить неожиданные результаты.**

Example:

Array.sort myTownArray$[]

Array.binary.search myTownArray$[], "Paris", resultIndex

### Array.Rnd Array[] {{{{{{, <length\_nexp>}, <low|offset\_nexp>}, <high| scale\_nexp>}, <seed\_nexp>}, <type\_nexp>}, <generator\_nexp>}

Creates a numeric array with pseudo-random numbers by using a new random generator. The array length is described by the optional <length\_nexp>. If not given an array with only one member is returned.

The optional <low|offset\_nexp> sets the starting value or the offset for Gaussianrandom numbers. Default is0.

The optional <high|scale\_nexp> sets the ending value or the scale for Gaussianrandom numbers. Default is1.

The optional <seed\_nexp> sets the seed. Default is 0 for unpredictable and not reproducible numbers. Seed number > 0 create reproducible results.

The optional <type\_nexp> sets the returned type of the random numbers. Default is 0 for normal random numbers. If <type\_nexp> is 1, Gaussian random numbers are returned. If

<type\_nexp> is 2, boolean random numbers are returned.

The optional <generator\_nexp> sets the random generator. Default is 0 for the standard generator. If <generator\_nexp> is 1, a secure random number generator (RNG) implementing the default random number algorithm will be constructed. Only usable for Android 4.4+ (KitKat+).

\*\*\*\*\*

Создает числовой массив с псевдослучайными числами, используя новый генератор случайных чисел. Длина массива описывается необязательным <length\_nexp>. Если не указан массив, возвращается только один член.

Необязательный <low | offset\_nexp> устанавливает начальное значение или смещение для гауссовских случайных чисел. По умолчанию is0.

Необязательный <high ​​| scale\_nexp> устанавливает конечное значение или масштаб для гауссовских случайных чисел. По умолчанию is1.

Необязательный <seed\_nexp> устанавливает начальное значение. По умолчанию 0 для непредсказуемых и невоспроизводимых чисел. Число семян> 0 создает воспроизводимые результаты.

Необязательный <type\_nexp> устанавливает возвращаемый тип случайных чисел. По умолчанию 0 для обычных случайных чисел. Если <type\_nexp> равен 1, возвращаются гауссовы случайные числа. Если

<type\_nexp> равно 2, возвращаются логические случайные числа.

Необязательный <generator\_nexp> устанавливает генератор случайных чисел. По умолчанию 0 для стандартного генератора. Если <generator\_nexp> равен 1, будет создан безопасный генератор случайных чисел (RNG), реализующий алгоритм случайных чисел по умолчанию. Используется только для Android 4.4+ (KitKat+).

Example:

Array.Rnd r[], 2, 0, 100, 4711, 0

Print r[1] % Returns 4.315872869138159 Print r[2] % Returns 37.83154009540295 Array.Rnd nr[]

Print nr[1] % Returns an unpredictable and not reproducible number between 0 and < 1.

##### Array.to.dims SourceArray[], DimensionArray[], DestinationArray[]Array.to.dims SourceArray$[], DimensionArray[], DestinationArray$[]

Copies all elements of an existing SourceArray[] to the DestinationArray[], which is specified in order by the DimensionArray[]. The number of source and destination array elements must be the equal. If the Destination Array exists, it will be overwritten. If the DestinationArray does not exist, a new array is created. The arrays may be either numeric or string arrays but they must both be of the same type.

\*\*\*\*\*

Копирует все элементы существующего SourceArray [] в DestinationArray [], который указывается в порядке DimensionArray []. Количество элементов массива источника и назначения должно быть одинаковым. Если Массив назначения существует, он будет перезаписан. Если DestinationArray не существует, создается новый массив. Массивы могут быть как числовыми, так и строковыми, но оба они должны быть одного типа.

Example:

Array.load s[], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 % 10 elements

Array.load d[], 5, 2 % 5 \* 2 = 10 Array.to.dims s[], d[], r[]

Print r[1, 1] % Returns0.0

Print r[5, 2] % Returns9.0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \ | **1** | **2** | **3** | **4** | **5** |
| **1** | *0* | *2* | *4* | *6* | *8* |
| **2** | *1* | *3* | *5* | *7* | *9* |

##### Array.Mat.Transpose SourceArray[], DestinationArray[]

**Array.Mat.Transpose SourceArray$[], DestinationArray$[]**

Copies all elements of an existing SourceArray[] to the DestinationArray[], but transposes over the element [1,1]. The sum of the source and target array elements is the same. If the target array is present, it will be overwritten. If the target array does not exist, a new array is created. The arrays can be either numeric or string arrays, but both must be of the same type.

\*\*\*\*\*

Копирует все элементы существующего SourceArray [] в DestinationArray [], но переносит поверх элемента [1,1]. Сумма исходного и целевого элементов массива одинакова. Если целевой массив присутствует, он будет перезаписан. Если целевой массив не существует, создается новый массив. Массивы могут быть как числовыми, так и строковыми, но оба должны быть одного типа.

Example:

Array.load s[], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 % 10 elements

Array.load d[], 5, 2 % 5 \* 2 = 10 Array.to.dims s[], d[], r[] Array.Mat.Transpose r[], n[]

Array.dims n[], d[] % d[] returns 2, 5 instead 5, 2 Before transposing:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| \ | **1** | **2** | **3** | **4** | **5** |
| **1** | *0* | *2* | *4* | *6* | *8* |
| **2** | *1* | *3* | *5* | *7* | *9* |

After transposing:

|  |  |  |
| --- | --- | --- |
| \ | **1** | **2** |
| **1** | *0* | *1* |
| **2** | *2* | *3* |
| **3** | *4* | *5* |
| **4** | *6* | *7* |
| **5** | *8* | *9* |

##### Array.math LeftArray[], RightArray[], <operator\_sexp>, ResultArray[]

Operates the left array with the right array by the given operator string <operator\_sexp>. For a command description see the corresponding BASIC! Functions.

The dimensions of the result array are equal to the **left** array.

The advantage if this command is the executing speed in opposite to For-Next loops. The possible operators are:

\_\*, \_/, \_+, \_-, \_min, \_max, \_atn2, \_hypot, \_pow, \_mod, \_=, \_<>, \_<, \_>, \_<=, \_>=,

\_shift, \_bor, \_band, \_bxor

\*\*\*\*\*

Управляет левым массивом с правым массивом с помощью заданной строки оператора <operator\_sexp>. Описание команды см. В соответствующем бейсике! Функции.

Размеры результирующего массива равны левому массиву.

Преимущество этой команды заключается в скорости выполнения в противоположность циклам For-Next. Возможные операторы:

\_\*, \_/, \_+, \_-, \_min, \_max, \_atn2, \_hypot, \_pow, \_mod, \_=, \_<>, \_<, \_>, \_<=, \_>=,

\_shift, \_bor, \_band, \_bxor

Example:

Array.load left[], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 % 10 elements

Array.load right[], 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 % 10 elements Array.math left[], right[], "\_+", res[]

Print res[1] % Returns 0.0 Print res[5] % Returns 12.0

Array.math left[], right[], "\_=", res[] Print res[3] % Returns 1.0 for true

The following functions are assigned to the members of the **right** array.

\*\*\*\*\*

Следующие функции назначены членам правого массива.

##### (op) can be +, -, \*, /

\_(op)abs, \_(op)sgn, \_(op)ceil, \_(op)floor, \_(op)int, \_(op)frac, \_(op)round,

\_(op)const\_pi, \_(op)sin, \_(op)sinh, \_(op)asin, \_(op)cos, \_(op)cosh, \_(op)acos,

\_(op)tan, \_(op)tanh, \_(op)atan, \_(op)todegrees, \_(op)toradians,

\_(op)sqr, \_(op)cbrt, \_(op)exp, \_(op)log, \_(op)log10, \_(op)const\_e,

\_(op)=0, \_(op)<>0, \_(op)bnot

If you have no left array, fill in a dummy with zeros and proceed as follows:

\*\*\*\*\*

Если у вас нет левого массива, заполните пустышку нулями и выполните следующие действия:

Example:

Array.length al, RightArray[] DIM LeftArray[al]

Array.fill LeftArray[], 0

! Return the content from RightArray[] as absolute values Array.math LeftArray[], RightArray[], "\_+abs", ResultArray[]

! Return the boolean values of the RightArray[] as 1.0 for true and 0.0 for false

! if the given value is not 0. Works like an on and off switch. Array.math LeftArray[], RightArray[], "\_+<>0", ResultArray[]

Example How to Scale a 3D Vector:

Array.load vector[], 1, 1, 4

s = 3

Array.load multi[], s, s, s

! Return the scaled vector. The result is in the vector array on the right side. Array.math vector[], multi[], "\_\*", vector[] % Result is 3, 3, 12

See also List.join, List.map.2d, List.map.3d, Array.Mat.Transpose

##### REDIM {<preserve\_nexp>}, Array[<nexp>{, <nexp> } ... ] ...

**REDIM {<preserve\_nexp>}, Array$[<nexp>{, <nexp> } ... ] ...**

Re-Dimensions an existing array with optional 'preserve contents' flag. Referenced arrays are not cut off (when passing to a user function).

Arrays must already exist before REDIM.

If the preserve flag is zero, the array will be full of 0's or "".

If the preserve flag is non-zero, the old contents will be preserved to the size of the old or new array, whichever was smallest. Any cells left-over will be 0's or "".

Preserve flags may be inserted anywhere in the arguments list.

Arrays following a preserve flag will honor that preserve flag until the next preserve flag. Preserve flags cannot be array expressions e.g. p[3], as this would re-dim p[3].

\*\*\*\*\*

Изменение размеров существующего массива с необязательным флагом «сохранить содержимое». Ссылочные массивы не обрезаются (при передаче пользовательской функции).

Массивы уже должны существовать до REDIM.

Если флаг сохранения равен нулю, массив будет заполнен нулями или "".

Если флаг сохранения не равен нулю, старое содержимое будет сохранено до размера старого или нового массива, в зависимости от того, какой из них был наименьшим. Любые оставшиеся ячейки будут 0 или "".

Флажки сохранения могут быть вставлены в любое место списка аргументов.

Массивы после флага сохранения сохраняют этот флаг до следующего флага сохранения. Флаги сохранения не могут быть выражениями массива, например p [3], поскольку это повторно уменьшило бы p [3].

Example:

REDIM 1, a[], 0, b[] 1,c[],d[] % b[] will not bepreserved

### Array Enhancements

##### Array Assignments

OliBasic accepts Array pointers e.g a[], a$[] for assignment and evaluation.

Assignments e.g a[] = b[]

a[] = b[] % makes an exact copy of b[] to a[] including the size and dimensions.

% background references (i.e passed by referenced variables) are preserved.

Evaluation (for future mod).

Evaluation of a[] alwaysreturns0 ( might be changed in the future ) Evaluation of a$[] alwaysreturns"" ( might be changed in the future ) Evaluation of non-existent a[] also returns 0 or "" ( might be changed in the future)

The term 'array pointer' is a reference to an array. You may think of it as representing the whole array. This is inline with traditional Basic 01.91 which already uses the syntax for some commands and functions.

The behavior for array pointers (e.g a[]) has only been changed for evaluation and assignment.

It has not been changed when given to Basic commands or functions e.g myfunc a[]). This is because Basic commands and functions do their own parsing and have their own rules for arrays without indices.e-Dimensions an existing array with optional 'preserve contents' flag.

\*\*\*\*\*

OliBasic принимает указатели массива, например a [], $ [] для назначения и оценки.

Назначения, например, a [] = b []

a [] = b []% делает точную копию b [] a [], включая размер и размеры.

% фоновых ссылок (то есть передаваемых ссылочными переменными) сохраняются.

Оценка (для будущего мода).

Оценка [] всегда возвращается 0 (может быть изменено в будущем) Оценка $ [] всегда возвращается "" (может быть изменено в будущем) Оценка несуществующего а [] также возвращает 0 или "" (может быть изменено в будущее)

Термин «указатель массива» является ссылкой на массив. Вы можете думать об этом как о представлении всего массива. Это соответствует традиционной базовой версии 01.91, которая уже использует синтаксис для некоторых команд и функций.

Поведение для указателей массива (например, []) было изменено только для оценки и назначения.

Он не был изменен, если он передан базовым командам или функциям, например, myfunc a []). Это связано с тем, что базовые команды и функции выполняют собственный анализ и имеют свои собственные правила для массивов без индексов. E-Измеряет существующий массив с необязательным флагом «сохранить содержимое».

### Functions

##### Fn.def name|name$( {nvar}|{svar}|Array[]|Array$[], ... {nvar}|{svar}|Array[]|Array$[])

**{[]}**

Begins the definition of a function. This command names the function and lists the parameters, if any.

If the function name ends with the $ character then the function will return a string, otherwise it will return a number. Ends the command with [], the function returns a string array or numeric array. The parameter list can contain as many parameters as needed, or none at all. The parameters may be numeric or string, scalar or array.

Your program must execute **Fn.def** before it tries to call the named function. Your program must not attempt to create more than one function with the same name, or the same function more than once. However, you may override a built-in function by defining your own function with the same name.

The following are all valid:

FN.DEF cut$(a$, left, right)

FN.DEF sum(a, b, c, d, e, f, g, h, i, j) FN.DEF sort(v$[], direction)

FN.DEFpi() % Overrides built-in. You can make π =3!

Parameters create variables visible only inside the function. They can be used like other variables created inside the function (see Variable Scope, above).

There are two types of parameters: call by reference and call by value. Call by value means that the calling variable value (or expression) is copied into the called variable. Changes made to the called variable within the function do not affect the value of the calling variable. Call by reference means that the calling variable value is changed if the called variable value is changed within the function.

Scalar (non-array) function variables can be either call by value or call by reference. Which type the variable will be depends upon how it is called. If the calling variable has the "&" character in front of it, then the variable is call by reference. If there is no "&" in front of the calling variable name then the variable is call by value.

FN.DEF test(a) a = 9

FN.RTN a FN.END

a =1

PRINT test(a), a %will print: 9, 1 PRINT test(&a), a %will print: 9, 9

Array parameters are always call by reference.

FN.DEF test(a[]) a[1] = 9 FN.RTN a[1] FN.END

DIM a[1] a[1] = 1

PRINT test(a[]), a[1] % prints: 9, 9

Along with the function’s return value, you can use parameters passed by reference to return information to a function’s caller.

\*\*\*\*\*

Начинается определение функции. Эта команда называет функцию и перечисляет параметры, если таковые имеются.

Если имя функции заканчивается символом $, то функция вернет строку, в противном случае вернет число. Завершает команду с помощью [], функция возвращает строковый массив или числовой массив. Список параметров может содержать столько параметров, сколько необходимо, или ни одного. Параметры могут быть числовыми или строковыми, скалярными или массивными.

Ваша программа должна выполнить Fn.def, прежде чем она попытается вызвать указанную функцию. Ваша программа не должна пытаться создать более одной функции с одинаковым именем или одну и ту же функцию более одного раза. Однако вы можете переопределить встроенную функцию, определив собственную функцию с тем же именем.

Следующие все действительны:

FN.DEF вырезать $ (a $, слева, справа)

Сумма FN.DEF (a, b, c, d, e, f, g, h, i, j) Сортировка FN.DEF (v $ [], направление)

FN.DEFpi ()% Переопределяет встроенный. Вы можете сделать π = 3!

Параметры создают переменные, видимые только внутри функции. Они могут использоваться как другие переменные, созданные внутри функции (см. Область переменных выше).

Существует два типа параметров: вызов по ссылке и вызов по значению. Вызов по значению означает, что значение вызывающей переменной (или выражение) копируется в вызываемую переменную. Изменения, внесенные в вызываемую переменную внутри функции, не влияют на значение вызывающей переменной. Вызов по ссылке означает, что значение вызывающей переменной изменяется, если в функции изменяется значение вызываемой переменной.

Скалярные (не массив) переменные функции могут быть либо вызовом по значению, либо вызовом по ссылке. Тип переменной будет зависеть от того, как она вызывается. Если вызывающая переменная имеет символ «&» перед ней, то переменная вызывается по ссылке. Если перед именем вызывающей переменной нет символа «&», тогда переменная вызывается по значению.

FN.DEF тест (а) а = 9

FN.RTN FN.END

а = 1

PRINT test (a),% напечатает: 9, 1 PRINT test (& a),% напечатает: 9, 9

Параметры массива всегда вызываются по ссылке.

Тест FN.DEF (a []) a [1] = 9 FN.RTN a [1] FN.END

DIM a [1] a [1] = 1

PRINT test (a []), a [1]% отпечатков: 9, 9

Наряду с возвращаемым значением функции вы можете использовать параметры, переданные по ссылке, для возврата информации вызывающей функции.

##### Fn.rtn <sexp>|<nexp>{[]}

Causes the function to terminate execution and return the value of the return expression

<sexp>|<nexp> or an array of those types. The return expression type, string, ~~or~~number or array, must match the type of the function name. **Fn.rtn** statements may appear anywhere in the program that they are needed.

A function can return ~~only~~a single scalar value. It can~~not~~return an array also. It cannot return a data structure (List, Stack, Bundle, or graphical object), but it can return a pointer to a data structure.

Note: You can also return information to a function’s caller through parameters passed by reference.

\*\*\*\*\*

Заставляет функцию завершить выполнение и вернуть значение возвращаемого выражения

<sexp> | <nexp> или массив этих типов. Тип возвращаемого выражения, строка, номер или массив должны соответствовать типу имени функции. Операторы Fn.rtn могут появляться в любом месте программы, где они необходимы.

Функция может возвращать только одно скалярное значение. Он также не может вернуть массив. Он не может вернуть структуру данных (List, Stack, Bundle или графический объект), но он может вернуть указатель на структуру данных.

Примечание. Вы также можете вернуть информацию вызывающей функции через параметры, переданные по ссылке.

##### Fn.end

Ends the definition of a user-defined function. Every function definition must end with

##### Fn.end.

When your function is running, executing the **Fn.end** statement causes the function to terminate and return a default value. If the function type is numeric then the default return value is 0.0. A string function returns the empty string (""). A string function returns an one- dimensional array with one element (containing 0.0 or "").

Note: Maybe in future empty arrays will be allowed.

\*\*\*\*\*

Заканчивает определение пользовательской функции. Каждое определение функции должно заканчиваться

Fn.end.

Когда ваша функция выполняется, выполнение оператора Fn.end приводит к завершению функции и возвращению значения по умолчанию. Если тип функции числовой, то возвращаемое значение по умолчанию равно 0,0. Строковая функция возвращает пустую строку (""). Строковая функция возвращает одномерный массив с одним элементом (содержащий 0.0 или "").

Примечание. Возможно, в будущем пустые массивы будут разрешены.

### Advanced LIST Commands for Advanced Users

The next few commands take more effort to learn how they work and where the benefits are. One of them is an up to 30 time increasing speed instead of normal loops.

List.join is very powerful, but also complex.

It is useful to put your special solutions in separate functions with a simpler interface.

\*\*\*\*\*

Следующие несколько команд потребуют больше усилий, чтобы узнать, как они работают и каковы преимущества. Один из них - увеличение скорости до 30 раз вместо обычных петель.

List.join очень мощный, но и сложный.

Полезно поместить ваши специальные решения в отдельные функции с более простым интерфейсом.

### List.split {<left\_nexp>}, {<right\_nexp>}, <source\_nexp>, <by\_reg\_sexp>

**{{{, <start\_nexp>}, <end\_nexp>}, <add\_nexp>}**

Splits the source list <source\_nexp> into two lists and place them into <left\_nexp> and

<right\_nexp> by the regular expression <by\_reg\_sexp> item by item. If the right part does not exist, an empty string "" or 0.0 is returned.

With the list type (S or N) you control the type of your output. The type of the source list is detected automatically.

The <left\_nexp> and <right\_nexp> lists are optional, but you definitely need one.

The parameters <start\_nexp> and <end\_nexp> point to the range of the source list to work with. If no value is given the begin respectively the end is used.

The <add\_nexp> argument let you add (<add\_nexp> = 1) the results to the output lists. If this argument is 0 (default) the output lists are cleared before execution.

Warning: A source list can not be an output list in the same command.

\*\*\*\*\*

Разбивает исходный список <source\_nexp> на два списка и помещает их в <left\_nexp> и

<right\_nexp> по регулярному выражению <by\_reg\_sexp> элемент за элементом. Если правая часть не существует, возвращается пустая строка "" или 0.0.

С типом списка (S или N) вы управляете типом вашего вывода. Тип списка источников определяется автоматически.

Списки <left\_nexp> и <right\_nexp> являются необязательными, но вам определенно нужен.

Параметры <start\_nexp> и <end\_nexp> указывают на диапазон списка источников для работы. Если значение не указано, начало и соответственно конец используются.

Аргумент <add\_nexp> позволяет добавить (<add\_nexp> = 1) результаты в выходные списки. Если этот аргумент равен 0 (по умолчанию), выходные списки очищаются перед выполнением.

Предупреждение: список источников не может быть списком вывода в той же команде.

See also: REPLACE$, SPLIT, SPLIT.ALL

### List.join <result\_nexp>, {<scr\_left\_nexp>|<scr\_left\_sexp>},

**{<scr\_right\_nexp>|<scr\_right\_sexp>}, <delim\_sexp> {{{{, <\_oper\_arg\_sexp>},**

**<start\_nexp>}, <end\_nexp>}, <add\_nexp>}**

Joins the optional source lists <scr\_left\_nexp> and <scr\_right\_nexp> into list

<result\_nexp> item by item. The list <scr\_right\_nexp> is optional.

With your chosen list type (S or N) you control the type of your output <result\_nexp> automatically. The type of source lists is recognized and converted internally and automatically into strings.

The optional <scr\_left\_sexp> and <scr\_right\_sexp> are representing a list with limited size and equal content. The size limits depend on the size of the other list or the arguments

<end\_nexp> and <add\_nexp>. Numeric values have to be converted to Strings maybe with STR$(<nexp>).

Is the item count of the lists <scr\_left\_nexp> and <scr\_right\_nexp> different, the item count of the list with the most items or the setting is used. In this case the empty item returns "" or 0.0 ( \_+(.), \_-(.) ) or 1.0 ( \_\*(.), \_/(.)([0-9]) ). **Study with trying two lists with different length and learn with the results!** Compare with the example line which ends with **!\*/!.**

The delimiter <delim\_sexp> is added normally after the <scr\_left\_nexp> item. After that the optional <\_oper\_arg\_sexp> is executed before adding into the list

<result\_nexp>.

The parameters <start\_nexp> and <end\_nexp> point to the range of the source list to work with. If no value is given the begin respectively the maximum end is used.

Remember that <end\_nexp> is **not limited!** See the example line which ends with "A index list function".

The <add\_nexp> argument lets you add (<add\_nexp> = 1) the results to the output lists. If this argument is 0 (default) the output lists are cleared before execution.

##### Operators

For String expressions valid operators are: \_+$

For Numeric expressions valid operators are: \_+, \_-, \_\*, \_/{scale\_nexp},

\_\*sin, \_\*cos, \_\*tan, \_\*asin, \_\*acos, \_\*atan, \_\*sqr

You can also put a point (**.**) into the operator to account for zeros (0) before missed decimal points like , \_+**.**, \_-**.**, \_\***.**, \_/**.**{scale\_nexp}, \_\***.**sin …. .

The optional {scale\_nexp} is set per default to 16.

\_+i equals the list index and \_-i equals the list index as a negative numper

\_atan4 is special, because it returns the angle for the term y / x in the range [0° … 360°]

**counterclockwise**. <scr\_**left**\_nexp> is **y!!!,** <scr\_**right**\_nexp> is **x.**

Note, if y = 0 and x = 0, the result is 0.

\_min, \_max are special, because it returns the minimum or maximum of <scr\_**left**\_nexp> and <scr\_**right**\_nexp> is **x.**

The four basic arithmetic operations are calculated as BigDecimal, after the entire input has been converted to strings before. Other operations will be computed as Double.

Keep in mind, that values from type Double contain only maximal 15 **correct** digits. Trigonometric functions use **degrees** as in- and output.

If the <\_oper\_arg\_sexp> starts additionally with **\_D** all arithmetic operations are calculated as Double, that increases the speed round about 130% with lost of accuracy. In this case the scale argument has no effect.

##### Arguments

Arguments as string expressions have to be enclosed in quotation marks "text" ( \"text\" or + CHR$(34) + "text" + CHR$(34) + ).

\*\*\*\*\*

Объединяет необязательные исходные списки <scr\_left\_nexp> и <scr\_right\_nexp> в список

<result\_nexp> элемент за элементом. Список <scr\_right\_nexp> не является обязательным.

Выбрав тип списка (S или N), вы автоматически управляете типом вывода <result\_nexp>. Тип списков источников распознается и автоматически и автоматически преобразуется в строки.

Необязательные <scr\_left\_sexp> и <scr\_right\_sexp> представляют список с ограниченным размером и равным содержанием. Ограничения размера зависят от размера другого списка или аргументов

<end\_nexp> и <add\_nexp>. Числовые значения должны быть преобразованы в строки, возможно с помощью STR $ (<nexp>).

Отличается ли количество элементов в списках <scr\_left\_nexp> и <scr\_right\_nexp>, используется ли количество элементов в списке с наибольшим количеством элементов или используется настройка. В этом случае пустой элемент возвращает "" или 0.0 (\_ + (.), \_- (.)) Или 1.0 (\_ \* (.), \_ / (.) ([0-9])). Учитесь, пробуя два списка разной длины, и учитесь с результатами! Сравните с примером строки, которая заканчивается на! \* / !.

Разделитель <delim\_sexp> обычно добавляется после элемента <scr\_left\_nexp>. После этого необязательный <\_oper\_arg\_sexp> выполняется перед добавлением в список

<Result\_nexp>.

Параметры <start\_nexp> и <end\_nexp> указывают на диапазон списка источников для работы. Если значение не указано, начало и соответственно используется максимальное значение.

Помните, что <end\_nexp> не ограничен! Смотрите пример строки, которая заканчивается «Функция списка индексов».

Аргумент <add\_nexp> позволяет добавить (<add\_nexp> = 1) результаты в выходные списки. Если этот аргумент равен 0 (по умолчанию), выходные списки очищаются перед выполнением.

операторы

Для строковых выражений допустимы следующие операторы: \_ + $

Для числовых выражений допустимы следующие операторы: \_ +, \_-, \_ \*, \_ / {scale\_nexp},

\_ \* грех, \_ \* cos, \_ \* загар, \_ \* асин, \_ \* acos, \_ \* atan, \_ \* sqr

Вы также можете поместить точку (.) В оператор для учета нулей (0) перед пропущенными десятичными точками, такими как, \_ +., \_-., \_ \*., \_ /. {Scale\_nexp}, \_ \*. Sin…. ,

Необязательный {scale\_nexp} установлен по умолчанию на 16.

\_ + i равен индексу списка, а \_-i равен индексу списка как отрицательный номер

\_atan4 особенный, потому что он возвращает угол для члена y / x в диапазоне [0 ° ... 360 °]

против часовой стрелки. <scr\_left\_nexp> - это y !!!, <scr\_right\_nexp> - это x.

Обратите внимание, если y = 0 и x = 0, результат равен 0.

\_min, \_max являются особыми, потому что возвращает минимум или максимум <scr\_left\_nexp>, а <scr\_right\_nexp> равен x.

Четыре базовые арифметические операции вычисляются как BigDecimal после того, как весь вход был преобразован в строки ранее. Другие операции будут вычислены как Double.

Помните, что значения типа Double содержат только максимум 15 правильных цифр. Тригонометрические функции используют градусы как на входе, так и на выходе.

Если <\_oper\_arg\_sexp> начинается дополнительно с \_D, все арифметические операции рассчитываются как двойные, что увеличивает скорость примерно на 130% с потерей точности. В этом случае аргумент масштаба не имеет никакого эффекта.

аргументы

Аргументы в виде строковых выражений должны быть заключены в кавычки «текст» (\ «текст \» или + CHR $ (34) + «текст» + CHR $ (34) +).

(LL)LEFT(LR) DELIMITER OPERATOR&ARGUMENT (RL)RIGHT(RR)

## \ /

\ /

\ /

(OL)OUTPUT RESULT(OR)

Examples:

a <scr\_left\_nexp> item = 20

a <scr\_rigth\_nexp> item = 010

<delim\_sexp> = "." and <\_oper\_arg\_sexp> = "" returns 20.010

<delim\_sexp> = "" and <\_oper\_arg\_sexp> = "\_+" returns 30

<delim\_sexp> = "." and <\_oper\_arg\_sexp> = "\_+" returns 20.1

<delim\_sexp> = "" and <\_oper\_arg\_sexp> = "\_+**.**" returns 20.01

<delim\_sexp> = "" and <\_oper\_arg\_sexp> = "\_/**.**3" returns 2000.000

<delim\_sexp> = "" and <\_oper\_arg\_sexp> = "\_min**.**" returns 0.01

<delim\_sexp> = "§" and <\_oper\_arg\_sexp> = "\_+$\_" + "#" returns #20§010#

only <scr\_left\_nexp> item = 20

<delim\_sexp> = "" and <\_oper\_arg\_sexp> = "\_-**.**\_04" returns 19.96 only <scr\_rigth\_nexp> item = 010

<delim\_sexp> = "" and <\_oper\_arg\_sexp> = "\_\***.**\_2.5" returns 0.025 % If left side is "", 1 is used **!\*/!**

no source lists, <start\_nexp> = 5, <end\_nexp> = 6,

<delim\_sexp> = "" and <\_oper\_arg\_sexp> = "\_-i" returns -5 and -6 % A index list function

Warning: A source list can not be an output list in the same command.

Note: The special Floating Point numbers NaN and Infinity are not supported, if using BigDecimal.

\*\*\*\*\*

Предупреждение: список источников не может быть списком вывода в той же команде.

Примечание: специальные числа с плавающей точкой NaN и Infinity не поддерживаются, если используется BigDecimal.

### List.binary.search <pointer\_nexp>, search\_nexp|search\_sexp, <result\_nvar>

Searches the specified list for the specified string or numeric value. The position of the first (left-most) occurrence is returned in the numeric variable <result\_nvar>. If the value is not found in the list then the result is zero.

This command use the very fast binary-search-method. **Keep in mind, that you have to sort the list before to prevent unexpected results.**

**\*\*\*\*\***

**Поиск в указанном списке указанной строки или числового значения. Позиция первого (крайнего левого) вхождения возвращается в числовой переменной <result\_nvar>. Если значение не найдено в списке, результат равен нулю.**

**Эта команда использует очень быстрый двоичный метод поиска. Имейте в виду, что вы должны отсортировать список, чтобы избежать неожиданных результатов.**

Example:

List.sort lPtr, 0, "fr\_FR"

List.binary.search lPtr, "Paris", resultIndex

### List.match {<index\_nexp>}, {<result\_nexp>}, <source\_nexp>,

**<by\_find\_sexp> {{{{, <start\_nexp>}, <end\_nexp>}, <add\_nexp>},**

**<mode\_sexp>}**

Checks with the expression <by\_reg\_sexp> for matches in the optional source list

<source\_nexp> and put the corresponding index into the optional list <index\_nexp> and the result into the list <result\_nexp>.

With your chosen list type (S or N) you control the type of your output <result\_nexp> automatically. The type of the source list is detected automatically.

The parameters <start\_nexp> and <end\_nexp> point to the area of the source list with which you want to work. If no value is given the begin respectively the end is used.

The <add\_nexp> argument let you add (<add\_nexp> = 1) the results to the output lists. If this argument is 0 (default) the output lists are cleared before execution.

Options of matching mode <mode\_sexp>:

"\_RegEx" stands for regular expressions.

Valid **String expressions** are: \_Default = \_Is\_In, \_Is\_In\_IgnoreCase, \_Starts\_With,

\_Starts\_With\_IgnoreCase, \_Ends\_With, \_Ends\_With\_IgnoreCase,\_Equals,

\_Equals\_IgnoreCase, \_RegEx\_First and \_RegEx\_First\_IgnoreCase. Valid **Numeric expressions** are: \_Default = \_Is\_In\_IgnoreCase,

\_Starts\_With\_IgnoreCase, \_Ends\_With\_IgnoreCase, \_Equals\_IgnoreCase,

\_RegEx\_First\_IgnoreCase and \_Equals\_Numeric.

Warning: A source list can not be an output list in the same command.

\*\*\*\*\*

Проверяет с помощью выражения <by\_reg\_sexp> на совпадения в дополнительном списке источников

<source\_nexp> и поместите соответствующий индекс в необязательный список <index\_nexp>, а результат - в список <result\_nexp>.

Выбрав тип списка (S или N), вы автоматически управляете типом вывода <result\_nexp>. Тип списка источников определяется автоматически.

Параметры <start\_nexp> и <end\_nexp> указывают на область списка источников, с которой вы хотите работать. Если значение не указано, начало и соответственно конец используются.

Аргумент <add\_nexp> позволяет добавить (<add\_nexp> = 1) результаты в выходные списки. Если этот аргумент равен 0 (по умолчанию), выходные списки очищаются перед выполнением.

Опции режима соответствия <mode\_sexp>:

«\_RegEx» обозначает регулярные выражения.

Допустимые строковые выражения: \_Default = \_Is\_In, \_Is\_In\_IgnoreCase, \_Starts\_With,

\_Starts\_With\_IgnoreCase, \_Ends\_With, \_Ends\_With\_IgnoreCase, \_Equals,

\_Equals\_IgnoreCase, \_RegEx\_First и \_RegEx\_First\_IgnoreCase. Допустимые числовые выражения: \_Default = \_Is\_In\_IgnoreCase,

\_Starts\_With\_IgnoreCase, \_Ends\_With\_IgnoreCase, \_Equals\_IgnoreCase,

\_RegEx\_First\_IgnoreCase и \_Equals\_Numeric.

Предупреждение: список источников не может быть списком вывода в той же команде.

See also: REPLACE$, SPLIT, SPLIT.ALL

##### List.sort <pointer\_nexp>{{{, <sort\_mode\_nexp>}, <locale\_sexp>},

**<strength\_sexp>}**

Sorts the content of the given list. Options: <sort\_mode\_nexp>:

* 0 Sorted in ascending, UTF-8 character table numbered order(default)
* 1 Sorted in descending, UTF-8 character table numberedorder

If the <locale\_sexp> is set, the output is based on language and region. The locale specifies the language and region with standardized codes. The <locale\_sexp> is a string containing zero or more codes separated by underscores.

The function accepts up to three codes. The first must be a language code, such as "en", "de" or "ja".

The second must be a region or country code, such as "FR", "US", or "IN". Some language and country combinations can accept a third code, called the "variant code".

The function also accepts the standard three-letter codes and numeric codes for country or region. For example, "fr\_FR", "fr\_FRA", and "fr\_250" are all equivalent.

If <locale\_sexp> = "" , meaning "use my default locale".

If <locale\_sexp> = empty, the list is sorted by the order of the character map like Array.sort.

To control the strength of the sorting use <strength\_sexp> with the keys described in List.sort.by.

\*\*\*\*\*

Сортирует содержимое данного списка. Параметры: <sort\_mode\_nexp>:

0 Сортировка по возрастанию, нумерованная таблица символов UTF-8 (по умолчанию)

1 отсортировано по убыванию, нумерованный порядок таблицы символов UTF-8

Если установлено значение <locale\_sexp>, выходные данные основаны на языке и регионе. Локаль указывает язык и регион со стандартными кодами. <Locale\_sexp> - это строка, содержащая ноль или более кодов, разделенных подчеркиванием.

Функция принимает до трех кодов. Первым должен быть код языка, такой как «en», «de» или «ja».

Вторым должен быть код региона или страны, например «FR», «US» или «IN». Некоторые комбинации языков и стран могут принимать третий код, называемый «код варианта».

Функция также принимает стандартные трехбуквенные коды и цифровые коды для страны или региона. Например, «fr\_FR», «fr\_FRA» и «fr\_250» все эквивалентны.

Если <locale\_sexp> = "", что означает "использовать мою локаль по умолчанию".

Если <locale\_sexp> = пусто, список сортируется по порядку карты символов, например Array.sort.

Чтобы контролировать силу сортировки, используйте <strong\_sexp> с ключами, описанными в List.sort.by.

##### List.sort.by {<index\_nexp>}, {<toSort\_nexp>}, <by\_nexp>{{{, <sort\_mode\_nexp>},

**<locale\_sexp>}, <strength\_sexp>}**

Creates an optional index list <index\_nexp> by sorting the content of the given list

<by\_nexp>. Optional will be the list <toSort\_nexp> ordered by the internally created index. The list <index\_nexp> will be overwriten.

The lists <toSort\_nexp> and <by\_nexp> should be the same size and their types numeric and string can be combined.

Options: <sort\_mode\_nexp>:

* 0 Sorted in ascending, UTF-8 character table numbered order(default)
* 1 Sorted in descending, UTF-8 character table numberedorder

If the <locale\_sexp> is set, the output is based on language and region. The locale specifies the language and region with standardized codes. The <locale\_sexp> is a string containing zero or more codes separated by underscores.

The function accepts up to three codes. The first must be a language code, such as "en", "de" or "ja".

The second must be a region or country code, such as "FR", "US", or "IN". Some language and country combinations can accept a third code, called the "variant code".

The function also accepts the standard three-letter codes and numeric codes for country or region. For example, "fr\_FR", "fr\_FRA", and "fr\_250" are all equivalent.

If <locale\_sexp> = "" , meaning "use my default locale".

If <locale\_sexp> = empty, the list is sorted by the order of the character map like Array.sort.

To control the strength of the sorting use <strength\_sexp> with the following keys. "\_Primary". Typically, recognizes differences in the base character so that "a" is smaller than "b". There are no differences between accents and umlauts, so that "a", "ä" and "á" are the same.

"\_Secondary". **Is the default key**. Detects characters with accents. So "a" and "á" are not the same anymore as in \_Primary. Accents in the characters are considered secondary differences (for example, "as" < "às" < "at"). Other differences between letters can also be considered secondary differences, depending on the language. A secondary difference is ignored when there is a primary difference anywhere in the strings.

"\_Tertiary". Distinguishes in upper and lower case; in \_Primary and \_Secondary the spelling does not matter, and "a" is equal to "A". Upper and lower case differences in characters are distinguished at tertiary strength (for example, "ao" < "Ao" < "aò"). In addition, a variant of a letter differs from the base form on the tertiary strength (such as "A" and "Ⓐ"). Another example is the difference between large and small Kana. A tertiary difference is ignored when there is a primary or secondary difference anywhere in the strings.

"\_Identical". Really all Unicode characters are different. While the first three constants treat non-visible characters like CHR$(1) or CHR$(6) the same, they're really different under

\_Identical. When all other strengths are equal, the \_Identical strength is used as a tiebreaker. For example, Hebrew cantillation marks are only distinguished at this strength. This strength should be used sparingly, as only code point value differences between two strings are an extremely rare occurrence. Using this strength substantially decreases the performance.

\*\*\*\*\*

rting используйте <strong\_sexp> со следующими ключами. "\_Primary". Как правило, распознает различия в базовом символе, так что «а» меньше, чем «б». Между акцентами и умлаутами нет различий, поэтому «а», «а» и «а» - это одно и то же.

"\_Secondary". Ключ по умолчанию. Обнаруживает персонажей с акцентами. Так что «а» и «а» больше не совпадают с «Первичным». Акценты в символах считаются вторичными различиями (например, «as» <«às» <«at»). Другие различия между буквами также могут рассматриваться как второстепенные различия, в зависимости от языка. Вторичная разница игнорируется, когда в строках есть первичная разница.

"\_Tertiary". Различает в верхнем и нижнем регистре; в \_Primary и \_Secondary написание не имеет значения, и «a» равно «A». Различия символов в верхнем и нижнем регистре различаются с высшей силой (например, «ao» <«Ao» <«aò»). Кроме того, вариант буквы отличается от базовой формы третичной прочностью (например, «А» и «Ⓐ»). Другим примером является разница между большой и маленькой кана. Третичная разница игнорируется, когда в строках есть первичная или вторичная разница.

"\_Identical". На самом деле все символы Юникода разные. Хотя первые три константы обрабатывают невидимые символы, такие как CHR $ (1) или CHR $ (6), они действительно различаются

\_Identical. Когда все остальные сильные стороны равны, \_идентичная сила используется в качестве разрыва связи. Например, еврейские метки кантилляции различаются только при этой силе. Эту силу следует использовать с осторожностью, поскольку различия между значениями только между двумя строками встречаются крайне редко. Использование этой силы существенно снижает производительность.

Examples for the "de" language code:

**\_Primary**

abc = ABC Quäken = Quaken boß = boss

boß <boxen

**\_Secondary**

abc = ABC Quäken > Quaken boß = boss

boß < boxen

**\_Tertiary**

abc < ABC Quäken > Quaken boß > boss

boß < boxen

Example for special German sortings: DEBUG.ON

LIST.CREATE s, toSort LIST.CREATE s, by

LIST.ADD toSort, "Goldmann", "Göbel", "Goethe", "Göthe", "Götz" PRINT "List of German words to sort"

DEBUG.DUMP.LIST toSort

LIST.TOARRAY toSort, mem$[] % Copy the content of the toSort list LIST.ADD.ARRAY by, mem$[] % into the by list.

PRINT "German DIN 5007 variant 1; used for words, such as in dictionaries" LIST.SORT.BY , toSort, by, , "de", "\_Secondary"

DEBUG.DUMP.LIST toSort LIST.SIZE by, lS

PRINT "German DIN 5007 variant 2; special sorting for name lists, "; PRINT "for example in telephone directories"

FOR i = 1 TO lS

LIST.GET by, i, str$ str$ = LOWER$(str$)

str$ = REPLACE$(str$, "ä", "ae") str$ = REPLACE$(str$, "ö", "oe") str$ = REPLACE$(str$, "ü", "ue") LIST.REPLACE by, i, str$

NEXT

LIST.CLEAR toSort % Refresh the toSort list LIST.ADD.ARRAY toSort, mem$[] % with the original content. LIST.SORT.BY , toSort, by, , "de", "\_Secondary" DEBUG.DUMP.LIST toSort

PRINT "Austrian telephone directory sorting" LIST.CLEAR by % Refresh the toSort list LIST.ADD.ARRAY by, mem$[] % with the original content. LIST.SIZE by,lS

FOR i = 1 TOlS

LIST.GET by, i, str$ str$ = LOWER$(str$)

str$ = REPLACE$(str$, "ä", "az ") str$ = REPLACE$(str$, "ö", "oz ") str$ = REPLACE$(str$, "ü", "uz ") str$ = REPLACE$(str$, "ß", "ssz ") str$ = REPLACE$(str$, "st.", "sankt ") LIST.REPLACE by, i,str$

NEXT

LIST.CLEAR toSort LIST.ADD.ARRAY toSort, mem$[]

LIST.SORT.BY , toSort, by, , "de", "\_Secondary" DEBUG.DUMP.LIST toSort

Example how to use a sorting index:

ARRAY.LOAD x[],10,2,9,7,-8,99,4,7,1,4,-13

ARRAY.LOAD y[],-8,99,4,7,1,4,-13,10,2,9,7

ARRAY.LOAD z[],10,2,9,1,4,-13,7,-8,99,4,7

PRINT "In this case the source data have not to be copied or changed." PRINT " X", " Y", " Z", "sorted by Y"

LIST.CREATE n, byY LIST.ADD.ARRAY byY, y[]

LIST.CREATE n, idxList LIST.SORT.BY idxList, , byY LIST.TOARRAY idxList, idx[] ARRAY.LENGTH aL, idx[] FOR i = 1 TO aL

IF i = 1 THEN PRINT "The min: "; IF i = aL THEN PRINT "The max: ";

PRINT x[idx[i]], y[idx[i]], z[idx[i]] NEXT

##### List.map.2d <pointer\_nexp>, {{{{{{{{dx1},dy1},agl1},dx2},dy2},agl2},mulx},muly}

All Arguments are of the type <nexp>.

Maps results of 2D operations by translation 1, rotation 1, translation 2, rotation 2 and multiplication (in this order) back into a given **x/y** value list.

This covers different cases of a combined workflow, which is often needed.

The arguments for translation are dx1, dy1, dx2 and dy2 (mostly in longitudinal units). The arguments for rotation are agl1 and agl2 in degrees.

The arguments for multiplication are mulx and muly.

The default arguments for translation and rotation are set to 0. The default arguments for multiplication are set to 1.

For a complete 2D operation you need the x and the y value, in this consequence an even number of values.

Is the number of the list items not even, the last value is only computed by dx1, dx2 and mulx. So you can use also 1D value lists if the x/y arguments are equal and agl1 respectively agl2 are zero.

If you use **Screen Coordinates** you get a **clockwise** rotation.

If you use **World Coordinates** respectively the **Right-Hand Rule** you a get **counter- clockwise** rotation or known as a mathematically positive rotation direction.

\*\*\*\*\*

Все аргументы имеют тип <nexp>.

Отображает результаты 2D-операций с помощью перевода 1, вращения 1, перевода 2, вращения 2 и умножения (в этом порядке) обратно в заданный список значений x / y.

Это охватывает различные случаи комбинированного рабочего процесса, который часто необходим.

Аргументами для перевода являются dx1, dy1, dx2 и dy2 (в основном в продольных единицах). Аргументы для поворота - agl1 и agl2 в градусах.

Аргументами для умножения являются mulx и muly.

Аргументы по умолчанию для перевода и поворота установлены в 0. Аргументы по умолчанию для умножения установлены в 1.

Для полной 2D операции вам нужны значения x и y, в результате чего четное число значений.

Если число элементов списка не является четным, последнее значение вычисляется только по dx1, dx2 и mulx. Таким образом, вы можете также использовать списки значений 1D, если аргументы x / y равны и agl1 соответственно agl2 равны нулю.

Если вы используете координаты экрана, вы получаете вращение по часовой стрелке.

Если вы используете Мировые координаты, соответственно Правило правой руки, вы получаете вращение против часовой стрелки или известно как математически положительное направление вращения.

Example:

List.create n, l2d

List.add l2d, 0, 1, 1, 1, 7 %The number of list items is not even!

List.map.2d l2d, 0,0, 45, 0,0, 0, 2,2

Debug.on

Debug.dump.list l2d % Look what happens with the 7.

See also Array.Mat.Transpose, Array.Math

##### List.map.3d <pointer\_nexp>,

**{{{{{{{{{{{{{{{dx1},dy1},dz1},agl1x},agl1y},agl1z},dx2},dy2},dz2},agl2x},agl2y},agl2z}, mulx},muly},mulz}**

All Arguments are of the type <nexp>.

Maps results of 3D operations by translation 1, rotation 1, translation 2, rotation 2 and multiplication (in this order) back into a given **x/y/z** value list.

This covers different cases of a combined workflow, which is often needed.

The arguments for translation are dx1, dy1,dz1, dx2, dy2 and dz2 (mostly in longitudinal units).

The arguments for rotation are agl1x, agl1y, agl1z, agl2x , agl2y and agl2z in degrees. The arguments for multiplication are mulx, muly and mulz.

The default arguments for translation and rotation are set to 0. The default arguments for multiplication are set to 1.

For a complete 3D operation you need the x, y and the z value, in this consequence the division from the number of list items by 3 has to be an integer.

If the list has one or two items more, this items are in **opposite** to List.map.2d **not changed**.

If you use **Screen Coordinates** you get a **clockwise** rotation.

If you use **World Coordinates** respectively the **Right-Hand Rule** you a **get counter- clockwise** rotation or known as a mathematically positive rotation direction.

\*\*\*\*\*

Все аргументы имеют тип <nexp>.

Отображает результаты 3D-операций с помощью перевода 1, вращения 1, перевода 2, вращения 2 и умножения (в этом порядке) обратно в заданный список значений x / y / z.

Это охватывает различные случаи комбинированного рабочего процесса, который часто необходим.

Аргументы для перевода: dx1, dy1, dz1, dx2, dy2 и dz2 (в основном в продольных единицах).

Аргументами для поворота являются agl1x, agl1y, agl1z, agl2x, agl2y и agl2z в градусах. Аргументами для умножения являются mulx, muly и mulz.

Аргументы по умолчанию для перевода и поворота установлены в 0. Аргументы по умолчанию для умножения установлены в 1.

Для полной трехмерной операции вам нужны значения x, y и z, поэтому деление числа элементов списка на 3 должно быть целым числом.

Если в списке есть один или два элемента больше, эти элементы в отличие от List.map.2d не изменяются.

Если вы используете координаты экрана, вы получаете вращение по часовой стрелке.

Если вы используете Мировые координаты, соответственно Правило правой руки, вы получаете вращение против часовой стрелки или известно как математически положительное направление вращения.

See also Array.Mat.Transpose, Array.Math

Example:

List.create n, l3d

List.add l3d, 0, 1, 0, 1, 1, 0, 7 % The division of the number of list items

% by 3 is not an integer.

Array.load t3d1[], 0,0,0 % Translation 1[x,y,z]

Array.load r3a1[], 0,0,45 % Rotation 1 in degrees around [x,y,z]-Axis Array.load t3d2[], 0,0,0 % Translation 2 [x,y,z]

Array.load r3a2[], 0,0,0 % Rotation 2 in degrees around [x,y,z]-Axis Array.load m3d[], 1,1,1 % Multiplication[x,y,z]

List.map.3d l3d, t3d1[1],t3d1[2], t3d1[3], r3a1[1], r3a1[2], r3a1[3], ~

t3d2[1],t3d2[2], t3d2[3], r3a2[1], r3a2[2], r3a2[3], ~ m3d[1], m3d[2], m3d[3]

Debug.on

Debug.dump.list l3d % Look what happens with the 7.

### SQLITE Command Enhancements

Also good sources are at [http://www.sqlitetutorial.net](http://www.sqlitetutorial.net/)and https://[www.sqlite.org/faq.html#q2.](http://www.sqlite.org/faq.html#q2)

##### Sql.new\_table <DB\_pointer\_nvar>, <table\_name\_sexp>, C1$, C2$, ...,CN$Sql.new\_table <DB\_pointer\_nvar>, <table\_name\_sexp>, <delim\_row\_sexp>

A single database may contain many tables. A table is made of rows of data. A row of data consists of columns of values. Each value column has a column name associated with it.

This command creates a new table with the name <table\_name\_sexp> in the referenced opened database, but only if the table does not exist. The column names for that table are defined by the following: C1$, C2$, ..., CN$. At least one column name is required. You may create as many column names as you need.

BASIC! always adds a Row Index Column named "\_id" to every table. The value in this Row Index Column is automatically incremented by one for each new row inserted. This gives each row in the table a unique identifier. This identifier can be used to connect information in one table to another table. For example, the \_id value for customer information in a customer table can be used to link specific orders to specific customers in an outstanding order database.

The alternative <delim\_row\_sexp> is a string expression starting with "\_Delimiter:" + one

**d**elimiter **c**haracter as a numeric string + "**;**"+ C1$ + dC$ + C1$ + … + dC$ + CN$

Using other delimiter characters is a little advantage. It is possible to program the number of columns.

\*\*\*\*\*

Одна база данных может содержать много таблиц. Таблица состоит из строк данных. Строка данных состоит из столбцов значений. С каждым столбцом значений связано имя столбца.

Эта команда создает новую таблицу с именем <table\_name\_sexp> в указанной открытой базе данных, но только в том случае, если таблица не существует. Имена столбцов для этой таблицы определяются следующим образом: C1 $, C2 $, ..., CN $. Требуется хотя бы одно имя столбца. Вы можете создать столько имен столбцов, сколько вам нужно.

BASIC! всегда добавляет столбец индекса строки с именем "\_id" к каждой таблице. Значение в этом столбце индекса строки автоматически увеличивается на единицу для каждой новой вставленной строки. Это дает каждой строке в таблице уникальный идентификатор. Этот идентификатор может использоваться для связи информации в одной таблице с другой таблицей. Например, значение \_id для информации о клиенте в таблице клиентов может использоваться для связи конкретных заказов с конкретными клиентами в базе данных ожидающих заказов.

Альтернативой <delim\_row\_sexp> является строковое выражение, начинающееся с "\_Delimiter:" + one

символ разделителя в виде числовой строки + ";" + C1 $ + dC $ + C1 $ +… + dC $ + CN $

Использование других символов-разделителей - небольшое преимущество. Можно запрограммировать количество столбцов.

Example:

delimN = ucode("€") dC$ = chr$(delimN)

head$ = "\_Delimiter:" + int$(delimN) + ";"

columns$ = head$ + c1$ + dC$ + c2$ + dC$ + c3$ + dC$ + c4$ SQL.NEW\_TABLE DB\_Ptr, tbname$, columns$

##### Sql.drop\_table <DB\_pointer\_nvar>, <table\_name\_sexp>

The table named <table\_name\_sexp> in the opened database pointed to by

<DB\_pointer\_nvar> will be dropped (deleted) from the database if the table exists.

##### Sql.insert<DB\_poinnameter\_nvar>,<table sexp>, C1$, V1$, C2$, V2$, ..., CN$,VN$

**Sql.insert <DB\_pointer\_nvar>, <table\_name\_sexp>, <delim\_row\_sexp>** Inserts a new row of data columns and values into a table in a previously opened database.

The <table\_name\_sexp> is the name of the table into which the data is to be inserted.All newly inserted rows are inserted after the last, existing row of the table.

C1$, V1$, C2$, V2$, ..., CN$, VN$: The column name and value pairs for the new row. These parameters must be in pairs. The column names must match the column names used to create the table. Note that the values are all strings. When you need a numeric value for a column, use the BASIC! STR$(n) to convert the number into a string. You can also use the BASIC! FORMAT$(pattern$, N) to create a formatted number for a value. (The Values-as-strings requirement is a BASIC! SQL Interface requirement, not a SQLite requirement. While SQLite, itself, stores all values as strings, it provides transparent conversions to other data types. I have chosen not to complicate the interface with access to these SQLite conversions since BASIC! provides its own conversion capabilities.)

The alternative <delim\_row\_sexp> is a string expression starting with "\_Delimiter:" + one **d**elimiter **c**haracter as a numeric string + "**;**" + C1$ + dC$ + V1$ + ..., CN$ + dC$ + VN$ Using other delimiter characters is a little advantage. It is possible to program the number of columns.

\*\*\*\*\*

новая строка столбцов данных и значений в таблицу в ранее открытой базе данных.

<Table\_name\_sexp> - это имя таблицы, в которую должны быть вставлены данные. Все вновь вставленные строки вставляются после последней существующей строки таблицы.

C1 $, V1 $, C2 $, V2 $, ..., CN $, VN $: пары имен и значений для новой строки. Эти параметры должны быть в парах. Имена столбцов должны совпадать с именами столбцов, использованными для создания таблицы. Обратите внимание, что значения являются строками. Когда вам нужно числовое значение для столбца, используйте BASIC! STR $ (n), чтобы преобразовать число в строку. Вы также можете использовать бейсик! FORMAT $ (pattern $, N) для создания отформатированного числа для значения. (Требование Values-as-strings - это требование BASIC! SQL Interface, а не требование SQLite. Хотя SQLite сам хранит все значения в виде строк, он обеспечивает прозрачные преобразования в другие типы данных. Я решил не усложнять интерфейс с помощью доступ к этим преобразованиям SQLite, поскольку BASIC! предоставляет свои собственные возможности преобразования.)

Альтернативой <delim\_row\_sexp> является строковое выражение, начинающееся с "\_Delimiter:" + один символ-разделитель в виде числовой строки + ";" + C1 $ + dC $ + V1 $ + ..., CN $ + dC $ + VN $ Использование других символов-разделителей - небольшое преимущество. Можно запрограммировать количество столбцов.

Example:

delimN = ucode("€") dC$ = chr$(delimN)

head$ = "\_Delimiter:" + int$(delimN) + ";"

row$ = head$ + c1$ + dC$ + fn$ + dC$ + c2$ + dC$ + ln$ + dC$ + c3$ + dC$ + ga$ + ~ dC$ + c4$ + dC$ + pa$

SQL.INSERT DB\_Ptr, tbname$, row$

##### Sql.update <DB\_pointer\_nvar>, <table\_name\_sexp>, C1$, V1$, C2$, V2$,...,CN$, VN$

**{: <where\_sexp>}**

**Sql.update <DB\_pointer\_nvar>, <table\_name\_sexp>, <delim\_row\_sexp> {:**

**<where\_sexp>}**

In the named table of a previously opened database, change column values in specific rows selected by the Where$ parameter <where\_sexp>. The C$,V$ parameters must be in pairs. The colon character terminates the C$,V$ list and must precede the Where$ in this command. The Where$ parameter and preceding colon are optional.

BASIC! also uses the colon character to separate multiple commands on a single line. The use of a colon in this command conflicts with that feature. Use caution when using both together.

If you put a colon on a line after this command, the preprocessor **always** assumes the colon is part of the command and not a command separator. If you are not certain of the outcome, the safest action is to put the **Sql.update** command on a line by itself, or at the end of a multi-command line.

The alternative <delim\_row\_sexp> is a string expression starting with "\_Delimiter:" + one **d**elimiter **c**haracter as a numeric string + "**;**" + C1$ + dC$ + V1$ + ..., CN$ + dC$ + VN$ Using other delimiter characters is a little advantage and programming the number ofthe parameter in pairs ispossible.

\*\*\*\*\*

В именованной таблице ранее открытой базы данных измените значения столбцов в определенных строках, выбранных параметром Where $ <where\_sexp>. Параметры C $, V $ должны быть в парах. Символ двоеточия завершает список C $, V $ и должен предшествовать символу $ в этой команде. Параметр Where $ и предшествующее двоеточие являются необязательными.

BASIC! также использует символ двоеточия для разделения нескольких команд в одной строке. Использование двоеточия в этой команде конфликтует с этой функцией. Будьте осторожны при использовании обоих вместе.

Если после этой команды в строку ставится двоеточие, препроцессор всегда предполагает, что двоеточие является частью команды, а не разделителем команд. Если вы не уверены в результате, самое безопасное действие - поместить команду Sql.update в отдельную строку или в конец многокомпонентной строки.

Альтернативой <delim\_row\_sexp> является строковое выражение, начинающееся с "\_Delimiter:" + один символ-разделитель в виде числовой строки + ";" + C1 $ + dC $ + V1 $ + ..., CN $ + dC $ + VN $ Использование других символов-разделителей является небольшим преимуществом, и программирование числа параметров в парах возможно.

Example:

delimN = ucode("€") dC$ = chr$(delimN)

head$ = "\_Delimiter:" + int$(delimN) + ";"

Where$ = "first\_name = 'Tamasin' AND last\_name = 'Washington' "

! SQL.UPDATE DB\_Ptr, tbname$, c3$, "94": Where$ **or**

row$= head$ + c3$ + de$ + "94"

SQL.UPDATE DB\_Ptr, tbname$, row$: Where$

##### Sql.query <cursor\_nvar>, <DB\_pointer\_nvar>, <table\_name\_sexp>,

**<columns\_sexp> {, <where\_sexp> {, <order\_sexp>} }**

Queries a table of a previously-opened database for some specific data. The command returns a Cursor named <Cursor\_nvar> to be used in stepping through Query results. The <columns\_sexp> is a string expression with a list of the names of the columns to be returned. The column names must be separated by commas. An example is Columns$ =

"First\_name, Last\_name, Sex, Age". If you want to get the automatically incremented Row Index Column then include the "\_id" column name in your column list. Columns may be listed in any order. The column order used in the query will be the order in which the rows are returned.

The optional <where\_sexp> is an SQL expression string used to select which rows to return. In general, an SQL expression is of the form <Column Name> <operator> <Value>. For example, Where$ = "First\_name = 'John' " Note that the Value must be contained in single quotes. Full details about the SQL expressions can be found **[here](http://www.sqlite.org/lang_expr.html)**. If the Where parameter is omitted, all rows will be returned.

The optional <order\_sexp> specifies the order in which the rows are to be returned. It identifies the column upon which the output rows are to be sorted. It also specifies whether the rows are to be sorted in ascending (ASC) or descending (DESC) order. For example, Order$ = "Last\_Name ASC" would return the rows sorted by Last\_Name from A to Z. If the Order parameter is omitted, the rows are not sorted.

The result-sorting feature natively sorts upper and lower case separately, which is often not what is desired. However there is a feature that joins upper and lower case letters alphabetically in the sort. This can be invoked by replacing the "ASC" or "DESC" (ascending or descending) part of the <order\_sexp> argument with "COLLATE NOCASE ASC" OR "COLLATE NOCASE DESC" respectively. For example, Order$ = "Last\_Name COLLATE NOCASE ASC" would return the rows sorted by Last\_Name from A to Z also if Last\_Name starts with lower case like "van der Meer".

COLLATE UNICODE sorts in Unicode order and COLLATE LOCALIZED in order of the current database language setting. If not specified by Sql.set\_locale the current system language setting is used.

If the Order parameter is present, the Where parameter must be present. If you wantto return all rows, just set Where$ =""

\*\*\*\*\*

Запрашивает таблицу ранее открытой базы данных для некоторых конкретных данных. Команда возвращает курсор с именем <Cursor\_nvar>, который будет использоваться при пошаговом выполнении результатов запроса. <Columns\_sexp> является строковым выражением со списком имен столбцов, которые должны быть возвращены. Имена столбцов должны быть разделены запятыми. Пример - столбцы $ =

«Имя, Фамилия, Пол, Возраст». Если вы хотите получить автоматически увеличенный столбец индекса строки, включите имя столбца «\_id» в список столбцов. Столбцы могут быть перечислены в любом порядке. Порядок столбцов, используемый в запросе, будет порядок, в котором строки возвращаются.

Необязательный <where\_sexp> - это строка выражения SQL, используемая для выбора возвращаемых строк. В общем случае выражение SQL имеет форму <имя столбца> <оператор> <значение>. Например, где $ = "First\_name = 'John'" Обратите внимание, что значение должно быть заключено в одинарные кавычки. Полную информацию о выражениях SQL можно найти здесь. Если параметр Where не указан, будут возвращены все строки.

Необязательный <order\_sexp> указывает порядок, в котором должны быть возвращены строки. Он идентифицирует столбец, по которому должны быть отсортированы выходные строки. Он также указывает, следует ли сортировать строки в порядке возрастания (ASC) или убывания (DESC). Например, Order $ = "Last\_Name ASC" вернет строки, отсортированные по Last\_Name от A до Z. Если параметр Order не указан, строки не сортируются.

Функция сортировки результатов изначально сортирует верхний и нижний регистры отдельно, что часто не то, что нужно. Однако есть функция, которая объединяет прописные и строчные буквы в алфавитном порядке в сортировке. Это может быть вызвано заменой части «ASC» или «DESC» (восходящей или нисходящей) аргумента <order\_sexp> на «COLLATE NOCASE ASC» ИЛИ «COLLATE NOCASE DESC» соответственно. Например, Order $ = "Last\_Name COLLATE NOCASE ASC" вернет строки, отсортированные по Last\_Name от A до Z, также, если Last\_Name начинается с нижнего регистра, например "van der Meer".

COLLATE UNICODE сортирует в порядке Юникода и COLLATE LOCALIZED в порядке текущей настройки языка базы данных. Если Sql.set\_locale не указан, используется текущая настройка языка системы.

Если параметр Order присутствует, параметр Where должен присутствовать. Если вы хотите вернуть все строки, просто установите Where $ = ""

##### Sql.set\_locale <DB\_pointer\_nvar>, <locale\_sexp>

The <locale\_sexp> secified the output based on language and region. The localespecifies the language and region with standardized codes. The <locale\_sexp> is a string containing zero or more codes separated byunderscores.

The function accepts up to three codes. The first must be a language code, such as "en", "de" or "ja".

The second must be a region or country code, such as "FR", "US", or "IN". Some language and country combinations can accept a third code, called the "variant code".

The function also accepts the standard three-letter codes and numeric codes for country or region. For example, "fr\_FR", "fr\_FRA", and "fr\_250" are all equivalent.

The locale take only effect, if COLLATE LOCALIZED is part of the Sql.query <order\_sexp> statement. You may need to install your chosen language on the Android device first to get the full effect.

\*\*\*\*\*

<Locale\_sexp> обеспечивает вывод на основе языка и региона. Locales определяет язык и регион с помощью стандартизированных кодов. <Locale\_sexp> - это строка, содержащая ноль или более кодов, разделенных символами подчеркивания.

Функция принимает до трех кодов. Первым должен быть код языка, такой как «en», «de» или «ja».

Вторым должен быть код региона или страны, например «FR», «US» или «IN». Некоторые комбинации языков и стран могут принимать третий код, называемый «код варианта».

Функция также принимает стандартные трехбуквенные коды и цифровые коды для страны или региона. Например, «fr\_FR», «fr\_FRA» и «fr\_250» все эквивалентны.

Локаль вступает в силу только в том случае, если COLLATE LOCALIZED является частью оператора Sql.query <order\_sexp>. Возможно, вам придется сначала установить выбранный язык на устройстве Android, чтобы получить полный эффект.

Example:

Sql.set\_locale DB\_Ptr, "de\_CH"

##### Sql.exec <DB\_pointer\_nvar>, <command\_sexp>

ExecuteANY non-query SQL command string ("CREATE TABLE", "DELETE", "INSERT", etc.) using a previously opened database.

\*\*\*\*\*

Выполните команду SQL без запроса SQL («CREATE TABLE», «DELETE», «INSERT» и т. Д.) С использованием ранее открытой базы данных.

Example:

tbname$ = "Birthdates"

mColumns$ = "FirstName TEXT, Name TEXT "

CommandString$ = "CREATE TABLE IF NOT EXISTS " + tbname$ + "( " ~

+ "\_id INTEGER PRIMARY KEY AUTOINCREMENT, "+ mColumns$ + " )"

Sql.exec DB\_Ptr, CommandString$ % Create a table with two columns CommandString$ = "ALTER TABLE "+tbname$+" ADD " + "Birthdate TEXT" Sql.exec DB\_Ptr, CommandString$ % Add a coumn

CommandString$ = "ALTER TABLE " + tbname$ + " RENAME TO " + "SavedData" Sql.exec DB\_Ptr,CommandString$

CommandString$ = "VACUUM" % Rebuild the database to smaller size. Sql.exec DB\_Ptr,CommandString$

##### Sql.raw\_query <cursor\_nvar>, <DB\_pointer\_nvar>, <query\_sexp>

Execute ANY SQL Query command using a previously opened database and return a Cursor for the results.

\*\*\*\*\*

Выполните команду ЛЮБОЙ SQL-запрос с использованием ранее открытой базы данных и верните курсор для результатов.

Example:

sql$ = "SELECT sql FROM sqlite\_master WHERE tbl\_name = '"+tbname$+"' AND type = 'table';" SQL.RAW\_QUERY mCursor, DB\_Ptr, sql$

SQL.NEXT xdone, mCursor, rawTableSql$ LIST.CREATE s, fieldNames

IF rawTableSql$ = ""

PRINT "No database table with this name found!" ELSE

SPLIT.ALL a$[], rawTableSql$, "[()]"

SPLIT.ALL b$[], a$[2], "," ARRAY.LENGTH al, b$[] FOR i = 2 TO al SPLIT.ALL c$[], b$[i], " "

LIST.ADD fieldNames, c$[2] NEXT

ENDIF

##### SQL.PING <result\_nvar>, <DB\_pointer\_nvar> {,<table\_name\_sexp>

**{,<column\_name\_sexp>}}**

1. SQL.PINGr,db % ping only thedatabase
2. SQL.PING r,db,"mytable" % ping atable,
3. SQL.PING r, db,"mytable","col1" % ping a table and column Returns the size (number of rows) of a database or table tor.

In case (A), this is the total number of user tables in the db. The return code is as follows;

\*\*\*\*\*

Возвращает размер (количество строк) базы данных или таблицы tor.

В случае (A) это общее количество пользовательских таблиц в БД. Код возврата выглядит следующим образом;

|  |  |
| --- | --- |
| -1 | The table name does not exist |
| -2 | The table name exists but column does not exist in the table |
| ≥0 | The total number of tables in the database or rows in the table. |

In case (C), this also means both table and column exists.

\*\*\*\*\*

В случае (C) это также означает, что существует таблица и столбец.

[ack:humpty0250]

### File Handling

##### A new feature is the ability to cross the directory borders of BASIC! with the URI start"file://".

##### \*\*\*\*\*

##### Новая функция - возможность пересекать границы каталога BASIC! с началом URI "file: //".

Example:

FILE.ROOT dataPath$, "\_Dcim" fn$ = "file://" + dataPath$

About Android Directories

A distinction is made between **internal** and **external** directories.

**Internal** directories are owned by the App, their indicator is the App’s package name (e.g. com.rfo.basic) in the absolute path. If the App is be deinstalled **all** the **internal** directories are deleted, too. The exceptions are **internal** directories upon **removable** SD cardsbefore Android 4.4 KitKat (API19). This BASIC! version creates a **private internal** directory at firststart.

**External** directories are member of the **public** part of Android’s file system. From the view of the app it is controlled **externally**. So you need permissions finally in the file AndroidManifest.xml to access **external** directories. Please take a look at the compiler options, if you want to create an App.

The terms sd-ext or extSdcard are a case for a misunderstanding. These are expressions for **removable** SD cards.

**Removable** SD cards or USB sticks are normally **external** directories, too.

A bit odd in conjunction with this logic is the possibility to create an **internal** directory upon an **external** directory and sometimes also on a **removable** data carrier.

**Important: Internal** directories on **removable** data carrier are **public** and automatically created by Android! **Starting** with **Android 10** these are **private**.

If you install the BASIC! Interpreter you get an **external public** directory rfo-basic.

If you deinstall BASIC! this directory and its contents is not removed. Only standard files like the code samples are changed after reinstalling.

If you create an APK with assets, read the article in the manual appendix about handling files from APKs and Resources (manual page 225).

Note, that you get not in all cases direct access to the files in Assets, because they are converted under certain circumstances. Maybe using sub directories is one of them.

It seems, that RFO-BASIC 1.91 has the same issue. If you rename fly.gif in the data folder you should get access to the asset data folder, but no access!

In this case you have to copy these files to the different cache directory with Byte.open.

\*\*\*\*\*

О каталогах Android

Различают внутренние и внешние каталоги.

Внутренние каталоги принадлежат приложению, их индикатор - это имя пакета приложения (например, com.rfo.basic) в абсолютном пути. Если приложение будет удалено, все внутренние каталоги также будут удалены. Исключением являются внутренние каталоги на съемных SD-картах до Android 4.4 KitKat (API19). Это бейсик! версия создает личный внутренний каталог при первом запуске.

Внешние каталоги являются частью публичной части файловой системы Android. С точки зрения приложения это контролируется извне. Таким образом, вам нужны разрешения, наконец, в файле AndroidManifest.xml для доступа к внешним каталогам. Пожалуйста, обратите внимание на параметры компилятора, если вы хотите создать приложение.

Термины sd-ext или extSdcard являются причиной недопонимания. Это выражения для съемных SD-карт.

Съемные SD-карты или USB-накопители также обычно являются внешними каталогами.

Немного странным в сочетании с этой логикой является возможность создания внутреннего каталога на внешнем каталоге, а иногда и на съемном носителе данных.

Важно: внутренние каталоги на съемном носителе являются общедоступными и автоматически создаются Android! Начиная с Android 10 это частные.

Если вы установите бейсик! Интерпретатор вы получаете внешний публичный каталог rfo-basic.

Если вы деинсталлируете BASIC! этот каталог и его содержимое не удаляются. Только стандартные файлы, такие как примеры кода, меняются после переустановки.

Если вы создаете APK с ресурсами, прочитайте статью в приложении к руководству об обработке файлов из APK и ресурсов (страница руководства 225).

Обратите внимание, что не во всех случаях вы получаете прямой доступ к файлам в активах, поскольку они конвертируются при определенных обстоятельствах. Может быть, использование подкаталогов является одним из них.

Похоже, что RFO-BASIC 1.91 имеет ту же проблему. Если вы переименуете fly.gif в папку с данными, вы должны получить доступ к папке с данными об активах, но без доступа!

В этом случае вы должны скопировать эти файлы в другой каталог кэша с помощью Byte.open.

### File.root <full\_path\_svar>{, <dirType\_sexp>}

Returns the canonical path from the file system root to the default "<pref base drive>/rfo- basic/data", the default data directory, in **<full\_path\_svar>**. The <pref\_base\_drive> is expanded to the full absolute path from the file system root, "/".

The system constants in **<dirType\_sexp>** enables easy access to BASIC! and other folders:

\_Alarms, \_App, \_AppPath, \_Asset\_Cache, \_BasicSystem, \_Bluetooth, \_Cache,

\_Data, \_Database, \_Dcim, \_Documents, \_Downloads, \_External, \_Internal,

\_InternalOnExternal, \_InternalOnSdRemovable\*, \_Mnt, \_Movies, \_Music,

\_Notifications, \_Pictures, \_Podcasts, \_ProgramPath, \_Ringtones,

\_SdRemovable\*, \_Service, \_Source, \_SourceSamples, \_ScreenShots, \_Storage,

\_System

\* Full available with Android 4.4 KitKat (API19) and later. At sooner APIs the interpreter searches for sdcard1, sdcard2, extSdcard and sd-ext.

If an \_Internal… folder is not been created, it will create by the first call. The folder

\_Documents is not created on some devices. With File.mkdir you can do it yourself.

If you use the internal cache folder (\_Cache) this files will be ones that get deleted first, if the device runs low on storage. There is no indication when these files will be deleted. Note: you should not rely on the system deleting these files for you; you should always have a reasonable maximum, such as 1 MB, for the amount of space you consume with cache files, and prune those files when exceeding that space.

The \_Asset\_Cache folder as part of the \_Cache folder will be created at calling Byte.open myFileTable, "asset://" + myFileNamePath$ if the folder does not exists.

Note: **"file://"** + <full\_path\_svar> is required if you want to use it directly in other BASIC! commands.

\*\*\*\*\*

Возвращает канонический путь от корня файловой системы к каталогу по умолчанию «<pref base drive> / rfo-basic / data», каталогу данных по умолчанию, в <full\_path\_svar>. <Pref\_base\_drive> раскрывается до полного абсолютного пути от корня файловой системы, "/".

Системные константы в <dirType\_sexp> обеспечивают легкий доступ к BASIC! и другие папки:

\_Alarms, \_App, \_AppPath, \_Asset\_Cache, \_BasicSystem, \_Bluetooth, \_Cache,

\_Data, \_Database, \_Dcim, \_Documents, \_Downloads, \_External, \_Internal,

\_InternalOnExternal, \_InternalOnSdRemovable \*, \_Mnt, \_Movies, \_Music,

\_Notifications, \_Pictures, \_Podcasts, \_ProgramPath, \_Ringtones,

\_SdRemovable \*, \_Service, \_Source, \_SourceSamples, \_ScreenShots, \_Storage,

\_System

\* Полная версия доступна с Android 4.4 KitKat (API19) и выше. В более ранних API интерпретатор ищет sdcard1, sdcard2, extSdcard и sd-ext.

Если папка \_Internal ... не была создана, она будет создана при первом вызове. Папка

\_Documents не создается на некоторых устройствах. С File.mkdir вы можете сделать это самостоятельно.

Если вы используете папку внутреннего кэша (\_Cache), это будут те файлы, которые будут удалены первыми, если на устройстве недостаточно места для хранения. Нет указания, когда эти файлы будут удалены. Примечание: вы не должны полагаться на систему, удаляющую эти файлы за вас; у вас всегда должен быть разумный максимум, например, 1 МБ, для объема пространства, которое вы используете для файлов кэша, и обрезать эти файлы при превышении этого пространства.

Папка \_Asset\_Cache как часть папки \_Cache будет создана при вызове Byte.open myFileTable, «asset: //» + myFileNamePath $, если папка не существует.

Примечание: "file: //" + <full\_path\_svar> требуется, если вы хотите использовать его непосредственно в другом BASIC! команды.

Example:

FILE.ROOT dataPath$,"\_Documents" fn$ = "file://" + dataPath$ FILE.EXISTS ok,fn$

IF ok = 0

FILE.ROOT newFolder$, "\_External"

newFolder$ = "file://" + newFolder$ + "/Documents" FILE.MKDIR newFolder$

ENDIF

**Experimental:** RFO-BASIC and hBASIC ignore the slash at the beginning, because it is **not needed.**

If you want to use an absolute path in OliBasic your path$ should begin with "file://" + ... In the new OliBasic version a **slash** at the path beginning has the same function as "file://" as well as Android it does.

Thus file.root mPath$ returns an absolute path, which can be used directly like this: mPath$ = mPath$ + "/" + "cartman.png". **If you get in trouble use** "**file://**" in case of absolute file or directory paths.

\*\*\*\*\*

Эксперимент: RFO-BASIC и hBASIC игнорируют косую черту в начале, потому что она не нужна.

Если вы хотите использовать абсолютный путь в OliBasic, ваш путь $ должен начинаться с «file: //» + ... В новой версии OliBasic косая черта в начале пути выполняет ту же функцию, что и «file: //». как Android это делает.

Таким образом, file.root mPath $ возвращает абсолютный путь, который можно использовать напрямую следующим образом: mPath $ = mPath $ + "/" + "cartman.png". Если у вас возникли проблемы, используйте «file: //» в случае абсолютных путей к файлам или каталогам.

### File.exists <lvar>, <path\_sexp>

Reports if the <path\_sexp> directory or file exists. If the directory or file does not exist, the

<lvar> will contain zero. If the file or directory does exist, the <lvar> will be returned as non-zero. If the file or directory is readable <lvar> returns 2.0. If the file or directory is writeable <lvar> returns 3.0. If the file or directory is readable and writeable <lvar> returns 4.0.

The default path is "<pref base drive>/rfo-basic/data/". It is also possible to use an URI with "file://" as start.

Sometimes you get a file path like “/external/images/media/556”.

With "file://" + “/external/images/media/556”, File.exists is able to handle it.

In this case <path\_sexp> returns a new absolute file path, **but only** if <path\_sexp> is a

**single value** like fn$ and **not** a simple string expression.

File.exists has no access to **assets** and **res**ources in conjunction with APKs.

\*\*\*\*\*

Сообщает, существует ли каталог или файл <path\_sexp>. Если каталог или файл не существует,

<lvar> будет содержать ноль. Если файл или каталог существует, <lvar> будет возвращен как ненулевой. Если файл или каталог доступен для чтения, <lvar> возвращает 2.0. Если файл или каталог доступны для записи, <lvar> возвращает 3.0. Если файл или каталог доступны для чтения и записи, <lvar> возвращает 4.0.

Путь по умолчанию: «<pref base drive> / rfo-basic / data /». Также можно использовать URI с «file: //» в качестве начала.

Иногда вы получаете путь к файлу, например «/ external / images / media / 556».

С помощью «file: //» + «/ external / images / media / 556» File.exists может справиться с этим.

В этом случае <path\_sexp> возвращает новый абсолютный путь к файлу, но только если <path\_sexp> является

одно значение типа fn $, а не простое строковое выражение.

File.exists не имеет доступа к ресурсам и ресурсам вместе с APK.

### File.md5 <nvar>, <path\_sexp>

Returns the MD5 hash of the file specified by <path\_sexp>.

\*\*\*\*\*

Возвращает MD5-хеш файла, указанного в <path\_sexp>.

### File.lastmodified <nvar>, <path\_sexp>

Returns the time when this file was last modified, measured in milliseconds since January 1st, 1970, midnight GMT.

\*\*\*\*\*

Возвращает время последнего изменения этого файла, измеренное в миллисекундах с 1 января 1970 года, в полночь по Гринвичу.

Example:

FILE.ROOT pp$ ,"\_SourceSamples"

FILE.LASTMODIFIED lm, "file://"+ pp$ + "/" +"f01\_commands.bas"

? USING$("", "%TF %tT", lm, lm) % Returns "2019-06-3021:01:01"

### File.absolute <absolute\_svar>, <path\_sexp>

Returns the absolute file path by <absolute\_svar>.

It tries to convert a document path beginning with “content://” into an absolute file path also. In this case min. KitKat 4.4 (API 19) is needed. If is ":open uri withADOC.Read or ADOC.Write" returned, use this ADOC commands for access.

\*\*\*\*\*

Возвращает абсолютный путь к файлу через <absolute\_svar>.

Он также пытается преобразовать путь документа, начинающийся с «content: //», в абсолютный путь к файлу. В этом случае мин. Требуется KitKat 4.4 (API 19). Если возвращено «: open uri withADOC.Read or ADOC.Write», используйте эти команды ADOC для доступа.

### File.dir <path\_sexp>, Array$[]{{{{,<dirmark\_sexp>},<timeStamp\_nexp>},

**<recursive\_nexp>}, <type\_sexp>}**

Returns the names of the files and directories in the path specified by <path\_sexp>. The path is relative to "<pref base drive>/rfo-basic/data/".

When you initially add "**asset://**", you will get the directories and files from the **APK** assets. Depending on the structure of the assets, the distinction between directory or file is only possible via the point. There is no opportunity to get file time stamps in this case.

Keep also in mind, that there are **no empty directories** in assets!

The names are placed into Array$[]. The array is sorted alphabetically with the directories at the top of the list. If the array exists, it is overwritten, otherwise a new array is created. The result is always a one-dimensional array.

A directory is identified by a marker appended to its name. The default marker is the string "(d)". You can change the marker with the optional directory mark parameter

<dirmark\_sexp>. If you do not want directories to be marked, set <dirmark\_sexp> to an empty string, "".

**Dir** is a valid alias for this command.

If the directory is empty, File.dir returns an array with one item and a string with one space (" ") in it.

Options of <timeStamp\_nexp>:

* + 0 no time stamp(default)
  + 1 with time stamp as time in milliseconds + ":" + file name, butunsorted
  + 2 with time stamp as time in milliseconds + ":" + file name, sorted in ascendingorder
  + 3 with time stamp as time in milliseconds + ":" + file name, sorted in descendingorder

If <recursive\_nexp> is > 0 all sub directories are searched as well. Default is 0.

If only files or directories are needed use <type\_sexp> with "\_F" for files and "\_D" for directories. Default is "\_DF".

\*\*\*\*\*

Возвращает имена файлов и каталогов по пути, указанному в <path\_sexp>. Путь относительно "<pref base drive> / rfo-basic / data /".

При первоначальном добавлении «asset: //» вы получите каталоги и файлы из ресурсов APK. В зависимости от структуры активов, различие между каталогом или файлом возможно только через точку. В этом случае нет возможности получить метки времени файла.

Помните также, что в активах нет пустых каталогов!

Имена помещаются в массив $ []. Массив отсортирован в алфавитном порядке с каталогами в верхней части списка. Если массив существует, он перезаписывается, в противном случае создается новый массив. Результатом всегда является одномерный массив.

Каталог идентифицируется маркером, добавленным к его имени. Маркер по умолчанию - это строка "(d)". Вы можете изменить маркер с помощью дополнительного параметра метки каталога

<Dirmark\_sexp>. Если вы не хотите, чтобы каталоги были помечены, установите <dirmark\_sexp> в пустую строку "".

Dir является допустимым псевдонимом для этой команды.

Если каталог пуст, File.dir возвращает массив с одним элементом и строку с одним пробелом ("").

Опции <timeStamp\_nexp>:

0 нет отметки времени (по умолчанию)

1 с меткой времени как время в миллисекундах + ":" + имя файла, но не отсортировано

2 с меткой времени как время в миллисекундах + ":" + имя файла, отсортированное по возрастанию

3 с меткой времени как время в миллисекундах + ":" + имя файла, отсортированное по убыванию

Если <recursive\_nexp> равен> 0, все подкаталоги также ищутся. По умолчанию 0.

Если нужны только файлы или каталоги, используйте <type\_sexp> с «\_F» для файлов и «\_D» для каталогов. По умолчанию "\_DF".

Example:

FILE.ROOT path$**,** "\_Mnt"

FILE.DIR path$**,** dirArray$[]**,** ""**,** 0 **,** 1 **,** "\_F" LIST.CREATE s, dirItems LIST.ADD.ARRAY dirItems, dirArray$[] LIST.CREATE s**,** dirFilteredItems

LIST.MATCH **,** dirFilteredItems**,** dirItems**,** "png"**,,,,** "\_Ends\_With\_IgnoreCase" DEBUG.ON

DEBUG.DUMP.LIST dirFilteredItems

### Byte.open {r|w|a}, <file\_table\_nvar>, <path\_sexp>

The file specified by the path string expression <path\_sexp> is opened. If the path is a URL starting with "http…" then an Internet file is opened. Otherwise, the <path\_sexp> string is appended to the default path "<pref base drive>/rfo-basic/data/".

If the URL starts with "asset://" + myFileNamePath$ the \_Asset\_Cache folder as part of the \_Cache folder will be created if the folder does not exists. So a later Byte.copy orsome other commands are able to access the file from the App’s Asset Cache Directory. Larger files should be deleted after use.

The first parameter is a single character that sets the I/O mode for this file:

\*\*\*\*\*

Файл, указанный в строковом выражении <path\_sexp>, открыт. Если путь представляет собой URL-адрес, начинающийся с «http…», то открывается интернет-файл. В противном случае строка <path\_sexp> добавляется к пути по умолчанию «<pref base drive> / rfo-basic / data /».

Если URL начинается с «asset: //» + myFileNamePath $, папка \_Asset\_Cache как часть папки \_Cache будет создана, если папка не существует. Таким образом, более поздняя версия Byte.copy или другие команды могут получить доступ к файлу из каталога кэша активов приложения. Большие файлы должны быть удалены после использования.

Первый параметр - это один символ, который устанавливает режим ввода-вывода для этого файла:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Mode** | **Notes** |
| r | read | File exists: Reads from the start of the file. File does not exist: Error (see below). |
| w | write | File exists: Writes from the start of the file. Writes over any existing data.  File does not exist: Creates a new file. Writes from the start of the file. |
| a | append | File exists: Writing starts after the last line in the file.  File does not exist: Creates a new file. Writes from the start of the file. |

Afiletablenumberisplacedintothenumericvariable<file\_table\_nvar>.Thisvalueisfor use in subsequent **Byte.read.\***, **Byte.write**.\*, **Byte.eof**, **Byte.position.\***, **Byte.truncate**, **Byte.copy**, or **Byte.close**commands.

If a file being opened for read does not exist then the <file\_table\_nvar> will be set to -1. The BASIC! programcan check for this and either create the file or report the error to the user. Information about the error isavailable from the **GETERROR$()** function.

\*\*\*\*\*

Afiletablenumber помещается в числовую переменную <file\_table\_nvar>. Это значение используется для последующих команд Byte.read. \*, Byte.write. \*, Byte.eof, Byte.position. \*, Byte.truncate, Byte.copy или Byte.close.

Если файл, открытый для чтения, не существует, тогда для <file\_table\_nvar> будет установлено значение -1. Основа! Программа может проверить это и либо создать файл, либо сообщить об ошибке пользователю. Информация об ошибке доступна из функции GETERROR $ ().

GitHub#249

### Byte.copy <file\_table\_nexp>, <output\_file\_sexp>{{,<append\_nexp>},

**<break\_nexp>}**

Copies the previously open input file represented by <file\_table\_nexp> to the file whose path is specified by <output\_file\_sexp>. The default path is "<pref base

drive>/rfo-basic/data/".

If <file\_table\_nexp> = -1 then a run-time error will be thrown.

All bytes from the current position of the input file to its end are copied to the to the output file. Both files are then closed.

If you have read from the input file, and you want to copy the whole file, you must reset the file position to 0 with **Byte.position.set**. However, if you have changed the file mark with **Byte.position.mark**, or if you reading a non-local (internet) file, you can’t reset the file position to 0. Instead, you must close and reopen the file.

You should use **Byte.copy** if you are using Byte I/O for the sole purpose of copying. It is thousands (literally) of times faster than using **Byte.read**/**Byte.write**.

If you want to append the input file content to the output file, the optional <append\_nexp> parameter has to be > 0. Default is 0.

If the output file does not exists in both cases, a new file will be created before writing. If <break\_nexp> is greater than zero, Byte.copy can be interrupted by an interrupt.

Копирует ранее открытый входной файл, представленный <file\_table\_nexp>, в файл, путь которого указан в <output\_file\_sexp>. Путь по умолчанию "<pref base

диск> / РФО основной / данные /».

Если <file\_table\_nexp> = -1, будет выдана ошибка времени выполнения.

Все байты от текущей позиции входного файла до его конца копируются в выходной файл. Оба файла затем закрываются.

Если вы прочитали из входного файла и хотите скопировать весь файл, вы должны сбросить позицию файла в 0 с помощью Byte.position.set. Однако, если вы изменили метку файла с помощью Byte.position.mark или читаете нелокальный (интернет) файл, вы не можете сбросить позицию файла до 0. Вместо этого вы должны закрыть и снова открыть файл.

Вам следует использовать Byte.copy, если вы используете байтовый ввод-вывод с единственной целью копирования. Это в тысячи раз (буквально) быстрее, чем при использовании Byte.read/Byte.write.

Если вы хотите добавить содержимое входного файла в выходной файл, необязательный параметр <append\_nexp> должен быть> 0. По умолчанию 0.

Если выходной файл не существует в обоих случаях, новый файл будет создан перед записью. Если <break\_nexp> больше нуля, Byte.copy может быть прерван прерыванием.

### Byte.read.buffer <file\_table\_nexp>, <count\_nexp>, <buffer\_svar>{,

**<charset\_sexp>}**

Reads the specified number of bytes (<count\_nexp>) into the buffer string variable (<buffer\_svar>) from the file. The string length (len(<buffer\_svar>)) will be the number of bytes actually read. If the end of file is reached, the string length may be less than the requested count.

A buffer string is a special use of the BASIC! string. Each character of a string is 16 bits. When used as a buffer, one byte of data is written into the lower 8 bits of each 16-bit character. The upper 8 bits are 0. Extract the binary data from the string, one byte at a time, with the **ASCII()** or **UCODE()** functions.

The format of the buffer string read by this command is compatible with the **DECODE$()** function. If you know that part of your data contains an encoded string, you can extract the substring (using a function like **MID$()**), then pass the substring to **DECODE$()** to convert it to a BASIC! String.

Using the optional <charset\_sexp> you can choose between following character sets: "\_US-ASCII" and "\_ISO-8859-1"(default).

In most cases the command GrabFile is the better choice.

\*\*\*\*\*

Считывает указанное количество байтов (<count\_nexp>) в строковую переменную буфера (<buffer\_svar>) из файла. Длина строки (len (<buffer\_svar>)) будет количеством фактически прочитанных байтов. Если достигнут конец файла, длина строки может быть меньше запрошенного числа.

Строка буфера - это специальное использование BASIC! строка. Каждый символ строки составляет 16 бит. При использовании в качестве буфера один байт данных записывается в младшие 8 бит каждого 16-битного символа. Старшие 8 бит равны 0. Извлекайте двоичные данные из строки, по одному байту за раз, с помощью функций ASCII () или UCODE ().

Формат строки буфера, читаемой этой командой, совместим с функцией DECODE $ (). Если вы знаете, что часть ваших данных содержит закодированную строку, вы можете извлечь подстроку (используя такую ​​функцию, как MID $ ()), а затем передать подстроку в DECODE $ (), чтобы преобразовать ее в BASIC! Строка.

Используя опциональное <charset\_sexp>, вы можете выбирать между следующими наборами символов: "\_US-ASCII" и "\_ISO-8859-1" (по умолчанию).

В большинстве случаев команда GrabFile является лучшим выбором.

### Byte.write.buffer <file\_table\_nexp>, <buffer\_sexp>{, <charset\_sexp>}

Writes the entire contents of the string expression to the file. The string is assumed to be a buffer string holding binary data, as described in **Byte.read.buffer**. The writer discards the upper 8 bits of each 16-bit character, writing one byte to the file for each character in the string.

The **Byte.read.buffer** command and the **ENCODE$()** function always create these "buffer strings". You can construct one by using, for example, the **CHR$()** function with values less than 256.

If you use only ASCII characters in a string, you can use this function to write the string to a file. The output is the same as if you had written it with **Text.writeln**, except that it will have no added newline.

Using the optional <charset\_sexp> you can choose between following character sets: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" and "\_ISO-8859-

1"(default).

If you want to save international text use"\_UTF-8".

This command expects only character codes up to 255 for "\_US-ASCII" or for binary data "\_ISO-8859-1" and writes only one byte.

If you want to convert an integer to a byte array use bit shifting like this:

\*\*\*\*\*\*

Записывает все содержимое строкового выражения в файл. Предполагается, что строка является строкой буфера, содержащей двоичные данные, как описано в Byte.read.buffer. Модуль записи отбрасывает старшие 8 бит каждого 16-битного символа, записывая один байт в файл для каждого символа в строке.

Команда Byte.read.buffer и функция ENCODE $ () всегда создают эти «строки буфера». Вы можете создать его, используя, например, функцию CHR $ () со значениями меньше 256.

Если вы используете в строке только символы ASCII, вы можете использовать эту функцию для записи строки в файл. Вывод такой же, как если бы вы написали его с помощью Text.writeln, за исключением того, что в нем не будет добавлен символ новой строки.

Используя опциональное <charset\_sexp>, вы можете выбирать между следующими наборами символов: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" и "\_ISO-8859-

1" (по умолчанию).

Если вы хотите сохранить международный текст, используйте «\_UTF-8».

Эта команда ожидает только код символов до 255 для «\_US-ASCII» или для двоичных данных «\_ISO-8859-1» и записывает только один байт.

Если вы хотите преобразовать целое число в байтовый массив, используйте сдвиг битов следующим образом:

FN.DEF integer2Str$( val) b0$=CHR$(band(val,255)) b1$=CHR$(band(shift(val,8),255)) b2$=CHR$(band(shift(val,16),255)) b3$=CHR$(band(shift(val,24),255)) FN.RTN b0$+b1$+b2$+b3$

FN.END

Using "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" with Byte.write.buffer write byte arrays with up to 4 bytes for each character.

If you get in trouble with binary data using Byte.write.buffer try Byte.write.byte with your string. It is slow but it interprets character codes over 255 differently.

\*\*\*\*\*

массивы до 4 байтов для каждого символа.

Если у вас возникли проблемы с двоичными данными с помощью Byte.write.buffer, попробуйте Byte.write.byte со своей строкой. Это медленно, но по-разному интерпретирует коды символов свыше 255.22

See also Byte.write.byte, GrabFile, GrabURL

### GrabFile <result\_svar>, <path\_sexp>{, <unicode\_flag\_lexp>|<charset\_sexp>}

Copies the entire contents of the file at <path\_sexp> to the string variable <result\_svar>. By default, **GrabFile** assumes that the file contains binary bytes or ASCII characters. If the optional <unicode\_flag\_lexp> evaluates to true (a non-zero numeric value), **GrabFile** can read Unicode text.

If you change <unicode\_flag\_lexp> to <charset\_sexp> you can choose between following character sets: "\_US-ASCII"(1), "\_UTF-8"(1), "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE"and "\_ISO-8859-1"(0).

If the file does not exist or cannot be opened, the <result\_svar> is set to the empty string, "", and you can use the **GETERROR$()** function to get more information. If the file is empty, the <result\_svar> is an empty string, "", but **GETERROR$()** returns "No error".

For text files, either ASCII or Unicode, the **Split** command can be used to split the

<result\_svar> into an array of lines. **GrabFile** can also be used grab the contents of atext file for direct use with**Text.input**:

\*\*\*\*\*

Копирует все содержимое файла в <path\_sexp> в строковую переменную <result\_svar>. По умолчанию GrabFile предполагает, что файл содержит двоичные байты или символы ASCII. Если необязательный <unicode\_flag\_lexp> имеет значение true (ненулевое числовое значение), GrabFile может читать текст Unicode.

Если вы измените <unicode\_flag\_lexp> на <charset\_sexp>, вы можете выбрать один из следующих наборов символов: "\_US-ASCII" (1), "\_UTF-8" (1), "\_UTF-16", "\_UTF-16BE", "\_UTF -16LE "и" \_ISO-8859-1 "(0).

Если файл не существует или не может быть открыт, для <result\_svar> устанавливается пустая строка "", и вы можете использовать функцию GETERROR $ () для получения дополнительной информации. Если файл пуст, <result\_svar> является пустой строкой "", но GETERROR $ () возвращает "No error".

Для текстовых файлов, ASCII или Unicode, команда Split может использоваться для разделения

<result\_svar> в массив строк. GrabFile также может быть использован для захвата содержимого файла atext для непосредственного использования с Text.input:

GRABFILE text$, "MyJournal.txt" TEXT.INPUT EditedText$,text$

### GrabURI <result\_svar>, <uri\_sexp>{, <unicode\_flag\_lexp>|<charset\_sexp>}

Copies the entire contents of the file at <path\_sexp> to the string variable <result\_svar>. By default, **GrabFile** assumes that the file contains binary bytes or ASCII characters. If the optional <unicode\_flag\_lexp> evaluates to true (a non-zero numeric value), **GrabFile** can read Unicode text.

If you change <unicode\_flag\_lexp> to <charset\_sexp> you can choose between following character sets: "\_US-ASCII"(1), "\_UTF-8"(1), "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE"and "\_ISO-8859-1"(0).

If the file does not exist or cannot be opened, the <result\_svar> is set to the empty string, "", and you can use the **GETERROR$()** function to get more information. If the file is empty, the <result\_svar> is an empty string, "", but **GETERROR$()** returns "No error".

For text files, either ASCII or Unicode, the **Split** command can be used to split the

<result\_svar> into an array of lines. **GrabFile** can also be used grab the contents of atext file for direct use with**Text.input**:

\*\*\*\*\*

Копирует все содержимое файла в <path\_sexp> в строковую переменную <result\_svar>. По умолчанию GrabFile предполагает, что файл содержит двоичные байты или символы ASCII. Если необязательный <unicode\_flag\_lexp> имеет значение true (ненулевое числовое значение), GrabFile может читать текст Unicode.

Если вы измените <unicode\_flag\_lexp> на <charset\_sexp>, вы можете выбрать один из следующих наборов символов: "\_US-ASCII" (1), "\_UTF-8" (1), "\_UTF-16", "\_UTF-16BE", "\_UTF -16LE "и" \_ISO-8859-1 "(0).

Если файл не существует или не может быть открыт, для <result\_svar> устанавливается пустая строка "", и вы можете использовать функцию GETERROR $ () для получения дополнительной информации. Если файл пуст, <result\_svar> является пустой строкой "", но GETERROR $ () возвращает "No error".

Для текстовых файлов, ASCII или Unicode, команда Split может использоваться для разделения

<result\_svar> в массив строк. GrabFile также может быть использован для захвата содержимого файла atext для непосредственного использования с Text.input:

GRABFILE text$, "MyJournal.txt" TEXT.INPUT EditedText$,text$

### GrabURL <result\_svar>, <url\_sexp>{{, <timeout\_nexp>},<unicode\_flag\_lexp>|

**<charset\_sexp>}**

Copies the entire source text of the URL <url\_sexp> to the string variable <result\_svar>. The URL may specify an Internet resource or a local file. Cached files will be ignored. If the URL does not exist or the data cannot be read, the <result\_svar> is set to the empty string, "", and you can use the **GETERROR$()** function to get more information.

If the optional <timeout\_nexp> parameter is non-zero, it specifies a time-out in milliseconds. Thus 0 specified no time-out. This is meaningful only if the URL names a resource on a remote host. If the time-out time elapses and host does not connect or does not return any data, **GETERROR$** reports a socket timeout. By default, **GrabURL** assumes that the file contains Unicode text. If the optional <unicode\_flag\_lexp> evaluates to false (a zero numeric value), **GrabURL** can read binary bytes or ASCII characters.

Attention: The **default** <unicode\_flag\_lexp> setting is the opposite of **GrabFile**!

If you change <unicode\_flag\_lexp> to <charset\_sexp> you can choose between following character sets: "\_US-ASCII"(1), "\_UTF-8"(1), "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE"and "\_ISO-8859-1"(0).

If the named resource is empty, the <result\_svar> is empty, "", and **GETERROR$()** returns "No error".

The "Split" command can be used to split the <result\_svar> into an array of lines forASCII or Unicode text.

\*\*\*\*\*

Копирует весь исходный текст URL <url\_sexp> в строковую переменную <result\_svar>. URL может указывать интернет-ресурс или локальный файл. Кэшированные файлы будут игнорироваться. Если URL-адрес не существует или данные не могут быть прочитаны, для <result\_svar> устанавливается пустая строка "", и вы можете использовать функцию GETERROR $ () для получения дополнительной информации.

Если необязательный параметр <timeout\_nexp> не равен нулю, он указывает время ожидания в миллисекундах. Таким образом, 0 указывает отсутствие времени ожидания. Это имеет смысл, только если URL-адрес именует ресурс на удаленном хосте. Если время ожидания истекло, и хост не подключается или не возвращает никаких данных, GETERROR $ сообщает о времени ожидания сокета. По умолчанию GrabURL предполагает, что файл содержит текст Unicode. Если необязательный <unicode\_flag\_lexp> имеет значение false (нулевое числовое значение), GrabURL может читать двоичные байты или символы ASCII.

Внимание: настройка по умолчанию <unicode\_flag\_lexp> противоположна GrabFile!

Если вы измените <unicode\_flag\_lexp> на <charset\_sexp>, вы можете выбрать один из следующих наборов символов: "\_US-ASCII" (1), "\_UTF-8" (1), "\_UTF-16", "\_UTF-16BE", "\_UTF -16LE "и" \_ISO-8859-1 "(0).

Если указанный ресурс пуст, <result\_svar> пуст, "", и GETERROR $ () возвращает "Нет ошибки".

Команда «Разделить» может быть использована для разделения <result\_svar> на массив строк для текста ASCII или Unicode.

### Ftp.open <url\_sexp>, <port\_nexp>, <user\_sexp>, <pw\_sexp>

Connects to the specified url and port. Logs onto the server using the specified user name and password. For example:

**ftp.open "ftp.laughton.com", 21, "basic", "basic"**

You can also write:

ftp.open "ftp://ftp.laughton.com", 21, "basic", "basic"

Or:

ftp.open "ftps://myServer.net", 2221, "demo", "demo" ftp.open "ftpes://192.168.1.2", 2221, "demo", "demo"

In the last two cases you open a ftp over Transport Layer Security (TLS) connection, so called FTPS. This implementation works in explicit mode (also known as FTPES).

Working as a client with implicit mode is not supported by Android.

For a FTP/FTPES/FTPS server on Android is the **Wifi FTP Server** recommended. (id=com.medhaapps.wififtpserver or id=com.medhaapps.wififtpserver.pro)

If you want start and stop a FTP server, you can use the app **FTP-Server**, also. (id=com.theolivetree.ftpserver or id=com.theolivetree.ftpserverpro)

Start and Stop by the following intents: com.theolivetree.ftpserver.StartFtpServer or com.theolivetree.ftpserver.StartFtpServerPro com.theolivetree.ftpserver.StopFtpServer or com.theolivetree.ftpserver.StopFtpServerPro

If you want to use SFTP use the App **andFTP** (id=lysesoft.andftp and id=lysesoft.andftppro[Key only]). This App can be controlled by intents. More informations at [http://www.lysesoft.com/products/andftp/index.html.](http://www.lysesoft.com/products/andftp/index.html)For a SFTP server on Android is the **Ssh Server** recommended. (id=com.theolivetree.sshserver or id=com.theolivetree.sshserverpro) Start and Stop by the following intents:

\*\*\*\*\*

Подключается к указанному URL и порту. Вход на сервер с использованием указанного имени пользователя и пароля. Например:

ftp.open "ftp.laughton.com", 21, "базовый", "базовый"

Вы также можете написать:

ftp.open "ftp://ftp.laughton.com", 21, "базовый", "базовый"

Или же:

ftp.open "ftps: //myServer.net", 2221, "demo", "demo" ftp.open "ftpes: //192.168.1.2", 2221, "demo", "demo"

В последних двух случаях вы открываете FTP-соединение через TLS, так называемое FTPS. Эта реализация работает в явном режиме (также известный как FTPES).

Работа в качестве клиента в неявном режиме не поддерживается Android.

Для сервера FTP / FTPES / FTPS на Android рекомендуется использовать FTP-сервер Wifi. (id = com.medhaapps.wififtpserver или id = com.medhaapps.wififtpserver.pro)

Если вы хотите запустить и остановить FTP-сервер, вы также можете использовать приложение FTP-сервер. (id = com.theolivetree.ftpserver или id = com.theolivetree.ftpserverpro)

Запускать и останавливать с помощью следующих намерений: com.theolivetree.ftpserver.StartFtpServer или com.theolivetree.ftpserver.StartFtpServerPro com.theolivetree.ftpserver.StopFtpServer или com.theolivetree.ftpserver.StopFtpSerPro

Если вы хотите использовать SFTP, используйте приложение и FTP (id = lysesoft.andftp и id = lysesoft.andftppro [только ключ]). Это приложение может контролироваться намерениями. Дополнительная информация на http://www.lysesoft.com/products/andftp/index.html. Для SFTP-серверов на Android рекомендуется Ssh Server. (id = com.theolivetree.sshserver или id = com.theolivetree.sshserverpro) Запуск и остановка следующими намерениями:

com.theolivetree.sshserver.StartSshServer or com.theolivetree.sshserverpro.StartSshServer com.theolivetree.sshserver.StopSshServer or com.theolivetree.sshserverpro.StopSshServer

### Ftp.dir <list\_nvar> {{,<dirmark\_sexp>},<timeStamp\_nexp>}

Creates a list of the names of the files and directories in the current working directory and places it in a BASIC! List data structure. A pointer to the new List is returned in the variable

<list\_nvar>.

A directory is identified by a marker appended to its name. The default marker is the string "(d)". You can change the marker with the optional directory mark parameter

<dirmark\_sexp>. If you do not want directories to be marked, set <dirmark\_sexp> to an empty string, "".

\*\*\*\*\*

Создает список имен файлов и каталогов в текущем рабочем каталоге и помещает его в бейсик! Структура данных списка. Указатель на новый список возвращается в переменной

<List\_nvar>.

Каталог идентифицируется маркером, добавленным к его имени. Маркер по умолчанию - это строка "(d)". Вы можете изменить маркер с помощью дополнительного параметра метки каталога

<Dirmark\_sexp>. Если вы не хотите, чтобы каталоги были помечены, установите <dirmark\_sexp> в пустую строку "".

Options of <timeStamp\_nexp>:

* + 0 no time stamp(default)
  + 1 with time stamp as time in milliseconds + ":" + file name, butunsorted
  + 2 with time stamp as time in milliseconds + ":" + file name, sorted in ascendingorder
  + 3 with time stamp as time in milliseconds + ":" + file name, sorted in descendingorder

The following code can be used to print the file names in that list:

**ftp.dir file\_list list.size file\_list,size**

**for i = 1 to size**

**list.get file\_list,i,name$ print name$**

**nextI**

GitHub#244

### Documents

**A**ndroid supports **doc**uments. An Android document can be a directory, a file, a database or something else. Each document contains content, that is provided by a content provider. A path pointing at a content begins with "content://".

A content provider gives access to the own or data to other applications also. It can as an example be a file system or a cloud storage like Google Drive.

Unfortunately OliBasic does not support an own content provider today, but you can provide data in a file saved into an **external** directory or a supported cloud storage. **External** means public file access. SD cards and USB sticks are **removable** data carrier.

Each valid document path gives you access to the document name and the default size, but not in all cases to the data.

Handling documents is a little different in opposite to files.

For better understanding we will look at a cloud storage first. A cloud storage is connected by a network.

If you use a mobile network, the link can be broken. To prevent data loss it needs an exact status each time.

If you want a directory document, you have to use Adoc.save with a document type named "\_Dir".

A new file needs also Adoc.save.

In this case you can preset the document type and its name. Creating a new directory in its workflow is possible too.

Now the document has to be filled by Adoc.write. This command uses strings.

If you want to copy or save binary data use "\_ISO-8859-1" as character set.

It is time to try to read the stored content. We start Adoc.open to choose a document.

If the network connection is not broken, we get a document path beginning with "content://".

Adoc.read gives the possibility to get the content as a string.

If you want to copy a document read and put the content into a string with the"\_ISO-8859- 1" character set and write the content into a new document crated by theAdoc.save dialog with the same character set. Try to enumerate the existing of the createddocument.

If a document has to be renamed, copy the document and delete the first documentby Adoc.delete.

A content path is different to a file path.

Sometimes it looks like a readable file file path, but in cases of cloud storage, last documents, downloads etc. it is coded or/and encrypted.

File.absolute try to convert the document path into an absolute file path.

If it fails using a valid document path, returns only the document name without any slash ("/").

Selecting a document in the Downloads document directory returns a file path, if the document name equals with a first level file name.

If it returns a valid absolute file path, you can operate with normal file dependent commands.

Otherwise copy documents into files to handle with.

What is the outlook?

Google will take more care about user's security.

The user will be more involved to confirm actions with possible data risks. That is not what developers normally want.

They like paths, which can be extend by easy readable directory and file names.

The consequence is storing selected document paths permanently in a private file folder created by File.root path$, "\_Internal".

\*\*\*\*\*

Android поддерживает документы. Документ Android может быть каталогом, файлом, базой данных или чем-то еще. Каждый документ содержит контент, предоставляемый поставщиком контента. Путь, указывающий на контент, начинается с «content: //».

Контент-провайдер предоставляет доступ к своим или данным другим приложениям. Например, это может быть файловая система или облачное хранилище, такое как Google Drive.

К сожалению, сегодня OliBasic не поддерживает собственного поставщика контента, но вы можете предоставить данные в файле, сохраненном во внешнем каталоге или в поддерживаемом облачном хранилище. Внешний означает открытый доступ к файлам. SD-карты и флешки являются съемными носителями информации.

Каждый допустимый путь к документу дает вам доступ к имени документа и размеру по умолчанию, но не во всех случаях к данным.

Обработка документов немного отличается от файлов.

Для лучшего понимания мы сначала рассмотрим облачное хранилище. Облачное хранилище связано сетью.

Если вы используете мобильную сеть, ссылка может быть разорвана. Чтобы предотвратить потерю данных, ему нужен точный статус каждый раз.

Если вам нужен документ каталога, вы должны использовать Adoc.save с типом документа «\_Dir».

Новый файл требует также Adoc.save.

В этом случае вы можете предварительно установить тип документа и его имя. Создание нового каталога в его рабочем процессе также возможно.

Теперь документ должен быть заполнен Adoc.write. Эта команда использует строки.

Если вы хотите скопировать или сохранить двоичные данные, используйте «\_ISO-8859-1» в качестве набора символов.

Настало время попробовать прочитать сохраненный контент. Мы запускаем Adoc.open, чтобы выбрать документ.

Если сетевое соединение не разорвано, мы получаем путь к документу, начинающийся с «content: //».

Adoc.read дает возможность получать контент в виде строки.

Если вы хотите скопировать документ, прочитайте и поместите содержимое в строку с набором символов «\_ISO-8859-1» и запишите содержимое в новый документ, созданный с помощью диалога Adoc.save с тем же набором символов. Попробуйте перечислить существующий из созданного документа.

Если документ необходимо переименовать, скопируйте документ и удалите первый документ с помощью Adoc.delete.

Путь к содержимому отличается от пути к файлу.

Иногда это выглядит как читаемый путь к файлу файла, но в случае облачного хранилища, последних документов, загрузок и т. Д. Он кодируется или / и шифруется.

File.absolute попытайтесь преобразовать путь к документу в абсолютный путь к файлу.

Если не удается использовать допустимый путь к документу, возвращается только имя документа без косой черты ("/").

Выбор документа в каталоге документов Downloads возвращает путь к файлу, если имя документа совпадает с именем файла первого уровня.

Если он возвращает правильный абсолютный путь к файлу, вы можете работать с обычными зависимыми от файла командами.

В противном случае скопируйте документы в файлы для обработки.

Каковы перспективы?

Google будет больше заботиться о безопасности пользователя.

Пользователь будет больше вовлечен, чтобы подтвердить действия с возможными рисками данных. Это не то, что обычно хотят разработчики.

Они любят пути, которые могут быть расширены легко читаемыми именами каталогов и файлов.

Следствием этого является постоянное хранение выбранных путей к документам в личной папке с файлами, созданной с помощью пути File.root $, "\_Internal".

Known issues:

<startPath\_sexp> does not work properly in all cases.

### Adoc.path <documentPath\_svar>, <filePath\_sexp>

Returns a document path beginning with “content://” by the given file path. If an error occurs or it is not possible an empty string ("") is returned.

\*\*\*\*\*

Возвращает путь к документу, начинающийся с «content: //» по указанному пути к файлу. Если возникает ошибка или это невозможно, возвращается пустая строка ("").

### Adoc.open <documentPath\_svar>{{, <startPath\_sexp>}, <mimeType\_sexp>}

Opens a system dialog for opening a document and returns the selected path by

<documentPath\_svar>. If <documentPath\_svar> returns an empty string (""), the operation was not successful.

An existing start path will be defined by <startPath\_sexp>. An empty string let the document browser start at the default data path. If this attribute is not used the start begins at the last visited location.

Android stores each document into a database and is also interested in the document type. To shrink the number of select-able documents use <mimeType\_sexp> to specify the wanted types.

It grants also permissions to the selected document until the system will be restarted.

##### So restart your system and test your app again before publishing.

The default MIME type is "\*/\*".

Common types are "image/png", "image/jpg", "text/plain" etc.. If a new document directory is needed, use "\_Dir".

Min. KitKat 4.4 (API 19) is needed.

\*\*\*\*\*

Открывает системный диалог для открытия документа и возвращает выбранный путь

<DocumentPath\_svar>. Если <documentPath\_svar> возвращает пустую строку (""), операция не была успешной.

Существующий начальный путь будет определен <startPath\_sexp>. Пустая строка позволяет браузеру документов начинаться с пути данных по умолчанию. Если этот атрибут не используется, начало начинается с последнего посещенного местоположения.

Android хранит каждый документ в базе данных, а также интересуется типом документа. Чтобы уменьшить количество документов, доступных для выбора, используйте <mimeType\_sexp>, чтобы указать нужные типы.

Он также предоставляет разрешения для выбранного документа, пока система не будет перезапущена.

Поэтому перезапустите систему и протестируйте приложение перед публикацией.

Тип MIME по умолчанию - "\* / \*".

Распространенными типами являются «image / png», «image / jpg», «text / plain» и т. Д. Если требуется новый каталог документов, используйте «\_Dir».

Минимум Требуется KitKat 4.4 (API 19).

Example:

! As a file browser

startPath$ = "" % Default data path ADOC.OPEN documentName$, startPath$ FILE.ABSOLUTE fileName$, documentName$ GRABFILE result$, fileName$

PRINT result$

or

! As a document browser

startPath$ = "" % Default data path ADOC.OPEN documentName$, startPath$ ADOC.READ result$, documentName$ PRINT result$

or

! As a document browser perhaps on Google Drive FILE.ROOT path$, "\_Internal"

absoluteDocsPath$ = "file://" + path$ + "docs.bn" FILE.EXISTS ok, absoluteDocsPath$

IF ok

BUNDLE.LOAD mDocs, absoluteDocsPath$

! bundle.clear mDocs % If something went wrong at the first tries

! In this case do not forget to delete all created Documents also

! Document Directories by the Google Drive App before. BUNDLE.CONTAIN mDocs, "myDocDirectoryOnGoogleDrive", ex

IF ex THEN BUNDLE.GET mDocs, "myDocDirectoryOnGoogleDrive", startPath$ ENDIF

IF !ok | !ex

rootDir$ = "myDocDirectory"

ADOC.SAVE startPath$, rootDir$, "", "\_Dir" PRINT "startPath$", startPath$

BUNDLE.PUT mDocs, "myDocDirectoryOnGoogleDrive", startPath$ BUNDLE.SAVE mDocs, absoluteDocsPath$

ENDIF

newDocName$ = "myFirstDocument.txt" BUNDLE.CONTAIN mDocs, "./" + newDocName$, ex IF ex

message$ = "Take care, because\n you are able to\n " ~

+ "create a second one\n with the same name"

DIALOG.MESSAGE "Document Exists", message$, sel , "SKIP", "OK" ENDIF

IF sel <> 1

ADOC.SAVE documentPath$, newDocName$, startPath$, "text/plain" BUNDLE.PUT mDocs, "./" + newDocName$, documentPath$ BUNDLE.SAVE mDocs, absoluteDocsPath$

ENDIF

ADOC.EXISTS ok, documentPath$

IF ok THEN ADOC.WRITE ws, documentPath$, "My First Text On Google Drive" ADOC.OPEN documentPath$ ,startPath$

ADOC.READ ok, result$, documentPath$ %, "\_UTF-8" PRINT result$, ok

END

! Before the third run change your root Document Directory name

! by the Google Drive App.

! You should see, that your Document is still selectable.

! With Adoc.name you can update your Docs bundle.

### Adoc.save <documentPath\_svar>, <documentName\_svar>{{,

**<startPath\_sexp>}, <mimeType\_sexp>}**

Opens a system dialog for saving a document and returns the created or selected path by

<documentPath\_svar>. If <documentPath\_svar> returns an empty string (""), the operation was not successful.

<documentName\_svar> presents the new document name.

An existing start path will be defined by <startPath\_sexp>. An empty string let the document browser start at the default data path. If this attribute is not used the start begins at the last visited location.

Android tries to put each document into a database and is also interested in the document type. To shrink the number of select-able documents use <mimeType\_sexp> to specifythe wantedtypes.

The default MIME type is "\*/\*".

Common types are "image/png", "image/jpg",, "text/plain" etc.. If a new document directory is needed, use "\_Dir".

Min. KitKat 4.4 (API 19) is needed.

\*\*\*\*\*

Открывает системный диалог для сохранения документа и возвращает созданный или выбранный путь

<DocumentPath\_svar>. Если <documentPath\_svar> возвращает пустую строку (""), операция не была успешной.

<documentName\_svar> представляет имя нового документа.

Существующий начальный путь будет определен <startPath\_sexp>. Пустая строка позволяет браузеру документов начинаться с пути данных по умолчанию. Если этот атрибут не используется, начало начинается с последнего посещенного местоположения.

Android пытается поместить каждый документ в базу данных, а также интересуется типом документа. Чтобы уменьшить количество документов, доступных для выбора, используйте <mimeType\_sexp>, чтобы указать нужные типы.

Тип MIME по умолчанию - "\* / \*".

Распространенными типами являются «image / png», «image / jpg», «text / plain» и т. Д. Если требуется новый каталог документов, используйте «\_Dir».

Минимум Требуется KitKat 4.4 (API 19).

### Adoc.get <documentPath\_svar>{{, <startPath\_sexp>}, <mimeType\_sexp>}

Opens a system dialog for opening a document and returns the selected path by

<documentPath\_svar>. If <documentPath\_svar> returns an empty string (""), the operation was not successful.

An existing start path will be defined by <startPath\_sexp>. An empty string let the document browser start at the default data path. If this attribute is not used the start begins at the last visited location.

To shrink the number of select-able documents use <mimeType\_sexp> to specify the wanted types.

Unlike Adoc.open, **only** document providers specified for **reading** can be selected. The possibility of choices is usually larger.

The returned URI will only be safe for read access. Se alsoAdoc.open

\*\*\*\*\*

Открывает системный диалог для открытия документа и возвращает выбранный путь

<DocumentPath\_svar>. Если <documentPath\_svar> возвращает пустую строку (""), операция не была успешной.

Существующий начальный путь будет определен <startPath\_sexp>. Пустая строка позволяет браузеру документов начинаться с пути данных по умолчанию. Если этот атрибут не используется, начало начинается с последнего посещенного местоположения.

Чтобы уменьшить количество документов, доступных для выбора, используйте <mimeType\_sexp>, чтобы указать нужные типы.

В отличие от Adoc.open, могут быть выбраны только провайдеры документов, указанные для чтения. Возможность выбора обычно больше.

Возвращенный URI будет безопасен только для доступа на чтение. Видеть

### Adoc.name <documentName\_svar>, <documentPath\_sexp>

Returns the human-friendly name of the document by the given document path. If this is not provided then the name should default to the last segment of the documents's URI. If an error occurs an empty string "" is returned.

\*\*\*\*\*

Возвращает понятное для человека имя документа по указанному пути документа. Если это не предусмотрено, то имя должно по умолчанию соответствовать последнему сегменту URI документов. Если возникает ошибка, возвращается пустая строка "".

### Adoc.size <documentSize\_nvar>, <documentPath\_sexp>

Returns the number of bytes in the document identified by the openable document path. Gives back -1 if unknown.

\*\*\*\*\*

Возвращает количество байтов в документе, идентифицированном открываемым путем документа. Возвращает -1, если неизвестно.

### Adoc.mimetype <success\_nvar>, <mimeType\_svar>, <documentPath\_sexp>

Returns 1 if the MIME type of the document specified by the document path was returned successfully. Otherwise 0 is returned. The value given by <documentMimeType\_nvar> returns the MIME type of this document.

\*\*\*\*\*

Возвращает 1, если MIME-тип документа, указанного в пути к документу, был возвращен успешно. В противном случае возвращается 0. Значение, данное <documentMimeType\_nvar>, возвращает тип MIME этого документа.

### Adoc.lastmodified <success\_nvar>, <documentLastModified\_nvar>,

**<documentPath\_sexp>**

Returns 1 if the last modification of the document specified by the document pathwas returned successfully. Otherwise 0 is returned. The value givenby

<documentLastModified\_nvar> returns the time when this document was last modified, measured in milliseconds since January 1st, 1970, midnightGMT.

\*\*\*\*\*

Возвращает 1, если последняя модификация документа, указанного в пути к документу, была возвращена успешно. В противном случае возвращается 0. Значение, данное

<documentLastModified\_nvar> возвращает время последнего изменения этого документа, измеренное в миллисекундах с 1 января 1970 года, midnightGMT.

Example:

FILE.ROOT pp$ ,"\_SourceSamples"

ADOC.PATH dp$,"file://"+ pp$ + "/" + "f01\_commands.bas" ADOC.LASTMODIFIED lm, dp$

? USING$("", "%TF %tT" ,lm, lm) % Returns "2019-06-30 21:01:01"

### Adoc.delete <success\_nvar>, <documentPath\_sexp>

Returns 1 if the document specified by the document path was deleted successfully. Otherwise 0 is returned.

Min. KitKat 4.4 (API 19) is needed.

\*\*\*\*\*

Возвращает 1, если документ, указанный в пути к документу, был успешно удален. В противном случае возвращается 0.

Минимум Требуется KitKat 4.4 (API 19).

### Adoc.rename <success\_nvar>, <documentPath\_sexp>, <newName\_sexp>

Returns 1 if the document specified by the document path was renamed specified by

<newName\_sexp> successfully. Otherwise 0 is returned. Min. Lollipop 5.0 (API 21) is needed.

\*\*\*\*\*

Возвращает 1, если документ, указанный в пути к документу, был переименован, как указано

<newName\_sexp> успешно. В противном случае возвращается 0. Минимум Требуется Lollipop 5.0 (API 21).

### Adoc.write <success\_nvar>, <documentPath\_sexp>, <newContent\_sexp>{,

**<charSet\_sexp>}**

Returns 1 if the document specified by the document path was completely filled with String content specified by <newContent\_sexp> successfully. Otherwise 0 is returned.

Note: The specified document has to be created before!

For binary data specify the <charSet\_sexp> with the "\_ISO-8859-1" character set instead of the default "\_UTF-8".

A required permission is granted until it is revoked or the app deinstalled.

You can choose between following character sets: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" and "\_ISO-8859-1".

Min. KitKat 4.4 (API 19) is needed.

\*\*\*\*\*

Возвращает 1, если документ, указанный в пути к документу, был полностью заполнен содержимым строки, заданным параметром <newContent\_sexp>. В противном случае возвращается 0.

Примечание: указанный документ должен быть создан раньше!

Для двоичных данных укажите <charSet\_sexp> с набором символов "\_ISO-8859-1" вместо значения по умолчанию "\_UTF-8".

Требуемое разрешение предоставляется до тех пор, пока оно не будет отменено или приложение не будет удалено.

Вы можете выбрать один из следующих наборов символов: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" и "\_ISO-8859-1".

Минимум Требуется KitKat 4.4 (API 19).

### Adoc.read <success\_nvar>, <result\_sexp>, <documentPath\_sexp>{,

**<charSet\_sexp>}**

Returns 1 if the document specified by the document path gives the full String content specified by <result\_sexp> successfully back. Otherwise 0 is returned.

For binary data specify the <charSet\_sexp> with the "\_ISO-8859-1" character set instead of the default "\_UTF-8".

A required permission is granted until it is revoked or the app deinstalled.

You can choose between following character sets: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" and "\_ISO-8859-1".

Min. KitKat 4.4 (API 19) is needed.

\*\*\*\*\*

Возвращает 1, если документ, указанный в пути к документу, полностью возвращает содержимое String, указанное в <result\_sexp>. В противном случае возвращается 0.

Для двоичных данных укажите <charSet\_sexp> с набором символов "\_ISO-8859-1" вместо значения по умолчанию "\_UTF-8".

Требуемое разрешение предоставляется до тех пор, пока оно не будет отменено или приложение не будет удалено.

Вы можете выбрать один из следующих наборов символов: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" и "\_ISO-8859-1".

Минимум Требуется KitKat 4.4 (API 19).

### Adoc.grab <result\_sexp>, <documentPath\_sexp>{, <charSet\_sexp>}

Puts the full String content specified by the <documentPath\_sexp> into <result\_sexp>. For binary data specify the <charSet\_sexp> with the "\_ISO-8859-1" character set instead of the default "\_UTF-8".

A required permission is granted until it is revoked or the app deinstalled.

You can choose between following character sets: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" and "\_ISO-8859-1".

Normally you will use this instead of Adoc.read, because you can read also only for reading specified content providers.

Keep in mind, that in this case a broken network connection is not checked.

\*\*\*\*\*

Помещает полное содержимое строки, указанное в <documentPath\_sexp>, в <result\_sexp>. Для двоичных данных укажите <charSet\_sexp> с набором символов "\_ISO-8859-1" вместо значения по умолчанию "\_UTF-8".

Требуемое разрешение предоставляется до тех пор, пока оно не будет отменено или приложение не будет удалено.

Вы можете выбрать один из следующих наборов символов: "\_US-ASCII", "\_UTF-8", "\_UTF-16", "\_UTF-16BE", "\_UTF-16LE" и "\_ISO-8859-1".

Обычно вы будете использовать это вместо Adoc.read, потому что вы можете читать только для чтения указанных поставщиков контента.

Имейте в виду, что в этом случае нарушенное сетевое соединение не проверяется.

### Adoc.exists <lvar>, <documentPath\_sexp>

Reports if the <documentPath\_sexp> directory or document exists. If the directory or document does not exist, the <lvar> will contain zero. If the file or directory does exist, the

<lvar> will be returned as non-zero.

A required permission is granted until it is revoked or the app deinstalled. Min. KitKat 4.4 (API 19) is needed.

\*\*\*\*\*

Сообщает, существует ли каталог или документ <documentPath\_sexp>. Если каталог или документ не существует, <lvar> будет содержать ноль. Если файл или каталог существует,

<lvar> будет возвращен как ненулевой.

Требуемое разрешение предоставляется до тех пор, пока оно не будет отменено или приложение не будет удалено. Минимум Требуется KitKat 4.4 (API 19).

### Adoc.revoke <success\_nvar>, <documentPath\_sexp>

Returns 1 if the document’s access permission specified by the document path was deleted successfully. Otherwise 0 is returned.

\*\*\*\*\*

Возвращает 1, если разрешение на доступ к документу, указанное в пути документа, было успешно удалено. В противном случае возвращается 0.

Min. KitKat 4.4 (API 19) is needed.

### Bluetooth

**Bt.write {<exp> {,|;}} ...**

**Bt.utf\_8.write {<exp> {,|;}} ...**

Writes data to the Bluetooth connection.

If the comma (,) separator is used then a comma will be printed between the values ofthe expressions.

If the semicolon (;) separator is used then nothing will separate the values ofthe expressions.

If the semicolon is at the end of the line, the output will be transmitted immediately, with no newline character(s) added.

The parameters are the same as the **Print** parameters. This command is essentially a

**Print** to the Bluetooth connection, with two differences:

* + - Only one byte is transmitted for each character; the upper byte is discarded. Binary data and ASCII text are sent correctly, but Unicode characters maynot be.

If you need the full Unicode character set use **Bt.utf\_8.write**.

* + - Alinethatendswithasemicolonissentimmediately,withnonewline character(s)added.

This command with no parameters sends a newline character to the Bluetooth connection.

\*\*\*\*\*

Записывает данные в соединение Bluetooth.

Если используется разделитель запятой (,), то между значениями выражений будет напечатана запятая.

Если используется разделитель точки с запятой (;), то ничто не будет разделять значения выражений.

Если точка с запятой находится в конце строки, выходные данные будут переданы немедленно, без добавления символа новой строки.

Параметры такие же, как параметры печати. Эта команда по сути

Печать через соединение Bluetooth, с двумя отличиями:

Только один байт передается для каждого символа; старший байт отбрасывается. Двоичные данные и текст ASCII отправляются корректно, но символы Unicode могут отсутствовать.

Если вам нужен полный набор символов Unicode, используйте Bt.utf\_8.write.

Сразу же после этого добавляются две строки:

Эта команда без параметров отправляет символ новой строки в соединение Bluetooth.

### Bt.read.bytes <svar>Bt.utf\_8.read.bytes <svar>

The next available message is placed into the specified string variable. If there is no message then the string variable will be returned with an empty string ("").

Each message byte is placed in one character of the string; the upper byte of each character is 0. This is similar to **Byte.read.buffer**, which reads binary data from a file into a buffer string.

If you need the full Unicode character set use **Bt.utf\_8.read.bytes**.

\*\*\*\*\*

Следующее доступное сообщение помещается в указанную строковую переменную. Если сообщения нет, то строковая переменная будет возвращена с пустой строкой ("").

Каждый байт сообщения помещается в один символ строки; старший байт каждого символа равен 0. Это похоже на Byte.read.buffer, который считывает двоичные данные из файла в строку буфера.

Если вам нужен полный набор символов Unicode, используйте Bt.utf\_8.read.bytes.

### Bt.open

Opens Bluetooth in Listen Mode. If you do not have Bluetooth enabled (using the Android SettingsApplication) then the person running the program will be asked whether Bluetooth should be enabled. After **Bt.open** is successfully executed, the code will listen for a device that wants to connect.

The optional parameter determines if BT will listen for a secure or insecure connection. If no parameter is given or if the parameter is 1, then a secure connection request will be listened for. Otherwise, an insecure connection will be listened for. It is not possible to listen for either a secure or insecure connection with one **Bt.open** command because the Android API requires declaring a specific secure/insecure open.

If **Bt.open** is used in graphics mode (after **Gr.open**), you will need to insert a **Pause 500**

statement after the **Bt.open** statement.

\*\*\*\*\*

Открывает Bluetooth в режиме прослушивания. Если у вас не включен Bluetooth (с помощью приложения «Настройки Android»), то у человека, запустившего программу, появится вопрос, нужно ли включить Bluetooth. После успешного выполнения Bt.open код будет прослушивать устройство, которое хочет подключиться.

Необязательный параметр определяет, будет ли BT прослушивать безопасное или незащищенное соединение. Если параметр не указан или параметр равен 1, запрос безопасного соединения будет прослушиваться. В противном случае небезопасное соединение будет прослушиваться. Невозможно прослушать безопасное или небезопасное соединение с помощью одной команды Bt.open, потому что API Android требует объявления определенного безопасного / незащищенного открытия.

Если Bt.open используется в графическом режиме (после Gr.open), вам нужно будет вставить Pause 500

заявление после заявления Bt.open.

### Bt.reconnect

This command will attempt to reconnect to a device that was previously connected (during this Run) with **Bt.connect** or a prior **Bt.reconnect**. The command cannot be used to reconnect to a device that was connected following a **Bt.open** or **Bt.disconnect** command (i.e. from the **Listening** status).

You should monitor the Bluetooth status for **Connected** (3) after executing **Bt.reconnect**.

\*\*\*\*\*

Эта команда попытается повторно подключиться к устройству, которое ранее было подключено (во время этого прогона) с помощью Bt.connect или предыдущего Bt.reconnect. Эту команду нельзя использовать для повторного подключения к устройству, которое было подключено с помощью команды Bt.open или Bt.disconnect (т. Е. Из состояния прослушивания).

Вы должны следить за состоянием Bluetooth для Connected (3) после выполнения Bt.reconnect.

### Bt.status {{<connect\_var>}{, <name\_svar>}{, <address\_svar>}}

Gets the current Bluetooth status and places the information in the return variables. The available data are the current connection status (in <connect\_var>), and the friendly name and MAC address of your Bluetooth hardware (in <name\_svar> and <address\_svar>).

All parameters are optional; use commas to indicate omitted parameters (see Optional Parameters).

If the connection status variable <connect\_var> is present, it may be either a numeric variable or a string variable. The table shows the possible return values of each type:

\*\*\*\*\*

Получает текущее состояние Bluetooth и помещает информацию в возвращаемые переменные. Доступными данными являются текущее состояние соединения (в <connect\_var>) и понятное имя и MAC-адрес вашего оборудования Bluetooth (в <name\_svar> и <address\_svar>).

Все параметры являются необязательными; используйте запятые для обозначения пропущенных параметров (см. Дополнительные параметры).

Если переменная состояния соединения <connect\_var> присутствует, она может быть либо числовой, либо строковой переменной. В таблице приведены возможные возвращаемые значения каждого типа:

|  |  |  |
| --- | --- | --- |
| **Numeric Value** | **String Value** | **Meaning** |
| -1 | Not enabled | Bluetooth not enabled |
| 0 | Idle | Nothing going on |
| 1 | Listening | Listening for connection |
| 2 | Connecting | Connecting to another device |
| 3 | Connected | Connected to another device |

If the device name string variable <name\_svar> is present, it is set to the friendly device name. If your device has no Bluetooth radio, the string will be empty.

If the address string variable <address\_svar> is present, it is set to the MAC address of your Bluetooth hardware, represented as a string of six hex numbers separated by colons: "00:11:22:AA:BB:CC".

\*\*\*\*\*

Если присутствует строковая переменная имени устройства <name\_svar>, ей присваивается понятное имя устройства. Если на вашем устройстве нет радио Bluetooth, строка будет пустой.

Если присутствует переменная адресной строки <address\_svar>, для нее устанавливается MAC-адрес вашего оборудования Bluetooth, представленный в виде строки из шести шестнадцатеричных чисел, разделенных двоеточиями: «00: 11: 22: AA: BB: CC».

### Bt.write {<exp> {,|;}} …

Writes data to the Bluetooth connection.

If the comma (,) separator is used then a comma will be printed between the values ofthe expressions.

If the semicolon (;) separator is used then nothing will separate the values ofthe expressions.

If the semicolon is at the end of the line, the output will be transmitted immediately, with no newline character(s) added.

The parameters are the same as the **Print** parameters. This command is essentially a

**Print** to the Bluetooth connection, with two differences:

* + - Only one byte is transmitted for each character; the upper byte is discarded. Binary data and ASCII text are sent correctly, but Unicode characters maynot be.
    - Alinethatendswithasemicolonissentimmediately,withnonewline character(s)added.

This command with no parameters sends a newline character to the Bluetooth connection.

\*\*\*\*\*

Записывает данные в соединение Bluetooth.

Если используется разделитель запятой (,), то между значениями выражений будет напечатана запятая.

Если используется разделитель точки с запятой (;), то ничто не будет разделять значения выражений.

Если точка с запятой находится в конце строки, выходные данные будут переданы немедленно, без добавления символа новой строки.

Параметры такие же, как параметры печати. Эта команда по сути

Печать через соединение Bluetooth, с двумя отличиями:

Только один байт передается для каждого символа; старший байт отбрасывается. Двоичные данные и текст ASCII отправляются корректно, но символы Unicode могут отсутствовать.

Сразу же после этого добавляются две строки:

Эта команда без параметров отправляет символ новой строки в соединение Bluetooth.

### Bt.read.ready <nvar>

Reports in the numeric variable the number of messages ready to be read. If the value is greater than zero then the messages should be read until the queue is empty.

\*\*\*\*\*

Сообщает в числовой переменной количество сообщений, готовых для чтения. Если значение больше нуля, сообщения следует читать до тех пор, пока очередь не станет пустой.

### OnBtReadReady:

Interrupt label that traps the arrival of a message received on the Bluetooth channel (see "Interrupt Labels"). If a Bluetooth message is ready (**Bt.read.ready** would return a non- zero value) BASIC! executes the statements after the **OnBtReady:** label, where you can read and handle the message. When done, execute the **Bt.onReadReady.Resume** command to resume the interrupted program.

\*\*\*\*\*

Метка прерывания, которая фиксирует прибытие сообщения, полученного по каналу Bluetooth (см. «Метки прерывания»). Если сообщение Bluetooth готово (Bt.read.ready вернет ненулевое значение) BASIC! выполняет операторы после метки OnBtReady :, где вы можете прочитать и обработать сообщение. Когда закончите, выполните команду Bt.onReadReady.Resume, чтобы возобновить прерванную программу.

##### Bt.onReadReady.resume

Resumes execution at the point in the program where it was interrupted by the Bluetooth Read Ready event.

\*\*\*\*\*

Возобновляет выполнение в той точке программы, где оно было прервано событием Bluetooth Read Ready.

### Bt.read.bytes <svar>

The next available message is placed into the specified string variable. If there is no message then the string variable will be returned with an empty string ("").

Each message byte is placed in one character of the string; the upper byte of each character is 0. This is similar to **Byte.read.buffer**, which reads binary data from a file into a buffer string.

\*\*\*\*\*

Следующее доступное сообщение помещается в указанную строковую переменную. Если сообщения нет, то строковая переменная будет возвращена с пустой строкой ("").

Каждый байт сообщения помещается в один символ строки; старший байт каждого символа равен 0. Это похоже на Byte.read.buffer, который считывает двоичные данные из файла в строку буфера.

### Bt.device.name <svar>

Returns the name of the connected device in the string variable. A run-time error will be generated if no device (Status <> 3) is connected.

\*\*\*\*\*

Возвращает имя подключенного устройства в строковой переменной. Ошибка времени выполнения будет генерироваться, если не подключено ни одно устройство (Status <> 3).

### Bt.set.UUID <sexp>

A Universally Unique Identifier (UUID) is a standardized 128-bit format for a string ID used to uniquely identify information. The point of a UUID is that it's big enough that you can select any random 128-bit number and it won't clash with any other number selected similarly. In this case, it's used to uniquely identify your application's Bluetooth service. To get a UUID to use with your application, you can use one of the many random UUID generators on the web.

Many devices have common UUIDs for their particular application. The default BASIC! UUID is the standard Serial Port Profile (SPP) UUID: "00001101-0000-1000-8000- 00805F9B34FB".

You can change the default UUID using this command.

\*\*\*\*\*

Универсальный уникальный идентификатор (UUID) - это стандартизированный 128-битный формат для идентификатора строки, используемый для уникальной идентификации информации. Суть UUID в том, что он достаточно большой, чтобы вы могли выбрать любое случайное 128-битное число, и он не будет конфликтовать с любым другим выбранным аналогичным числом. В этом случае он используется для уникальной идентификации службы Bluetooth вашего приложения. Чтобы получить UUID для использования с вашим приложением, вы можете использовать один из множества случайных генераторов UUID в сети.

Многие устройства имеют общие UUID для своего конкретного приложения. Бейсик по умолчанию! UUID - это стандартный UUID профиля последовательного порта (SPP): «00001101-0000-1000-8000-00805F9B34FB».

С помощью этой команды вы можете изменить UUID по умолчанию.

Some information about 16 bit and 128 bit UUIDs can be found at: <http://farwestab.wordpress.com/2011/02/05/some-tips-on-android-and-bluetooth/>

##### Run {{<filename\_sexp> }, <data\_sexp>}

This command will terminate the running of the current program and then load and run the BASIC! program named in the filename string expression. The filename is relative to BASIC’s "source/" directory. If the filename is "program.bas" and your <pref base drive> is "/sdcard" (the default), then the file "/sdcard/rfo-basic/source/program.bas" will be executed.

The optional data string expression provides for the passing of data to the next program. The passed data can be accessed in the next program by referencing the special variable, **##$**.

**Run** programs can be chained. A program loaded and run by means of the **Run** command can also run another program file. This chain can be a long as needed.

When the last program in a **Run** chain ends, tapping the BACK key will display the original program in the BASIC! Editor.

When a program ends with an error, the Editor tries to highlight the line where the error occurred. If the program with the error was started by a **Run** command, the Editor does not have that program loaded. Any highlighting that may be displayed is meaningless.

If the first parameter = "" or not given, a program with the current file path will load and run (if not compiled maybe with different code). In APK mode the App will be relaunched.

Note: "file://" + <full\_path\_svar> can now be used for absolute filepaths.

IF <filename\_sexp> = "" or is not defined, the current program itself is restarting again. Be carefully for not to be caught in an endless loop.

\*\*\*\*\*

Эта команда прервет выполнение текущей программы, а затем загрузит и запустит BASIC! Программа, названная в строковом выражении имени файла. Имя файла относится к каталогу «source /» BASIC. Если имя файла «program.bas», а ваш <pref base drive> - «/ sdcard» (по умолчанию), то будет выполнен файл «/sdcard/rfo-basic/source/program.bas».

Необязательное строковое выражение данных обеспечивает передачу данных в следующую программу. Доступ к переданным данным в следующей программе можно получить, обратившись к специальной переменной ## $.

Запускать программы можно по цепочке. Программа, загружаемая и запускаемая с помощью команды «Выполнить», также может запускать другой программный файл. Эта цепочка может быть длинной по мере необходимости.

Когда последняя программа в цепочке выполнения заканчивается, нажатие клавиши BACK покажет оригинальную программу в бейсике! Редактор.

Когда программа завершается с ошибкой, редактор пытается выделить строку, где произошла ошибка. Если программа с ошибкой была запущена командой «Выполнить», редактор не загружает эту программу. Любое выделение, которое может отображаться, не имеет смысла.

Если первый параметр = "" или не указан, программа с текущим путем к файлу будет загружена и запущена (если не скомпилирована, возможно, с другим кодом). В режиме APK приложение будет перезапущено.

Примечание: «file: //» + <full\_path\_svar> теперь можно использовать для абсолютных путей к файлам.

Если <filename\_sexp> = "" или не определено, текущая программа снова перезапускается. Будьте осторожны, чтобы не оказаться в бесконечной петле.

Example:

! Run "Files.bas"

! Run "Files.bas", ""

! Run %Endless loop?

! Run "" %Endless loop?

! Run " " %File Not found PROGRAM.INFO bdi

BUNDLE.GET bdi, "BasName", BasName$ PRINT BasName$

FILE.ROOT aPath$,"\_Source" IF ##$ <> "1"

! Run , "1"

! RUN "file://" + aPath$ + "/" + BasName$, "1" ENDIF

! Run "../source/Files.bas", "" FILE.ROOT dp$, "\_Source"

? dp$

RUN "file://"+dp$ + "/" + "Files.bas"

GitHub#215

### BigDecimal

In BASIC! numbers are normally stored as values from type Double. For precise calculation the BigDecimal engine is our choice. The best way to store these values lossless in BASIC! is using Strings.

You do not need always BigDecimals instead of Double. For addition and subtraction you can use also the exponential shifting hack in the range of the bounds of Double numbers.

k =10^21

c = (a\*k + b\*k)/k or c = (a\*k - b\*k) /k

10^21 is the useful maximum. If you want three correct digits behind the decimal point use 10^5. **But with Double you get maximal only 15 correct digits.**

**\*\*\*\*\***

**В бейсике! числа обычно хранятся в виде значений типа Double. Для точного расчета движок BigDecimal - наш выбор. Лучший способ хранить эти значения без потерь в бейсике! использует строки.**

**Вам не нужно всегда BigDecimals вместо Double. Для сложения и вычитания вы можете использовать также хак с экспоненциальным сдвигом в диапазоне границ двойных чисел.**

**к = 10 ^ 21**

**c = (a \* k + b \* k) / k или c = (a \* k - b \* k) / k**

**10 ^ 21 - полезный максимум. Если вы хотите, чтобы за десятичной точкой стояли три правильные цифры, используйте 10 ^ 5. Но с Double вы получите максимум только 15 правильных цифр.**

Source for PI: [http://www.pibel.de](http://www.pibel.de/)

Source for e:<http://www.gutenberg.org/files/127/127.txt>

##### BigD.add <result\_svar>, <first\_sexp>, <second\_sexp>

Returns a new BigDecimal as a String whose value is <first\_sexp> + <second\_sexp>. The scale of the result is the maximum of the scales of the two arguments.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой <first\_sexp> + <second\_sexp>. Масштаб результата - это максимум шкал двух аргументов.

##### BigD.sum <result\_svar>, Array$[]

Returns a new BigDecimal as a String whose value is the sum of all array String items. The scale of the result is the maximum of the scales of all arguments.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой является суммой всех элементов массива. Масштаб результата - это максимум шкал всех аргументов.

##### BigD.subtract <result\_svar>, <first\_sexp>, <second\_sexp>

Returns a new BigDecimal as a String whose value is <first\_sexp> - <second\_sexp>. The scale of the result is the maximum of the scales of the two arguments.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки со значением <first\_sexp> - <second\_sexp>. Масштаб результата - это максимум шкал двух аргументов.

##### BigD.multiply <result\_svar>, <first\_sexp>, <second\_sexp>

Returns a new BigDecimal as a String whose value is <first\_sexp> \* <second\_sexp>. The scale of the result is the sum of the scales of the two arguments.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой <first\_sexp> \* <second\_sexp>. Шкала результата представляет собой сумму шкал двух аргументов.

##### BigD.divide <result\_svar>, <first\_sexp>, <second\_sexp>, <scale\_nexp>,

**<roundingMode\_sexp>**

Returns a new BigDecimal as a String whose value is <first\_sexp> / divisor

<second\_sexp>. As scale of the result the parameter scale is used. If rounding is required to meet the specified scale, then the specified rounding mode <roundingMode\_sexp> is applied.

See also BigD.round for rounding details.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой <first\_sexp> / divisor

<Second\_sexp>. В качестве шкалы результата используется шкала параметров. Если для округления требуется соответствие указанному масштабу, то применяется указанный режим округления <roundingMode\_sexp>.

Смотрите также BigD.round для деталей округления.

##### BigD.remainDividing <integral\_svar>, <remainder\_svar>, <first\_sexp>,

**<second\_sexp>**

Returns a BigDecimal Strings which contain the integral part of <first\_sexp> / divisor

<second\_sexp> as <integral\_svar> and the remainder <first\_sexp> -

<first\_sexp>/int(divisor) \* divisor as <remainder\_svar>.

\*\*\*\*\*

Возвращает BigDecimal Strings, которые содержат неотъемлемую часть <first\_sexp> / divisor

<second\_sexp> как <integra\_svar> и остаток <first\_sexp> -

<first\_sexp> / int (divisor) \* делитель как <remainder\_svar>.

See also MOD()

### MOD(<nexp1>, <nexp2>)

Returns the remainder of <nexp1> divided by <nexp2>. If <nexp2> is 0, the function ~~generates a runtime error~~returns NaN.

\*\*\*\*\*

Возвращает остаток от <nexp1>, разделенный на <nexp2>. Если <nexp2> равно 0, функция генерирует ошибку времени выполнения, возвращает NaN.

##### BigD.abs <result\_svar>, <first\_sexp>

Returns a BigDecimal as a String whose value is the absolute value of <first\_sexp>. The scale of the result is the same as the scale of <first\_sexp>.

\*\*\*\*\*

Возвращает BigDecimal в виде строки, значение которой является абсолютным значением <first\_sexp>. Масштаб результата такой же, как масштаб <first\_sexp>.

See also ABS()

##### BigD.frac <result\_svar>, <first\_sexp>

Returns a new BigDecimal as a String whose value is the fractional part of <first\_sexp>. The scale of the result is the same as the scale of <first\_sexp>.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой является дробной частью <first\_sexp>. Масштаб результата такой же, как масштаб <first\_sexp>.

### FRAC(<nexp>)

Returns the fractional part of <nexp>. 3.4 becomes 0.4 and -3.4 becomes -0.4.

**FRAC(n)** is equivalent to "n – **INT(n)**".

\*\*\*\*\*

Возвращает дробную часть <nexp>. 3.4 становится 0,4, а -3,4 становится -0,4.

FRAC (n) эквивалентно "n - INT (n)".

##### BigD.int <result\_svar>, <first\_sexp>

Returns a new BigDecimal as a String whose value is the integral part of <first\_sexp>. See also INT(), ROUND()

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой является неотъемлемой частью <first\_sexp>. Смотрите также INT (), ROUND ()

### INT(<nexp>)

Returns the integer part of <nexp>. 3.X becomes 3 and -3.X becomes -3. This operation may also be called truncation, rounding down, or rounding toward zero.

So the decimal point and the digits behind him will be deleted.

The int() function in most other BASIC dialects works different. See ROUND() for more details.

\*\*\*\*\*

Возвращает целочисленную часть <nexp>. 3.X становится 3, а -3.X становится -3. Эту операцию также можно назвать усечением, округлением вниз или округлением до нуля.

Таким образом, десятичная точка и цифры за ним будут удалены.

Функция int () в большинстве других основных диалектов работает по-другому. Смотрите ROUND () для более подробной информации.

See also INT(), FRAC(), ROUND()

##### BigD.compare <result\_nvar>, <first\_sexp>, <second\_sexp>

Compares BigDecimal <first\_sexp> with <second\_sexp>. Returns one of the three values 1, 0, or -1. The method behaves as if <first\_sexp> - <second\_sexp> is computed. If this difference is > 0 then 1 is returned, if the difference is < 0 then -1 is returned, and if the difference is 0 then 0 is returned. This means, that if two decimal instances are compared which are equal in value but differ in scale, then these two instances are considered as equal.

\*\*\*\*\*

Сравнивает BigDecimal <first\_sexp> с <second\_sexp>. Возвращает одно из трех значений 1, 0 или -1. Метод ведет себя так, как будто вычислено <first\_sexp> - <second\_sexp>. Если эта разница> 0, то возвращается 1, если разница <0, возвращается -1, а если разница 0, возвращается 0. Это означает, что если сравниваются два десятичных экземпляра, которые равны по значению, но различаются по масштабу, то эти два экземпляра считаются равными.

##### BigD.equals <result\_nvar>, <first\_sexp>, <second\_sexp>

Returns 1.0 if <first\_sexp> and <second\_sexp> are BigDecimal instances and equal otherwise <result\_svar> returns 0.0. Two big decimals are equal if their unscaled value **and** their scale is equal. For example, 1.0 (10\*10-1) is not equal to 1.00 (100\*10-2).

Similarly, zero instances are not equal if their scale differs.

\*\*\*\*\*

Возвращает 1.0, если <first\_sexp> и <second\_sexp> являются экземплярами BigDecimal и равны, в противном случае <result\_svar> возвращает 0.0. Два больших десятичных знака равны, если их немасштабированное значение и их масштаб равны. Например, 1,0 (10 \* 10-1) не равно 1,00 (100 \* 10-2).

Аналогично, нулевые экземпляры не равны, если их масштаб отличается.

##### BigD.toDouble <result\_nvar>, <first\_sexp>

Returns <first\_sexp> BigDecimal as a double value. NaN (Not a Number), Double.POSITIVE\_INFINITY or Double.NEGATIVE\_INFINITY are not supported and thrown a runtime error.

Note, that if the unscaled value has more than 53 significant digits, then this decimal cannot be represented exactly in a double variable. In this case the result is rounded.

\*\*\*\*\*

Возвращает <first\_sexp> BigDecimal в виде двойного значения. NaN (не число), Double.POSITIVE\_INFINITY или Double.NEGATIVE\_INFINITY не поддерживаются и выдают ошибку времени выполнения.

Обратите внимание, что если немасштабированное значение имеет более 53 значащих цифр, то это десятичное число не может быть точно представлено в двойной переменной. В этом случае результат округляется.

##### BigD.FromDouble <result\_svar>, <number\_nexp>

Returns a new BigDecimal as a String from the Double <number\_nexp>.

NaN (Not a Number), Double.POSITIVE\_INFINITY or Double.NEGATIVE\_INFINITY are not supported and thrown a runtime error.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки из двойного <number\_nexp>.

NaN (не число), Double.POSITIVE\_INFINITY или Double.NEGATIVE\_INFINITY не поддерживаются и выдают ошибку времени выполнения.

##### BigD.toBase <result\_svar>, <string\_sexp>, <base\_sexp>

Returns a new BigDecimal as a String whose value is the result of the String

<string\_sexp> encoded by the base "\_Bin", "\_Oct" or "\_Hex"(default, if wrong also).

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой является результатом строки

<string\_sexp> кодируется с помощью базы "\_Bin", "\_Oct" или "\_Hex" (по умолчанию, если также неверно).

##### BigD.FromBase <result\_svar>, <string\_sexp>, <base\_sexp>

Returns a new BigDecimal as a String whose value is the result of the encoded String

<string\_sexp> decoded by the base "\_Bin", "\_Oct" or "\_Hex"(default, if wrong also).

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой является результатом закодированной строки

<string\_sexp> декодируется с помощью базы "\_Bin", "\_Oct" или "\_Hex" (по умолчанию, если также неверно).

##### BigD.hashcode <result\_svar>, <first\_sexp>

Returns a hash code as a String for <first\_sexp> as a BigDecimal.

\*\*\*\*\*

Возвращает хеш-код в виде строки для <first\_sexp> как BigDecimal.

##### BigD.max <result\_svar>, <first\_sexp>, <second\_sexp>

Returns a new BigDecimal as a String whose value is the maximum of <first\_sexp> and

<second\_sexp> as BigDecimal.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой является максимальным значением <first\_sexp> и

<second\_sexp> как BigDecimal.

See also MAX()

##### BigD.min <result\_svar>, <first\_sexp>, <second\_sexp>

Returns a new BigDecimal as a String whose value is the minimum of <first\_sexp> and

<second\_sexp> as BigDecimal.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значением которой является минимум <first\_sexp> и

<second\_sexp> как BigDecimal.

See also MIN()

##### BigD.movePointLeft <result\_svar>, <first\_sexp>, <n\_nexp>

Returns a new BigDecimal instance as a String where the decimal point has been moved n places to the left. If n < 0 then the decimal point is moved -n places to the right.

The result is obtained by changing its scale. If the scale of the result becomes negative, then its precision is increased such that the scale is zero.

Note, that movePointLeft with n=0 returns a result which is mathematically equivalent, but which has scale ≥ 0.

\*\*\*\*\*

Возвращает новый экземпляр BigDecimal в виде строки, в которой десятичная точка была перемещена на n мест влево. Если n <0, то десятичная точка сдвигается на n мест вправо.

Результат получается путем изменения его масштаба. Если масштаб результата становится отрицательным, то его точность увеличивается так, что масштаб равен нулю.

Обратите внимание, что movePointLeft с n = 0 возвращает результат, который математически эквивалентен, но имеет масштаб ≥ 0.

##### BigD.movePointRight <result\_svar>, <first\_sexp>, <n\_nexp>

Returns a new BigDecimal instance as a String where the decimal point has been moved n places to the right. If n < 0 then the decimal point is moved -n places to the left.

The result is obtained by changing its scale. If the scale of the result becomes negative, then its precision is increased such that the scale is zero.

Note, that movePointRight with n=0 returns a result which is mathematically equivalent, but which has scale ≥ 0.

\*\*\*\*\*

Возвращает новый экземпляр BigDecimal в виде строки, в которой десятичная точка была перемещена на n позиций вправо. Если n <0, то десятичная точка перемещается на n мест влево.

Результат получается путем изменения его масштаба. Если масштаб результата становится отрицательным, то его точность увеличивается так, что масштаб равен нулю.

Обратите внимание, что movePointRight с n = 0 возвращает результат, который математически эквивалентен, но имеет масштаб ≥ 0.

##### BigD.pow <result\_svar>, <first\_sexp>, <n\_nexp>

Returns a new BigDecimal as a String whose value is <first\_sexp> raised to the <n\_nexp> power → <first\_sexp>**n**. The scale of the result is n \* the scale from <first\_sexp>.

BigD.pow x$, 0.0, r$ returns "1", even if x$ = "0".

Implementation Note: The implementation is based on the ANSI standard X3.274-1996 algorithm.

Operation Note: n have to be in the bounds of 0 to 999999999.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой <first\_sexp> повышается до значения <n\_nexp> power → <first\_sexp> n. Масштаб результата равен n \* шкале от <first\_sexp>.

BigD.pow x $, 0.0, r $ возвращает «1», даже если x $ = «0».

Замечание по реализации: Реализация основана на алгоритме стандарта ANSI X3.274-1996.

Операция Примечание: n должно быть в пределах от 0 до 999999999.

See also POW()

##### BigD.sqr <result\_svar>, <first\_sexp>, <scale\_nexp>

Returns a new BigDecimal as a String whose value is the closest approximation of the positive square root of <first\_sexp>. If the value of <first\_sexp> is negative, the function generates a runtime error. The maximum scale is given by <scale\_nexp>.

\*\*\*\*\*

Возвращает новый BigDecimal в виде строки, значение которой является ближайшим приближением положительного квадратного корня из <first\_sexp>. Если значение <first\_sexp> отрицательно, функция генерирует ошибку во время выполнения. Максимальный масштаб задается как <scale\_nexp>.

See also: BigD.scale, SQR()

##### BigD.precision <result\_nvar>, <first\_sexp>

Returns the precision as a Double of <first\_sexp> as BigDecimal. The precision is the number of decimal digits used to represent this decimal. It is equivalent to the number of digits of the unscaled value. The precision of 0 is 1 (independent of the scale).

\*\*\*\*\*

Возвращает точность в виде Double of <first\_sexp> как BigDecimal. Точность - это количество десятичных цифр, используемых для представления этого десятичного числа. Это эквивалентно количеству цифр немасштабированного значения. Точность 0 равна 1 (независимо от масштаба).

**BigD.round <result\_svar>, <first\_sexp>, <scale\_nexp>, <roundingMode\_sexp>** Returns a new BigDecimal as a String whose value is <first\_sexp>, rounded according to scale and rounding mode. As scale of the result the parameter scale is used. If rounding is required to meet the specified scale, then the specified rounding mode

<roundingMode\_nexp> is applied. There are eight rounding modes:

\*\*\*\*\*

новый BigDecimal в виде строки со значением <first\_sexp>, округленной в соответствии с масштабом и режимом округления. В качестве шкалы результата используется шкала параметров. Если требуется округление в соответствии с указанным масштабом, то указанный режим округления

<roundingMode\_nexp> применяется. Есть восемь режимов округления:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mode:** | **Meaning:** | **-3.8** | **-3.5** | **-3.1** | **-3.0** | **3.0** | **3.1** | **3.5** | **3.8** |
| "HD" | Half-down | -4.0 | -3.0 | -3.0 | -3.0 | 3.0 | 3.0 | 3.0 | 4.0 |
| "HE" | Half-even | -4.0 | -4.0 | -3.0 | -3.0 | 3.0 | 3.0 | 4.0 | 4.0 |
| "HU" | Half-up | -4.0 | -4.0 | -3.0 | -3.0 | 3.0 | 3.0 | 4.0 | 4.0 |
| "D" | Down | -3.0 | -3.0 | -3.0 | -3.0 | 3.0 | 3.0 | 3.0 | 3.0 |
| "U" | Up | -4.0 | -4.0 | -4.0 | -3.0 | 3.0 | 4.0 | 4.0 | 4.0 |
| "F" | Floor | -4.0 | -4.0 | -4.0 | -3.0 | 3.0 | 3.0 | 3.0 | 3.0 |
| "C" | Ceiling | -3.0 | -3.0 | -3.0 | -3.0 | 3.0 | 4.0 | 4.0 | 4.0 |
| "LI" | Legacy int() | -4.0 | -4.0 | -4.0 | -3.0 | 3.0 | 3.0 | 3.0 | 3.0 |

In this table, "down" means "toward zero" and "up" means "away from zero" (toward ±∞) In most cases is "**Half-Up"→"HU"** the best choice.

See also ROUND()

### ROUND(<value\_nexp>{, <scale\_nexp>{, <roundingMode\_sexp>}})

In it simplest form, **ROUND(<value\_nexp>)**, this function returns the closest whole number to <nexp>. You can use the optional parameters to specify more complex operations.

The <scale\_nexp> is an optional decimal place count. It sets the number of places to the right of the decimal point. The last digit is rounded. The decimal place count must be ≥0. Omitting the parameter is the same as setting it tozero.

The <roundingMode\_sexp> is an optional rounding mode. It is a one- or two-character mnemonic code that tells **ROUND()** what kind of rounding to do. It is not case-sensitive. There are eight rounding modes:

\*\*\*\*\*

В простейшей форме, ROUND (<value\_nexp>), эта функция возвращает ближайшее целое число к <nexp>. Вы можете использовать дополнительные параметры для указания более сложных операций.

<Scale\_nexp> - это необязательное количество знаков после запятой. Он устанавливает количество мест справа от десятичной точки. Последняя цифра округляется. Количество знаков после запятой должно быть ≥0. Пропуск параметра аналогичен установке его в ноль.

<RoundingMode\_sexp> является необязательным режимом округления. Это мнемонический код, состоящий из одного или двух символов, который сообщает ROUND (), какое округление следует выполнить. Он не чувствителен к регистру. Есть восемь режимов округления:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Mode:** | **Meaning:** | **-3.8** | **-3.5** | **-3.1** | **-3.0** | **3.0** | **3.1** | **3.5** | **3.8** |
| "HD" | Half-down | -4.0 | -3.0 | -3.0 | -3.0 | 3.0 | 3.0 | 3.0 | 4.0 |
| "HE" | Half-even | -4.0 | -4.0 | -3.0 | -3.0 | 3.0 | 3.0 | 4.0 | 4.0 |
| "HU" | Half-up | -4.0 | -4.0 | -3.0 | -3.0 | 3.0 | 3.0 | 4.0 | 4.0 |
| "D" | Down | -3.0 | -3.0 | -3.0 | -3.0 | 3.0 | 3.0 | 3.0 | 3.0 |
| "U" | Up | -4.0 | -4.0 | -4.0 | -3.0 | 3.0 | 4.0 | 4.0 | 4.0 |
| "F" | Floor | -4.0 | -4.0 | -4.0 | -3.0 | 3.0 | 3.0 | 3.0 | 3.0 |
| "C" | Ceiling | -3.0 | -3.0 | -3.0 | -3.0 | 3.0 | 4.0 | 4.0 | 4.0 |
| "LI" | Legacy int()\* | -4.0 | -4.0 | -4.0 | -3.0 | 3.0 | 3.0 | 3.0 | 3.0 |

In this table, "down" means "toward zero" and "up" means "away from zero" (toward

±∞)

"Half" refers to behavior when a value is half-way between rounding up and rounding down(x.5 or -x.5). "Half-down" rounds x.5 towards zero and "half-up" rounds x.5 away from zero.

"Half-even" is either "half-down" or "half-up", whichever would make the result **even**. 4.5 and 3.5 both round to 4.0. "Half-even" is also called "banker’s rounding", because it tends to average out rounding errors.

In most cases is **"Half-Up"→"HU"** the best choice.

**\*Legacy int() →"LI"** is compatible to the most other BASIC dialects

\*\*\*\*\*

В этой таблице «вниз» означает «к нулю», а «вверх» означает «от нуля» (к

± ∞)

«Половина» относится к поведению, когда значение находится на полпути между округлением вверх и округлением вниз (x.5 или -x.5). «Половина» округляет х.5 к нулю, а «половина» округляет х.5 от нуля.

«Half-even» - это «half-down» или «half-up», в зависимости от того, какой результат будет равномерным. 4,5 и 3,5 оба раунда до 4,0. «Половина» также называется «округлением банкира», поскольку имеет тенденцию усреднять ошибки округления.

В большинстве случаев лучшим выбором будет «Half-Up» → «HU».

\* Legacy int () → «LI» совместим с большинством других основных диалектов

|  |  |
| --- | --- |
| **Examples of Most Other BASIC Dialects** | **Examples of BASIC!** |
| myNumber = Int(99.8) ' Returns 99. | myNumber = round(99.8, 0, "LI") % Returns 99. |
| myNumber = Fix(99.8) ' Returns 99. | myNumber = INT(99.8) % Returns 99. |
| myNumber = Int(-99.8) ' Returns -100. | myNumber = round(-99.8, 0, "LI") % Returns -100. |
| myNumber = Fix(-99.8) ' Returns -99. | myNumber = INT(-99.8) % Returns -99. |
| **myNumber = Int(-99.2) ' Returns -100.** | **myNumber = round(-99.2, 0, "LI") % Returns -100.** |
| myNumber = Fix(-99.2) ' Returns -99. | myNumber = INT(-99.2) % Returns -99. |

So BASIC!’s **INT(<nexp>)** works like the Fix() command in the table above.

If you do not provide a <roundingMode\_sexp>, **ROUND()** adds +0.5 and rounds down (toward zero). This is a bequest ~~behavior, copied~~from earlier versions of BASIC!. **ROUND(n)** is NOT the same as **ROUND(n, 0)**.

\*\*\*\*ç

Таким образом, INT (<nexp>) BASIC! Работает как команда Fix () в таблице выше.

Если вы не предоставите <roundingMode\_sexp>, ROUND () добавляет +0,5 и округляет (до нуля). Это наследственное поведение, скопированное из более ранних версий BASIC !. ROUND (n) НЕ совпадает с ROUND (n, 0).

**ROUND()** generates a runtime error if <count\_nexp> < 0 or <mode\_sexp> is not valid.

\*\*\*\*\*

ROUND () генерирует ошибку времени выполнения, если <count\_nexp> <0 или <mode\_sexp> недопустимо.

Examples:

**pi=ROUND(3.14159) % pi is3.0**

**pi =ROUND(3.14159,2) % pi is3.14**

**pi = ROUND(3.14159,, "U") % pi is4.0**

**pi = ROUND(3.14159,4,"F") % pi is 3.1415 negpi = ROUND(-3.14159,4, "D") % negpi is-3.1416**

Note that **FLOOR(n)** is exactly the same as **ROUND(n, 0, "F")**, but **FLOOR(n)** is a little faster. In the same way, **CEIL(n)** is the same as **ROUND(n, 0, "C")**, and **INT(n)** is the same as **ROUND(n, 0, "D")**.

\*\*\*\*\*

Обратите внимание, что FLOOR (n) точно такой же, как ROUND (n, 0, "F"), но FLOOR (n) немного быстрее. Таким же образом, CEIL (n) является таким же, как ROUND (n, 0, "C"), а INT (n) является таким же, как ROUND (n, 0, "D").

##### BigD.scale <result\_nvar>, <first\_sexp>

Returns the scale of <first\_sexp> BigDecimal as a Double. The scale is the number of digits behind the decimal point. The value of <first\_sexp> BigDecimal is the unsignedValue

\* 10-scale. If the scale is negative, then <first\_sexp> BigDecimal represents a big integer.

\*\*\*\*\*

Возвращает масштаб <first\_sexp> BigDecimal в виде Double. Шкала - это количество цифр за десятичной точкой. Значение <first\_sexp> BigDecimal является unsignedValue

\* 10 шкал. Если масштаб отрицательный, то <first\_sexp> BigDecimal представляет большое целое число.

##### BigD.sign <result\_nvar>, <first\_sexp>

Returns the signum function of the BigDecimal value of <first\_sexp> as a Double, representing its sign.

Returns

-1 if BigDecimal of <first\_sexp> <0,

1. if BigDecimal of <first\_sexp> =0,
2. if BigDecimal of <first\_sexp> >0.

\*\*\*\*\*

Возвращает функцию signum значения BigDecimal для <first\_sexp> в виде Double, представляющего его знак.

Возвращает

-1, если BigDecimal из <first\_sexp> <0,

если BigDecimal из <first\_sexp> = 0,

если BigDecimal из <first\_sexp>> 0.

See alsoSIGN()

##### BigD.nanoTime <result\_svar>

Returns the current timestamp of the most precise timer available on the local system, in nanoseconds. Equivalent to Linux's CLOCK\_MONOTONIC.

This timestamp should only be used to measure a duration by comparing it against another timestamp on the same device. Values returned by this method do not have a defined correspondence to wall clock times; the zero value is typically whenever the device last booted. Use currentTimeMillis if you want to know what time it is.

Возвращает текущую метку времени наиболее точного таймера, доступного в локальной системе, в наносекундах. Эквивалентно Linux CLOCK\_MONOTONIC.

\*\*\*\*\*

Эта временная метка должна использоваться только для измерения продолжительности путем сравнения ее с другой временной меткой на том же устройстве. Значения, возвращаемые этим методом, не имеют определенного соответствия временам настенных часов; нулевое значение, как правило, всегда, когда устройство загружалось в последний раз. Используйте currentTimeMillis, если вы хотите знать, который час.

##### BigD.time <result\_svar>

Returns the current time in milliseconds since January 1, 1970 00:00:00.0 UTC.

This method always returns UTC times, regardless of the system's time zone. This isoften called "Unix time" or "epoch time". ~~Use a java.text.DateFormat instance to format this timefor display to ahuman.~~

This method shouldn't be used for measuring timeouts or other elapsed time measurements, as changing the system time can affect the results. Use nanoTime for that.

\*\*\*\*\*

Возвращает текущее время в миллисекундах с 1 января 1970 года 00: 00: 00.0 UTC.

Этот метод всегда возвращает время UTC, независимо от часового пояса системы. Это часто называют «временем Unix» или «временем эпохи». Используйте экземпляр java.text.DateFormat, чтобы отформатировать это время для отображения в ahuman.

Этот метод не следует использовать для измерения времени ожидания или других измерений истекшего времени, так как изменение системного времени может повлиять на результаты. Используйте nanoTime для этого.

See also TIME()

##### BigD.toEngineering <result\_svar>, <first\_sexp>

Returns a string representation of <first\_sexp> BigDecimal. This representation always prints all significant digits of this value.

If the scale is negative or if scale - precision ≥ 6 then engineering notation is used. Engineering notation is similar to the scientific notation except that the exponent is made to be a multiple of 3 such that the integer part is ≥ 1 and < 1000.

\*\*\*\*\*

Возвращает строковое представление <first\_sexp> BigDecimal. Это представление всегда печатает все значащие цифры этого значения.

Если шкала отрицательная или если шкала с точностью ≥ 6, то используются инженерные обозначения. Инженерные обозначения аналогичны научным обозначениям, за исключением того, что показатель степени кратен 3, так что целая часть равна ≥ 1 и <1000.

##### BigD.toSientific <result\_svar>, <first\_sexp>

Returns a canonical string representation of <first\_sexp> BigDecimal. If necessary, scientific notation is used. This representation always prints all significant digits of this value.

If the scale is negative or if scale - precision ≥ 6 then scientific notation is used.

\*\*\*\*\*

Возвращает каноническое строковое представление <first\_sexp> BigDecimal. При необходимости используется научная запись. Это представление всегда печатает все значащие цифры этого значения.

Если шкала отрицательная или если шкала с точностью ≥ 6, то используются научные обозначения.

##### BigD.ulp <result\_svar>, <first\_sexp>

Returns the unit in the last place (ULP) of <first\_sexp> BigDecimal instance. An ULP is the distance to the nearest big decimal with the same precision.

The amount of a rounding error in the evaluation of a floating-point operation is often expressed in ULPs. An error of 1 ULP is often seen as a tolerable error.

For class BigDecimal, the ULP of a number is simply 10-scale. For example, BigD.ulp “123”, r$ returns “1”

BigD.ulp “1.23”, r$ returns “0.01”

\*\*\*\*\*

Возвращает единицу в последнем месте (ULP) экземпляра <first\_sexp> BigDecimal. ULP - это расстояние до ближайшего большого десятичного знака с той же точностью.

Величина ошибки округления при оценке операции с плавающей запятой часто выражается в ULP. Ошибка в 1 ULP часто рассматривается как допустимая ошибка.

Для класса BigDecimal ULP числа просто 10-масштабный. Например, BigD.ulp «123», r $ возвращает «1»

BigD.ulp «1.23», r $ возвращает «0.01»

### SHELL Command

##### Shell <result\_svar>, <command\_sexp>

Opens a **SHELL** to execute system commands <command\_sexp>. The working directory is in **opposite** to **System**.open set to "**root**". The command is waiting for a result

<result\_svar>.

\*\*\*\*\*

Открывает ОБОЛОЧКУ для выполнения системных команд <command\_sexp>. Рабочий каталог находится напротив System.open, установленного в «root». Команда ждет результата

<Result\_svar>.

Example:

FILE.ROOT path$

d$ = "cat " + path$ + "/" + "htmldemo1.html" SHELL r$, d$

PRINT r$

### TCP/IP Sockets

TCP/IP Sockets provide for the transfer of information from one point on the Internet to another. There are two genders of TCP/IP Sockets: Servers and Clients. Clients must talk to Servers. Servers must talk to Clients. Clients cannot talk to Clients. Servers cannot talk to Servers.

Every Client and Server pair have an agreed-upon protocol. This protocol determines who speaks first and the meaning and sequence of the messages that flow between them.

Most people who use a TCP/IP Socket will use a Client Socket to exchange messages with an existing Server with a predefined protocol. One simple example of this is the Sample Program file, **f31\_socket\_time.bas**. This program uses a TCP/IP client socket to get the current time from one of the many time servers in the USA.

ATCP/IPServercanbesetupinBASIC!;however,therearedifficulties.Thecapabilities of individual wireless networks vary. Some wireless networks allow servers. Most do not. Servers can usually be run on WiFi or Ethernet Local Area Networks(LAN).

If you want to set up a Server, the way most likely to work is to establish the Server inside a LAN. You will need to provide Port tunneling (forwarding) from the LAN’s external Internal IP to the device’s LAN IP. You must to be able to program (setup) the LAN router in order to do this.

Clients, whether running inside the Server’s LAN or from the Internet, should connectto the LAN’s external IP address using the pre-established, tunneled Port. This external or WAN IP can be foundusing:

Graburl ip$, "[http://icanhazip.com](http://icanhazip.com/)"

This is not the same IP that would be obtained by executing **Socket.myIP** on the server device.

Note: The specified IPs do not have to be in the numeric form. They can be in the name form.

The Sample Program, **f32\_tcp\_ip\_sockets.bas**, demonstrates the socket commands for a Server working in conjunction with a Client. You will need two Android devices to run this program.

On Android devices the default transfer character set is the character set of the filesystem here UTF-8.

If you need to transfer characters fom 0 to 255 maybe for binary data, you can choose the "\_ISO-8859-1" character set instead of the default "\_UTF-8".

But the Socket.Server.write.file, Socket.Server.read.file, Socket.Client.write.file or Socket.Client.read.file commands in conjunction with "\_ISO-8859-1" do not work as expected, because sending and receiving of Chr$(65535) is not possible.

\*\*\*\*\*

Сокеты TCP / IP обеспечивают передачу информации из одной точки Интернета в другую. Существует два вида сокетов TCP / IP: серверы и клиенты. Клиенты должны общаться с серверами. Серверы должны общаться с клиентами. Клиенты не могут общаться с клиентами. Серверы не могут общаться с серверами.

Каждая пара Клиент и Сервер имеет согласованный протокол. Этот протокол определяет, кто говорит первым, а также значение и последовательность сообщений, которые передаются между ними.

Большинство людей, которые используют сокет TCP / IP, будут использовать клиентский сокет для обмена сообщениями с существующим сервером с предопределенным протоколом. Одним из простых примеров этого является файл примера программы, f31\_socket\_time.bas. Эта программа использует сокет клиента TCP / IP для получения текущего времени от одного из многих серверов времени в США.

ATCP / IP-сервер может быть установлен в ОСНОВНОМ!, Однако есть трудности. Возможности отдельных беспроводных сетей различаются. Некоторые беспроводные сети позволяют серверам. Большинство нет. Серверы обычно могут работать в локальных сетях WiFi или Ethernet.

Если вы хотите настроить сервер, наиболее вероятный способ - это установить сервер внутри локальной сети. Вам нужно будет обеспечить туннелирование портов (переадресацию) с внешнего внутреннего IP-адреса локальной сети на IP-адрес устройства. Вы должны быть в состоянии запрограммировать (настроить) маршрутизатор локальной сети для того, чтобы сделать это.

Клиенты, независимо от того, работают ли они в локальной сети сервера или из Интернета, должны подключаться к внешнему IP-адресу локальной сети через предварительно установленный туннельный порт. Этот внешний или WAN IP можно найти с помощью:

Graburl ip $, "http://icanhazip.com"

Это не тот IP, который был бы получен при выполнении Socket.myIP на устройстве сервера.

Примечание. Указанные IP-адреса не обязательно должны быть в числовой форме. Они могут быть в форме имени.

Пример программы, f32\_tcp\_ip\_sockets.bas, демонстрирует команды сокета для сервера, работающего совместно с клиентом. Для запуска этой программы вам понадобятся два устройства Android.

На устройствах Android стандартным набором символов передачи является набор символов файловой системы UTF-8.

Если вам нужно передать символы от 0 до 255, возможно, для двоичных данных, вы можете выбрать набор символов «\_ISO-8859-1» вместо стандартного «\_UTF-8».

Но команды Socket.Server.write.file, Socket.Server.read.file, Socket.Client.write.file или Socket.Client.read.file в сочетании с "\_ISO-8859-1" не работают должным образом, потому что отправка и получение Chr $ (65535) невозможно.

### TCP/IP Client Socket Commands

##### Socket.client.connect <server\_sexp>, <port\_nexp> {{ , <wait\_lexp> },

**<char\_set\_sexp>}**

Create a Client TCP/IP socket and attempt to connect to the Server whose Host Name or IP Address is specified by the Server string expression using the Port specified by Port numeric expression.

The optional "wait" parameter determines if this command waits until a connection is made with the Server. If the parameter is absent or true (non-zero), the command will not return until the connection has been made or an error is detected. If the Server does not respond, the command should time out after a couple of minutes, but this is not certain.

If the parameter is false (zero), the command completes immediately. Use **Socket.client.status** to determine when the connection is made. If you monitor the socket status, you can set your own time-out policy. You must use the **Socket.client.close** command to stop a connection attempt that has not completed.

Using the optional <charset\_sexp> you can choose between following character sets: "\_ISO-8859-1" and “\_UTF-8”(default).

\*\*\*\*\*

Создайте клиентский сокет TCP / IP и попытайтесь подключиться к серверу, имя хоста или IP-адрес которого указан в строковом выражении сервера, используя порт, указанный числовым выражением порта.

Необязательный параметр «wait» определяет, будет ли эта команда ожидать установления соединения с сервером. Если параметр отсутствует или имеет значение true (отличное от нуля), команда не вернется, пока не будет установлено соединение или не будет обнаружена ошибка. Если сервер не отвечает, команда должна истечь через несколько минут, но это не точно.

Если параметр имеет значение false (ноль), команда завершается немедленно. Используйте Socket.client.status, чтобы определить, когда установлено соединение. Если вы отслеживаете состояние сокета, вы можете установить собственную политику тайм-аута. Вы должны использовать команду Socket.client.close, чтобы остановить попытку подключения, которая еще не завершена.

Используя опциональный <charset\_sexp>, вы можете выбирать между следующими наборами символов: «\_ISO-8859-1» и «\_UTF-8» (по умолчанию).

##### Socket.client.read.byte <svar>

Read a byte from the previously-connected Server and place the byte into the string variable. To avoid an infinite delay waiting for the Server to send a line, the **Socket.client.read.ready** command can be repeatedly executed with timeouts.

What About Reading Unknown Number of Bytes?

The best answer is that your application either needs to know beforehand how many bytes to expect, or the "application protocol" needs to somehow tell it how many bytes to

expect ... or when all bytes have been sent. Possible approaches are:

* + The application protocol uses fixed messagesizes
  + The application protocol message sizes are specified in messageheaders
  + The application protocol uses end-of-messagemarkers
  + The application protocol is not message based, and the other end closesthe connection to say "that's theend".

\*\*\*\*\*

Считайте байт с ранее подключенного Сервера и поместите его в строковую переменную. Чтобы избежать бесконечной задержки, ожидающей отправки сервером строки, команда Socket.client.read.ready может многократно выполняться с таймаутами.

Как насчет чтения неизвестного количества байтов?

Лучший ответ заключается в том, что ваше приложение должно либо заранее знать, сколько байтов ожидать, либо «протокол приложения» должен каким-то образом указать, сколько байтов следует ожидать.

ожидать ... или когда все байты были отправлены. Возможные подходы:

Протокол приложения использует фиксированные размеры сообщений

Размеры сообщений протокола приложения указаны в заголовках сообщений

Протокол приложения использует маркеры конца сообщения

Протокол приложения не основан на сообщениях, а другой конец - соединение, чтобы сказать «это конец».

line$ = "" DO

SOCKET.CLIENT.READ.BYTE mByte$

line$ = line$ + mByte$

UNTIL mByte$ = CHR$(10) | mByte$ = CHR$(13) line$ = REPLACE$(line$, CHR$(10), "")

line$ = REPLACE$(line$, CHR$(13), "") Is the same as SOCKET.CLIENT.READ.LINE line$

byteAsNumber = 0 ... 65535 out$ = CHR$(byteAsNumber )

SOCKET.SERVER.WRITE.BYTES out$

SOCKET.CLIENT.READ.BYTE in$

byteAsNumber = UCODE(in$)

##### Socket.client.read.line <line\_svar>

Read a line from the previously-connected Server and place the line into the line string variable. The command does not return until the Server sends a line. To avoid an infinite

delay waiting for the Server to send a line, the **Socket.client.read.ready** command can be repeatedly executed with timeouts.

Note, the end of the line is detected if a CR or LF is receiving.

\*\*\*\*\*

Считайте строку с ранее подключенного Сервера и поместите строку в строковую переменную строки. Команда не возвращается, пока Сервер не отправит строку. Чтобы избежать бесконечности

задерживая ожидание отправки сервером строки, команда Socket.client.read.ready может быть выполнена повторно с таймаутами.

Обратите внимание, что конец строки обнаруживается, если CR или LF получает.

### TCP/IP Server Socket Commands

##### Socket.server.create <port\_nexp>{, <char\_set\_sexp>}

Establish a Server that will listen to the Port specified by the numeric expression,

<port\_nexp>.

Using the optional <charset\_sexp> you can choose between following character sets: "\_ISO-8859-1" and “\_UTF-8”(default).

\*\*\*\*\*

Установите сервер, который будет прослушивать порт, указанный числовым выражением,

<Port\_nexp>.

Используя опциональный <charset\_sexp>, вы можете выбирать между следующими наборами символов: «\_ISO-8859-1» и «\_UTF-8» (по умолчанию).

##### Socket.server.read.byte <svar>

Read a byte sent from the previously-connected Client and place the byte into the string variable <svar>. To avoid an infinite delay waiting for the Client to send a line, the **Socket.server.read.ready** command can be repeatedly executed with timeouts.

See alsoSocket.client.read.byte.

\*\*\*\*\*

Считайте байт, отправленный ранее подключенным Клиентом, и поместите этот байт в строковую переменную <svar>. Чтобы избежать бесконечной задержки ожидания клиентом отправки строки, команда Socket.server.read.ready может многократно выполняться с таймаутами.

Смотрите также Socket.client.read.byte.

##### Socket.server.read.line<svar>

Read a line sent from the previously-connected Client and place the line into the string variable <svar>. The command does not return until the Client sends a line. To avoid an infinite delay waiting for the Client to send a line, the **Socket.server.read.ready** command can be repeatedly executed with timeouts.

Note, the end of the line is detected if a CR or LF is receiving.

\*\*\*\*\*

Прочитайте строку, отправленную с ранее подключенного Клиента, и поместите строку в строковую переменную <svar>. Команда не возвращается, пока Клиент не отправит строку. Чтобы избежать бесконечной задержки ожидания клиентом отправки строки, команда Socket.server.read.ready может многократно выполняться с таймаутами.

Обратите внимание, что конец строки обнаруживается, если CR или LF получает.

### UDP Socket Commands

User Datagram Protocol is a simpler message-based connectionless protocol. Connectionless protocols do not set up a dedicated end-to-end connection.

Communication is achieved by transmitting information in one direction from source to destination without verifying the readiness or state of the receiver.

**Unreliable** – When an UDP message is sent, it cannot be known if it will reach its destination; it could get lost along the way. There is no concept ofacknowledgment, retransmission, ortimeout.

**Not ordered** – If two messages are sent to the same recipient, the order in whichthey arrive cannot bepredicted.

**Lightweight** – There is no ordering of messages, no tracking connections, etc. It is a small transport layer designed on top of IP.

**Datagrams** – Packets are sent individually and are checked for integrity only if they arrive. Packets have definite boundaries which are honored upon receipt, meaning a read operation at the receiver socket will yield an entire message as it was originally sent.

**No congestion control** – UDP itself does not avoid congestion. Congestion control measures must be implemented at the application level.

**Broadcasts** - being connectionless, UDP can broadcast - sent packets can be addressed to be receivable by all devices on the subnet.

Source: Wikipedia

##### If you want a read and write communication (IoT), make sure that there is a pause of maybe 500ms for interaction.

##### \*\*\*\*\*

##### Протокол пользовательских дейтаграмм является более простым протоколом без установления соединения на основе сообщений. Протоколы без установления соединения не устанавливают выделенное сквозное соединение.

##### Связь достигается путем передачи информации в одном направлении от источника к месту назначения без проверки готовности или состояния приемника.

##### Ненадежный - при отправке UDP-сообщения неизвестно, достигнет ли оно своего пункта назначения; это может потеряться по пути. Не существует понятия подтверждения, повторной передачи или тайм-аута.

##### Не заказано - если два сообщения отправляются одному и тому же получателю, порядок их поступления не может быть предсказан.

##### Легкий - нет упорядочения сообщений, отслеживания соединений и т. Д. Это небольшой транспортный уровень, разработанный поверх IP.

##### Дейтаграммы - пакеты отправляются индивидуально и проверяются на целостность только в случае их поступления. Пакеты имеют определенные границы, которые учитываются при получении, что означает, что операция чтения в сокете-получателе приведет к тому, что сообщение будет полностью отправлено.

##### Отсутствие контроля перегрузки - сам UDP не предотвращает перегрузку. Меры контроля заторов должны быть реализованы на уровне приложений.

##### Широковещательные рассылки - при отсутствии соединения UDP может рассылать - отправленные пакеты могут быть адресованы для приема всеми устройствами в подсети.

##### Источник: Википедия

##### Если вы хотите обмен данными для чтения и записи (IoT), убедитесь, что для взаимодействия существует пауза около 500 мс.

**UDP.read <result\_svar>, <port\_nexp>, <wait\_nexp> {, <char\_set\_sexp>}**

Listens for an UDP message on port <port\_nexp> and read a string if any datagram message arrived.The command waits <wait\_nexp> milliseconds. If <wait\_nexp> is 0 it waits infinite. Using this command in a loop with 1200 ms for the first try is recommended. The character set can be specified by <char\_set\_sexp>. Use "\_UTF-8" (default) for text and "\_ISO-8859-1" for binary data. An input until 65527 bytes is accepted.

\*\*\*\*\*

Прослушивает UDP-сообщение через порт <port\_nexp> и читает строку, если поступило какое-либо сообщение дейтаграммы. Команда ожидает <wait\_nexp> миллисекунды. Если <wait\_nexp> равен 0, он ждет бесконечно. Рекомендуется использовать эту команду в цикле с 1200 мс для первой попытки. Набор символов может быть указан <char\_set\_sexp>. Используйте «\_UTF-8» (по умолчанию) для текста и «\_ISO-8859-1» для двоичных данных. Вход до 65527 байт принят.

Example:

DO

DO

UDP.READ r$, 12345, 1200, "\_ISO-8859-1" UNTIL r$ <> ""

PRINT r$ UNTIL 0

##### UDP.write <message\_svar>, <ip\_adress\_sexp>, <port\_nexp> {, <char\_set\_sexp>}

Writes a string <message\_svar> as an UDP datagram message to a local ip adress

<ip\_adress\_sexp> on port <port\_nexp>. The character set can be specified by

<char\_set\_sexp>. Use "\_UTF-8" (default) for text and "\_ISO-8859-1" for binary data.An output until 65507 bytes is accepted.

\*\*\*\*\*

Записывает строку <message\_svar> как сообщение дейтаграммы UDP на локальный IP-адрес

<ip\_adress\_sexp> на порту <port\_nexp>. Набор символов может быть указан

<Char\_set\_sexp>. Используйте «\_UTF-8» (по умолчанию) для текста и «\_ISO-8859-1» для двоичных данных. Вывод до 65507 байт.

Example:

m$= "My message! "

FOR i = 1 TO 100

mm$ = m$ + STR$(i)

UDP.WRITE mm$, "192.168.1.12", 12345, "\_ISO-8859-1"

PAUSE 50 NEXT

**Input {<prompt\_sexp>}, <result\_var>{, {<default\_exp>}{, <canceled\_nvar>}}** Generates a dialog box with an input area and an **OK** button. When the user taps the button, the value in the input area is written to the variable <result\_var>.

The <prompt\_sexp> will become the dialog box title. If the prompt expression is empty ("") or omitted, the dialog box will be drawn without a title area.

If the return variable <result\_var> is numeric, the input must be numeric, so the only key taps that will be accepted are 0-9, "+", "-" and ".". If <result\_var> is a string variable, the input may be any string.

If a <default\_exp> is given then its value will be placed into the input area of the dialog box. The default expression type must match the <result\_var> type.

The variable <canceled\_nvar> controls what happens if the user cancels the dialog, either by tapping the BACK key or by touching anywhere outside of the dialog box.

If you provide a <canceled\_nvar>, its value is set to **false** (0) if the user taps the **OK**

button, and **true** (1) if the users cancels the dialog.

If you do not provide a <canceled\_nvar>, a canceled dialog is reported as an error. Unless there is an "OnError:" the user will see the messages:

Input dialog cancelled Execution halted

If there is an "OnError:" label, execution will resume at the statement following the label. The <result\_var> parameter is required. All others are optional. These are all valid:

INPUT "prompt", result$, "default", isCanceled INPUT , result$, "default"

INPUT "prompt", result$, , isCanceled INPUT "prompt", result$

INPUT , result$

Note the use of commas as parameter placeholders (see Optional Parameters).

Note also, that in some Android versions in Graphic Mode the width of the box is shrinking according to the text size. Some spaces at the end of the prompt will help if needed.

\*\*\*\*\*

диалоговое окно с областью ввода и кнопкой ОК. Когда пользователь нажимает кнопку, значение в области ввода записывается в переменную <result\_var>.

<Prompt\_sexp> станет заголовком диалогового окна. Если выражение подсказки пустое ("") или опущено, диалоговое окно будет отображаться без области заголовка.

Если возвращаемая переменная <result\_var> является числовой, то ввод должен быть числовым, поэтому будут приниматься только нажатия клавиш 0-9, "+", "-" и ".". Если <result\_var> является строковой переменной, входные данные могут быть любой строкой.

Если задано <default\_exp>, его значение будет помещено в область ввода диалогового окна. Тип выражения по умолчанию должен соответствовать типу <result\_var>.

Переменная <cancelled\_nvar> управляет тем, что происходит, если пользователь отменяет диалоговое окно, либо нажимая клавишу BACK, либо касаясь любого места за пределами диалогового окна.

Если вы предоставляете <cancelled\_nvar>, его значение устанавливается в false (0), если пользователь нажимает кнопку OK.

Кнопка и true (1), если пользователь отменяет диалог.

Если вы не предоставите <cancelled\_nvar>, отмененное диалоговое окно будет отображено как ошибка. Если нет «OnError:», пользователь увидит сообщения:

Диалог ввода отменен. Выполнение остановлено.

Если есть метка «OnError:», выполнение возобновится с оператора, следующего за меткой. Параметр <result\_var> обязателен. Все остальные являются необязательными. Все они действительны:

INPUT "приглашение", результат $, "по умолчанию", isCanceled INPUT, результат $, "по умолчанию"

INPUT "подсказка", результат $, isCanceled INPUT "подсказка", результат $

INPUT, результат $

Обратите внимание на использование запятых в качестве заполнителей параметров (см. Дополнительные параметры).

Также обратите внимание, что в некоторых версиях Android в графическом режиме ширина поля уменьшается в зависимости от размера текста. Некоторые пробелы в конце подсказки помогут при необходимости.

##### Dialog.message {<title\_sexp>}, {<message\_sexp>}, <sel\_nvar> {, <button1\_sexp>{,

**<button2\_sexp>{, <button3\_sexp>}}}**

Generates a dialog box with a title, a message, and up to three buttons. When the user taps a button, the number of the selected button is returned in <sel\_nvar>. If the user taps the screen outside of the message dialog or presses the BACK key, then the returned value is 0.

The string <title\_sexp> becomes the title of the dialog box. The string <message\_sexp> is displayed in the body of the dialog, above the buttons. The strings <button1\_sexp>,

<button2\_sexp>, and <button3\_sexp> provide the labels on the buttons.

You may have 0, 1, 2, or 3 buttons. On most devices, the buttons are numbered from right- to-left, because Android style guides recommend the positive action on the right and the negative action on the left. Some devices differ. On compliant devices, tapping the right- most button returns 1.

All of the parameters except the selection index variable <sel\_nvar> are optional. If any parameter is omitted, the corresponding part of the message dialog is not displayed. Use commas to indicate omitted parameters (see Optional Parameters).

\*\*\*\*\*

Создает диалоговое окно с заголовком, сообщением и до трех кнопок. Когда пользователь нажимает кнопку, номер выбранной кнопки возвращается в <sel\_nvar>. Если пользователь нажимает на экран за пределами диалогового окна сообщения или нажимает клавишу BACK, возвращаемое значение равно 0.

Строка <title\_sexp> становится заголовком диалогового окна. Строка <message\_sexp> отображается в теле диалога над кнопками. Строки <button1\_sexp>,

<button2\_sexp> и <button3\_sexp> предоставляют метки на кнопках.

У вас может быть 0, 1, 2 или 3 кнопки. На большинстве устройств кнопки нумеруются справа налево, потому что руководства по стилю Android рекомендуют положительное действие справа и отрицательное действие слева. Некоторые устройства отличаются. На совместимых устройствах нажатие самой правой кнопки возвращает 1.

Все параметры, кроме индексной переменной выбора <sel\_nvar>, являются необязательными. Если какой-либо параметр пропущен, соответствующая часть диалогового окна сообщения не отображается. Используйте запятые для обозначения пропущенных параметров (см. Дополнительные параметры).

Examples:

Dialog.Message "Hey, you!", "Is this ok?", ok, "Sure thing!", "Don’t care", "No way!" Dialog.Message "Continue?", , go, "YES", "NO"

Dialog.Message , "Continue?", go, "YES", "NO" Dialog.Message , , b

The first command displays a full dialog with a title, a message, and three buttons. The second command displays a box with a title and two buttons – note that the **YES** button will be on the right and the **NO** button on the left. The third displays the same

information, but it looks a little different because the text is displayed as the message and not as the title. Note the commas.

The fourth command displays nothing at all. The screen dims and your program waits for a tap or the BACK key with no feedback to tell the user what to do.

Is a negative <sel\_nvar> at command start given, the message dialog will be finished in

<sel\_nvar> \* -1 milliseconds. In this case <sel\_nvar> returns 0.0.

Note, that in some Android versions in Graphic Mode the width of the box is shrinking according to the text size. Some spaces at the end of the title or the message will help if needed.

\*\*\*\*\*

Первая команда отображает полный диалог с заголовком, сообщением и тремя кнопками. Вторая команда отображает поле с заголовком и двумя кнопками - обратите внимание, что кнопка YES будет справа, а кнопка NO - слева. Третий отображает то же самое

информация, но выглядит немного иначе, потому что текст отображается как сообщение, а не как заголовок. Обратите внимание на запятые.

Четвертая команда вообще ничего не отображает. Экран тускнеет, и ваша программа ожидает нажатия или клавиши BACK без обратной связи, чтобы сообщить пользователю, что делать.

Если задано отрицательное значение <sel\_nvar> при запуске команды, диалог сообщения будет завершен через

<sel\_nvar> \* -1 миллисекунды. В этом случае <sel\_nvar> возвращает 0.0.

Обратите внимание, что в некоторых версиях Android в графическом режиме ширина поля уменьшается в зависимости от размера текста. Некоторые пробелы в конце заголовка или сообщения помогут при необходимости.

##### Dialog.select <sel\_nvar>, <Array$[]>|<list\_nexp> {,<title\_sexp>}

Generates a dialog box with a list of choices for the user. When the user taps a list item, the index of the selected line is returned in the <sel\_nvar>. If the user taps the screen outside of the selection dialog or presses the BACK key, then the returned value is 0.

<Array$[]> is a string array that holds the list of items to be selected. The array is specified without an index but must have been previously dimensioned or loaded via Array.load.

As an alternative to an array, a string-type list may be specified in the <list\_nexp>.

The <title\_sexp> is an optional string expression that will be displayed at the top of the selection dialog. If the parameter is not present, or the expression evaluates to an empty string (""), the dialog box will be displayed with no title.

This command also accepts optional <message\_sexp> and <press\_lvar> parameters like those described in the **Select** command, but they should not be used. The

<message\_sexp> is ignored and the <press\_lvar> will always be set to 0.

Is a negative <sel\_nvar> at command start given, the Select dialog will be finished in

<sel\_nvar> \* -1 milliseconds. In this case <sel\_nvar> returns 0.0.

\*\*\*\*\*

Создает диалоговое окно со списком вариантов для пользователя. Когда пользователь касается элемента списка, индекс выбранной строки возвращается в <sel\_nvar>. Если пользователь нажимает на экран за пределами диалогового окна выбора или нажимает клавишу BACK, возвращаемое значение равно 0.

<Array $ []> - это строковый массив, содержащий список элементов, которые нужно выбрать. Массив указывается без индекса, но должен быть предварительно измерен или загружен через Array.load.

В качестве альтернативы массиву в <list\_nexp> может быть указан список строкового типа.

<Title\_sexp> - это необязательное строковое выражение, которое будет отображаться в верхней части диалогового окна выбора. Если параметр отсутствует или выражение выражается пустой строкой (""), диалоговое окно будет отображаться без заголовка.

Эта команда также принимает необязательные параметры <message\_sexp> и <press\_lvar>, подобные тем, которые описаны в команде Select, но их не следует использовать.

<message\_sexp> игнорируется, а <press\_lvar> всегда будет установлен в 0.

Если задано отрицательное значение <sel\_nvar> при запуске команды, диалог выбора будет завершен через

<sel\_nvar> \* -1 миллисекунды. В этом случае <sel\_nvar> возвращает 0.0.

##### Select <sel\_nvar>, <Array$[]>|<list\_nexp> {,<title\_sexp> {, <message\_sexp> } }

**{{,<press\_nvar> },<layout\_bundle\_nexp> }**

The Select command generates a new screen with a list of choices for the user. When the user taps a screen line, the index of the selected line is returned in the <sel\_nvar>. If the user presses the BACK key, then the returned value is 0.

<Array$[]> is a string array that holds the list of items to be selected. The array is specified without an index but must have been previously dimensioned, loaded via Array.load, or created by another command.

As an alternative to an array, a string-type list may be specified in the <list\_nexp>.

The <title\_sexp> is an optional string expression that is placed into the title bar at the top of the selection screen. If the parameter is not present, the screen displays a default title.If the expression evaluates to an empty string ("") the title isblank.

The <message\_sexp> is an optional string expression that is displayed in a short Popup message. If the message is an empty string ("") there is no Popup. If the parameter is absent, the <title\_sexp> string is used instead, but if the <title\_sexp> is also missing or empty, there is no Popup.

If the optional <press\_nvar> is present, the type of user touch a short tap (0), a long press

(1) or a double tap (2) is returned in the <press\_nvar>. The delay for detecting the double tap corresponds to the default Android system settings.~~Its value will be 0 (false) if the touch was a short tap. Its value will be 1 (true) if the touch was a long press.~~

Use commas to indicate omitted optional parameters (see Optional Parameters).

Is a negative <sel\_nvar> at command start given, the Select dialog will be finished in

<sel\_nvar> \* -1 milliseconds. In this case <sel\_nvar> returns 0.0.

The optional layout bundle <layout\_bundle\_nexp> controls the Select output layout:

Команда Select создает новый экран со списком вариантов для пользователя. Когда пользователь нажимает на строку экрана, индекс выбранной строки возвращается в <sel\_nvar>. Если пользователь нажимает клавишу BACK, возвращаемое значение равно 0.

<Array $ []> - это строковый массив, содержащий список элементов, которые нужно выбрать. Массив указывается без индекса, но должен быть предварительно измерен, загружен через Array.load или создан другой командой.

В качестве альтернативы массиву в <list\_nexp> может быть указан список строкового типа.

<Title\_sexp> - это необязательное строковое выражение, которое помещается в строку заголовка в верхней части экрана выбора. Если параметр отсутствует, на экране отображается заголовок по умолчанию. Если выражение оценивается как пустая строка (""), заголовок является пустым.

<Message\_sexp> - это необязательное строковое выражение, которое отображается в коротком всплывающем сообщении. Если сообщение является пустой строкой (""), всплывающее окно отсутствует. Если параметр отсутствует, вместо него используется строка <title\_sexp>, но если <title\_sexp> также отсутствует или пуст, всплывающее окно отсутствует.

Если присутствует необязательный <press\_nvar>, тип пользователя касается короткого нажатия (0), длительного нажатия

(1) или двойное нажатие (2) возвращается в <press\_nvar>. Задержка для обнаружения двойного касания соответствует настройкам системы Android по умолчанию. Его значение будет 0 (ложно), если касание было коротким касанием. Его значение будет 1 (истина), если касание было долгим нажатием.

Используйте запятые для обозначения пропущенных необязательных параметров (см. Необязательные параметры).

Если задано отрицательное значение <sel\_nvar> при запуске команды, диалог выбора будет завершен через

<sel\_nvar> \* -1 миллисекунды. В этом случае <sel\_nvar> возвращает 0.0.

Необязательный пакет макета <layout\_bundle\_nexp> управляет макетом выбора вывода:

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** | **Description** |
| **\_TextSize** | numeric |  |
| **\_TextColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Note, \_TextFont or \_TextStyle is needed also! |
| \_**TextBackgroundColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Has to be "0,0,0,0" if you want a background color, wallpaper or bitmap  Note, \_TextFont or \_TextStyle is needed also! |
| **\_TextFont** | \_Default |  |
| \_Serif |  |
| \_Sans\_Serif |  |
| \_Monospace |  |
| **\_TextStyle** | \_Normal |  |

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** | **Description** |
|  | \_Bold |  |
| \_Bold\_Italic |  |
| \_Italic |  |
| **\_TextHtml** | 0 or 1 (numeric) | Returns displayable styled text from the provided HTML string. But not all tags are supported. Any <img> tags in the HTML will display an image. Absolute ("file://") and relative paths are allowed.  The image size has to be scaled before, because h= and w= are ignored. See\_HtmlBitmapScale.    Uses parts of TagSoup library to handle real HTML, including allof the brokenness found in thewild.  <a href="...">  <b>  <big>  <blockquote>  <br>  <cite>  <dfn>  <div align="...">? Use instead chr$(1564) [Arabic Letter] at line begin for align=’ right’  <em>  <font size="..." color="..." face="...">  <h1>, <h2>, <h3>, <h4>, <h5>,  <h6>  <i>  <img src="...">  <p>  <small>  <strike>? < A.7  <strong>  <sub>  <sup>  <tt>?  <u>  **Replace Space with &#160,**  **& with &amp,**  **< with &lt,**  **> with &gt, " with &quot if necessary.** |

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** | **Description** |
| **\_HtmlTextSelectable** | 0 or 1 (numeric) | Does only work in conjunction with  \_TextHtml, but the item selection works only with a **long** click. |
| **\_HtmlBitmapScale** | -1, 0, > 0 (numeric) | Does only work in conjunction with  \_TextHtml. Scales the included bitmaps in the following ways:  -1 No scaling,  0 (default) Only scaling proportional to the screen resolution  > 0 Proportional to the font size  If it is 1 the bitmap height is the same as the font size. |
| **\_DividerColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) |  |
| **\_DividerFilename** | bitmap file path |  |
| **\_DividerHeight** | numeric |  |
| **\_BackgroundWallpaper** | 0 or 1 (numeric) | ~~Min. Jelly Bean 4.1 (API 16)~~ |
| **\_BackgroundColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) |  |
| **\_BackgroundFilename** | bitmap file path | ~~Min. Jelly Bean 4.1 (API 16)~~ |
| **\_Orientation** | (numeric)  -1, 0, 1, 2 or 3 | The value sets the orientation of screen as follows:  -1 =Orientation  depends upon the sensors. 0 =Orientation  is forced to Landscape. 1 =Orientation  is forced toPortrait. 2 =Orientation  is forced to ReverseLandscape. 3 =Orientation  is forced to Reverse Portrait. |
| **\_SetSelection** | (numeric) | Sets a pre selected item. The item will not be selected but it will still be positioned appropriately. If the specified selection position is less  than 1, then the item at position 1 will be selected. |

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** | **Description** |
| **\_StackFromBottom** | 0 or 1 (numeric) | If 1 pin the view's content to the  bottom edge, 0 to pin the view's content to the top edge |
| **\_Subtitle** | String | Set the Action bar's subtitle. |
| **\_TitleShow** | 0 or 1 (numeric) | If 1 (default)  Show the Action bar if it is not currently showing. It will resize application content to fit the new space available.  If 0  Hide the Action bar if it is currently showing. It will resize application content to fit the new space available. |
| **\_TitleIcon** | Icon file path | Add a large icon to the notification content view.<http://romannurik.github.io/>  AndroidAssetStudio/index.html |
| **\_TitleHomeEnabled** | 0 or 1 (numeric) | Set whether to include the application home accordance in the Action bar. Home is presented as an activity icon.  Have to be 1 if you want to show the icon.  Have to be 0 if you want to hide the icon.  The default setting is API dependent. |
| **\_TitleBackground** | Background file path |  |

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** | **Description** |
| **\_TitleHtml** | 0 or 1 (numeric) | Returns displayable styled text from the provided HTML string. But not all tags are supported.    Uses parts of TagSoup library to handle real HTML, including all of the brokenness found in the wild.  <b>  <big>  <font size="..." color="..." face="...">  <h1>, <h2>, <h3>, <h4>, <h5>,  <h6>  <i>  <small>  <strike>? < A.7  <strong>  <sub>  <sup>  <tt>?  <u>  **Replace Space with &#160,**  **& with &amp,**  **< with &lt,**  **> with &gt, " with &quot if necessary.**  Usable for Title andSubtitle.  Keep in mind that theActionbar height will not beexpanded. |
| **\_StatusbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |
| **\_StatusbarLight** | 0 or 1 (numeric) | If 0 (default)  The Status bar background is dark. In this case the **bar content** will be **light**.  If 1  The Status bar background is light. In this case the **bar content** will be **dark**.  Min. Lollipop 5.0 (API 21) |

|  |  |  |
| --- | --- | --- |
| **Table of layout control options** | | |
| **Key** | **Value** | **Description** |
| **\_ShowNavigationbar** | 0, 1 or 2 (numeric) | If 1 (default)  The Navigation bar will be displayed.  If 2  The Navigation bar will be transparent displayed.  Min. Lollipop 5.0 (API 21)  If 0  The Navigation bar will be hidden to the background.  Min. Nougat 7.0 (API 24)  Will be switched to option 2 or 1 if the current API level is lower. |
| **\_NavigationbarColor** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |
| **\_NavigationbarLight** | 0 or 1 (numeric) | If 0 (default)  The Navigation bar background is dark.  In this case the **bar content** will be  **light**. If 1  The Navigation bar background is light.  In this case the **bar content** will be  **dark**.  Min. Lollipop 5.0 (API 21) |

Note, that you should scale the text size and the divider height in conjunction to the detected screen size.

If you want to change something in the layout bundle, it is sufficient to make the changes only in the bundle. Note that every change affects the whole thing.

\*\*\*\*\*

Обратите внимание, что вы должны масштабировать размер текста и высоту разделителя в соответствии с обнаруженным размером экрана.

Если вы хотите что-то изменить в пакете макетов, достаточно внести изменения только в пакет. Обратите внимание, что каждое изменение влияет на все это.

See also Console.layout, Console.title

Example:

ARRAY.LOAD months$[], "January", "February"~ "March", "April", "May", "June", "July", "August"~ "September", "October", "November", "December"

bundle.put layout,"\_TextSize", 30

bundle.put layout,"\_TextColor", "0,0,255"

bundle.put layout,"\_TextBackgroundColor", "250,250,250,250" bundle.put layout,"\_TextStyle", "\_Bold"

bundle.put layout,"\_SetSelection", 10

! bundle.put layout,"\_StackFromBottom", 1

! bundle.put layout,"\_DividerFilename", "cartman.png" bundle.put layout,"\_DividerColor", "255,0,255,0"

bundle.put layout,"\_DividerHeight", 5

! bundle.put layout,"\_BackgroundFilename", "cartman.png" tLong = 10

SELECT month, months$[], msg$, tLong ,layout PRINT months$[month]

##### Text.input <svar>{{, { <text\_sexp>} , <title\_sexp> }, <inputType\_nexp> }

This command is similar to "Input" except that it is used to input and/or edit a largequantity of text. It opens a new window with scroll bars and full text editing capabilities. You mayset the title of the new window with the optional <title\_sexp>parameter.

If the optional <text\_sexp> is present then that text is loaded into the text input windowfor editing. If <text\_sexp> is not present then the text.input text area will be empty.If

<title\_sexp> is needed but text.input text area is to be initially empty, use two commas to indicate the <sexp> specifies the title and not the initial text.

When done editing, tap the Finish ~~button~~item in the menu. The edited text is returned in

<svar>.

If you tap the BACK key or the Stop item in the menu then all text editing is discarded.

<svar> returns the original <sexp> text.

The optional parameter <inputType\_nexp> sets an input type. If it 1, text suggestion is switched off. Default is 0, text suggestion is switched on.

The following example grabs the Sample Program file, **f01\_commands.bas**, to string s$. It then sends s$ to text.input for editing. The result of the edit is returned in string r$. r$ is then printed to console.

\*\*\*\*\*

Эта команда похожа на «Ввод» за исключением того, что она используется для ввода и / или редактирования большого количества текста. Это открывает новое окно с полосами прокрутки и полными возможностями редактирования текста. Вы можете установить заголовок нового окна с помощью необязательного параметра <title\_sexp>.

Если присутствует необязательный <text\_sexp>, этот текст загружается в окно ввода текста для редактирования. Если <text\_sexp> нет, то текстовая область text.input будет пустой.

<title\_sexp> необходим, но текстовая область text.input должна быть изначально пустой, используйте две запятые, чтобы указать, что <sexp> указывает заголовок, а не исходный текст.

Когда вы закончите редактирование, нажмите кнопку Готово в меню. Отредактированный текст возвращается в

<SVAR>.

Если вы нажмете кнопку НАЗАД или пункт «Стоп» в меню, редактирование текста будет отменено.

<svar> возвращает исходный текст <sexp>.

Необязательный параметр <inputType\_nexp> устанавливает тип ввода. Если это 1, текстовое предложение выключено. По умолчанию 0, текстовое предложение включено.

В следующем примере файл примера программы f01\_commands.bas получает строку s $. Затем он отправляет s $ в text.input для редактирования. Результат редактирования возвращается в строке r $. Затем r $ выводится на консоль.

GRABFILE s$, "../source/ Sample\_Programs/f01\_commands.bas" TEXT.INPUT r$,s$

PRINT r$ END

##### Popup <message\_sexp> {{, <x\_nexp>}{, <y\_nexp>}{, <duration\_lexp>}}

Pops up a small message for a limited duration. The message is <message\_sexp>.

All of the parameters except the message are optional. If omitted, their default values are

0. Use commas to indicate omitted parameters (see Optional Parameters).

The simplest form of the **Popup** command, **Popup "Hello!"**, displays the message in the center of the screen for two seconds.

The default location for the Popup is the center of the screen. The optional <x\_nexp> and

<y\_nexp> parameters give a displacement from the center. The values may be negative. Select the duration of the Popup, either 2 seconds or 4 seconds, with the optional

<duration\_lexp> "long flag". If the flag is false (the expression evaluates to 0) the message is visible for 2 seconds. If the flag is true (non-zero) the message is visible for 4 seconds.If the flag is omitted the duration isshort.

The popup will be created outside from the current app in an Android messagequeue. These popup messages will be shown in any case step by step, also if the program is finished.

\*\*\*\*\*

Выскакивает небольшое сообщение в течение ограниченного времени. Это сообщение <message\_sexp>.

Все параметры, кроме сообщения, являются необязательными. Если опущено, их значения по умолчанию

0. Используйте запятые для обозначения пропущенных параметров (см. Дополнительные параметры).

Самая простая форма всплывающей команды, Popup «Hello!», Отображает сообщение в центре экрана в течение двух секунд.

Местом для всплывающего окна по умолчанию является центр экрана. Необязательный <x\_nexp> и

Параметры <y\_nexp> дают смещение от центра. Значения могут быть отрицательными. Выберите продолжительность всплывающего окна, 2 секунды или 4 секунды, с дополнительным

<duration\_lexp> "длинный флаг". Если флаг имеет значение false (выражение оценивается как 0), сообщение отображается в течение 2 секунд. Если флаг имеет значение true (отличное от нуля), сообщение отображается в течение 4 секунд. Если флаг не указан, длительность isshort.

Всплывающее окно будет создано вне текущего приложения в очереди сообщений Android. Эти всплывающие сообщения будут отображаться в любом случае шаг за шагом, даже если программа завершена.

### Scheduler Interrupt and Commands

You can set a scheduler that will interrupt the execution of your program at some set time or interval. When the scheduler expires, BASIC! jumps to the statements following the **OnSched:** label. When you have done whatever you need to do to handle this scheduler event, you use the **Sched.resume** command to resume execution of the program at the point where the scheduler interrupt occurred.

The scheduler cannot interrupt an executing command. When the scheduler expires, the current command is allowed to complete. Then the scheduler interrupt code after the **OnSched:** label executes. If the current command takes a long time to finish, it may appear that your scheduler is late.

The scheduler cannot interrupt another interrupt. If the scheduler expires while any interrupt event handler is running, the **OnSched:** interrupt will be delayed. If the scheduler expires while the **OnSched:** interrupt handler is running, the scheduler event will be lost. The **OnSched:** interrupt code must exit by running **Sched.resume**, or the scheduler interrupt can occur only once.

\*\*\*\*\*

Вы можете установить планировщик, который будет прерывать выполнение вашей программы через определенное время или интервал. Когда планировщик истекает, BASIC! переходит к операторам, следующим за меткой OnSched: Когда вы сделали все, что вам нужно сделать для обработки этого события планировщика, вы используете команду Sched.resume, чтобы возобновить выполнение программы в тот момент, когда произошло прерывание планировщика.

Планировщик не может прервать выполнение команды. Когда срок действия планировщика истекает, текущая команда может быть выполнена. Затем выполняется код прерывания планировщика после метки OnSched :. Если выполнение текущей команды занимает много времени, может показаться, что ваш планировщик опоздал.

Планировщик не может прерывать другое прерывание. Если срок действия планировщика истекает во время работы какого-либо обработчика события прерывания, прерывание OnSched: будет отложено. Если срок действия планировщика истекает во время работы обработчика прерываний OnSched:, событие планировщика будет потеряно. Код прерывания OnSched: должен завершиться с помощью Sched.resume, иначе прерывание планировщика может произойти только один раз.

##### Sched.set <firstInterrupt\_nexp>, <interval\_nexp> {,<date\_flag\_nexp>}

Sets a scheduler that will interrupt program execution after the in <firstInterrupt\_nexp> specified time or date in milliseconds. The parameter <interval\_nexp> sets a repeatedly program execution interrupt with the specified time interval in milliseconds. The optionally flag <date\_flag\_nexp> defines whether <firstInterrupt\_nexp> is a date driven time (>0) or a time distance (≤0). The date driven time is default. If <firstInterrupt\_nexp> and

<interval\_nexp> are equal and <date\_flag\_nexp> is 0 the functionality is the same as **Timer.set**. That is a good option, if you need **two timers**. The program must also contain an **OnSched:** label.

See also Timer.set.

t = TIME(2018, 1, 1, 0, 0, 1)

SCHED.SET t, 0 % Happy New Year 2018! One time interrupt.

! Is your device switched off at this date, the interrupt occur at the next program start.

t = TIME() + 4000 SCHED.SET t, 2000, 1

! First Interrupt at current time + 4 seconds and the interrupt repeats every 2 seconds. SCHED.SET 0, 2000, **0** % Interrupt immediately and repeats every 2 seconds.

SCHED.SET 2000, 2000, **0** % It is the same as TIMER.SET 2000

SCHED.SET 2000, 0, **0** % Interrupt one time in 2 seconds.

\*\*\*\*\*

Устанавливает планировщик, который будет прерывать выполнение программы после указанного в <firstInterrupt\_nexp> времени или даты в миллисекундах. Параметр <interval\_nexp> устанавливает многократное прерывание выполнения программы с указанным интервалом времени в миллисекундах. Опциональный флаг <date\_flag\_nexp> определяет, является ли <firstInterrupt\_nexp> датой времени (> 0) или расстоянием (≤0). Дата вождения время по умолчанию. Если <firstInterrupt\_nexp> и

<interval\_nexp> равны и <date\_flag\_nexp> равен 0, функциональность такая же, как Timer.set. Это хороший вариант, если вам нужны два таймера. Программа также должна содержать метку OnSched :.

Смотрите также Timer.set.

t = ВРЕМЯ (2018, 1, 1, 0, 0, 1)

SCHED.SET t, 0% С Новым 2018 годом! Одноразовое прерывание.

! Если ваше устройство выключено в эту дату, прерывание происходит при следующем запуске программы.

t = ВРЕМЯ () + 4000 SCHED.SET t, 2000, 1

! Первое прерывание в текущее время + 4 секунды, и прерывание повторяется каждые 2 секунды. SCHED.SET 0, 2000, 0% Немедленно прерывать и повторять каждые 2 секунды.

SCHED.SET 2000, 2000, 0% Это то же самое, что TIMER.SET 2000

SCHED.SET 2000, 0, 0% Прерывание один раз в 2 секунды.

##### OnSched:

Interrupt label for the scheduler interrupt. (See "Interrupt Labels".)

\*\*\*\*\*

Метка прерывания для прерывания планировщика. (См. «Метки прерываний».)

##### Sched.resume

Resumes execution at the point in the BASIC! program where the **OnSCHED:** interrupt occurred.

\*\*\*\*\*

Возобновляет выполнение в пункте BASIC! программа, в которой произошло прерывание OnSCHED:

##### Sched.clear

Clears the repeating scheduler. No further scheduler interrupts will occur.

\*\*\*\*\*

Очищает повторяющийся планировщик. Никаких дальнейших прерываний планировщика не произойдет.

### Sensor Commands

##### Sensors.list <sensor\_array$[]>{, <all\_nexp>}

Writes information about the sensors available on the Android device into the

<sensor\_array$[]> parameter. If the array exists, it is overwritten. Otherwise a new array is created. The result is always a one-dimensional array.

The array elements contain the names and types of the available sensors. For example, one element may be "Gyroscope, Type = 4".

Is <all\_nexp> = 1 and Android 5+ all available information will be presented. Default

<all\_nexp> is set to 0.

The following program snippet prints the elements of the sensor list.

\*\*\*\*\*

Записывает информацию о датчиках, доступных на устройстве Android, в

параметр <sensor\_array $ []>. Если массив существует, он перезаписывается. В противном случае создается новый массив. Результатом всегда является одномерный массив.

Элементы массива содержат имена и типы доступных датчиков. Например, одним элементом может быть «Гироскоп, Тип = 4».

Если <all\_nexp> = 1 и Android 5+, вся доступная информация будет представлена. По умолчанию

<all\_nexp> установлен в 0.

Следующий фрагмент программы печатает элементы списка датчиков.

SENSORS.LIST sensorarray$[] ARRAY.LENGTH size, sensorarray$[] FOR index = 1 TO size

PRINT sensorarray$[ index ] NEXT index

END

##### Sensors.read <sensor\_type\_nexp>, <p1\_nvar>, <p2\_nvar>, <p3\_nvar> {,

**<param\_array[]>}**

This command returns that latest values from the sensors specified by the "sensor\_type" parameters. The values are returned are placed into the p1, p2 and p3 parameters. The meaning of these parameters depends upon the sensor being read. Not all sensors return all three parameter values. In those cases, the unused parameter values will be set to zero. See [Android's Sensor Event](http://developer.android.com/reference/android/hardware/SensorEvent.html)web page for the meaning of these parameters.

The optional <param\_array[]> returns an array with three or more parameters. Sensortypes up to 30 and parameters up to 20 are supported.

\*\*\*\*\*

Эта команда возвращает последние значения от датчиков, указанных в параметрах «sensor\_type». Возвращаемые значения помещаются в параметры p1, p2 и p3. Значение этих параметров зависит от считываемого датчика. Не все датчики возвращают все три значения параметров. В этих случаях неиспользуемые значения параметров будут установлены на ноль. См. Страницу Android Eventweb для определения значения этих параметров.

Необязательный <param\_array []> возвращает массив с тремя или более параметрами. Поддерживаются сенсорные типы до 30 и параметры до 20.

### Special Floating Point Commands

##### Is\_NaN(<nexp>)

Returns 1.0 (true) if <nexp> is a NaN (Not a Number) Floating Point number else

0.0 if false.

\*\*\*\*\*

Возвращает 1.0 (true), если <nexp> - число с плавающей точкой NaN (не число), иначе

0.0 если ложно

**Is\_Infinite(<nexp>)**

Returns 1.0 (true) if <nexp> is an Infinit**y** Floating Point number else 0.0 if false. Note, that Is\_Infinit**e** ends with an **e**!

\*\*\*\*\*

Возвращает 1.0 (true), если <nexp> является числом с плавающей запятой бесконечности, иначе 0.0, если false. Обратите внимание, что Is\_Infinite заканчивается на e!

**Graphical Commands**

**Within(<sexp>, <arg1>, <arg2>, <arg3>, <arg4>...)**

Returns <> 0 (true) if the search for a graphical object described by <sexp> is successful. In other cases 0.0 false is returned.

As an example read for "\_point\_polygon": "**With** point **in** polygon?". If the point is directly onto a line, it is true also.

Is a positive number returned, the polygon was drawn clockwise.

Is a negative number returned, the polygon was drawn counterclockwise. If +1 or -1 is returned, the point is inside the polygon.

If +2 or 2 is returned, the point is inside an inner polygon of the polygon.

\*\*\*\*\*

Возвращает <> 0 (true), если поиск графического объекта, описанного <sexp>, выполнен успешно. В остальных случаях возвращается 0.0 false.

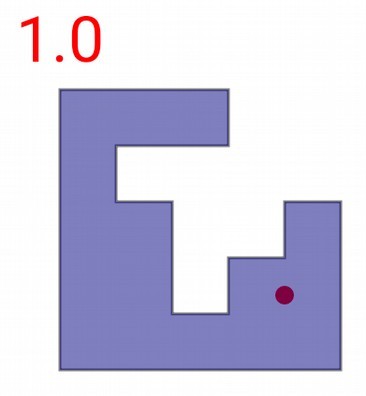
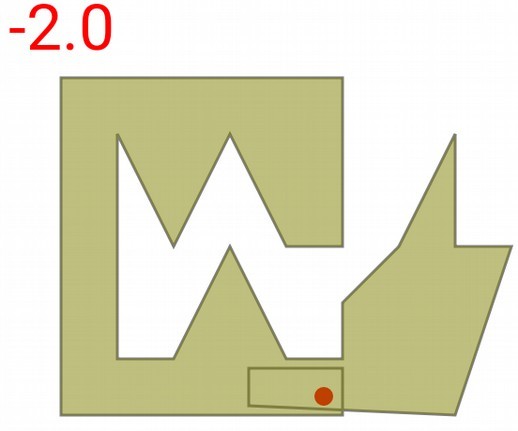
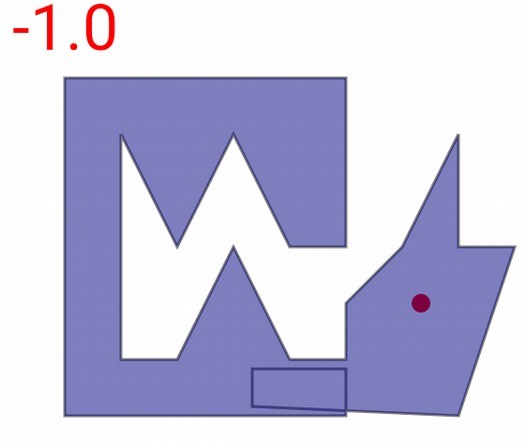
В качестве примера прочитайте «\_point\_polygon»: «С точкой в ​​многоугольнике?». Если точка находится прямо на линии, это также верно.

Возвращено положительное число, многоугольник нарисован по часовой стрелке.

Возвращено отрицательное число, многоугольник нарисован против часовой стрелки. Если +1 или -1 возвращается, точка находится внутри многоугольника.

Если возвращается +2 или 2, точка находится внутри внутреннего многоугольника.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| <sexp>/description | Argument number | Value type | Value | Default |
| **"\_point\_polygon"** |  |  |  |  |
| Point x | 1 | Double | Double |  |
| Point y | 2 | Double | Double |  |
| List of Polygon points as x values or x, y pairs | 3 | List numeric | Double |  |
| **{**List of Polygon points as y values**}** (Option) | 4 | List numeric | Double |  |
| For more information see <http://geomalgorithms.com/a03-_inclusion.html> | | | | |
|  |  |  |  |  |



**GR\_COLLISION(<object\_1\_nvar>, <object\_2\_nvar>{,<dist\_1\_nvar>,**

**<dist\_2\_nvar>})**

The variables <object\_1\_nvar> and <object\_2\_nvar> are the object pointers returned when the objects were created.

If the boundary boxes of the two objects overlap then the function will return true (not zero). If they do not overlap then the function will return false (zero).

Objects that may be tested for collision are: point, rectangle, bitmap, circle, arc, oval, and text. In the case of a circle, an arc, an oval, or text, the object’s rectangular boundary box is used for collision testing, not the actual drawn object. The arguments <dist\_1\_nvar> and <dist\_2\_nvar> allows to set back (-) or forth (+) the collision border. Take care that in a case of a backset your object is big enough. Example:

A good example is a circle that touches a rectangle. The circle has a diameter of 50 so a radius of 25.

With a circle backset of -25 (now a point) and a rectangle forthset of +25 the circle touches exactly the rectangle.

\*\*\*\*\*

Переменные <object\_1\_nvar> и <object\_2\_nvar> являются указателями объектов, возвращаемыми при создании объектов.

Если граничные блоки двух объектов перекрываются, то функция вернет true (не ноль). Если они не перекрываются, функция вернет false (ноль).

Объекты, которые могут быть проверены на столкновение: точка, прямоугольник, растровое изображение, круг, дуга, овал и текст. В случае круга, дуги, овала или текста прямоугольное граничное поле объекта используется для проверки столкновения, а не фактического нарисованного объекта. Аргументы <dist\_1\_nvar> и <dist\_2\_nvar> позволяют установить назад (-) или вперед (+) границу столкновения. Позаботьтесь, чтобы в случае сбоя ваш объект был достаточно большим. Пример:

Хорошим примером является круг, который касается прямоугольника. Круг имеет диаметр 50, а радиус 25.

При заданном смещении круга -25 (теперь это точка) и смещении прямоугольника +25 круг точно касается прямоугольника.

GR\_COLLISION(circle50, rectangle, -25, +25)

**Hide and Show Commands**

##### Gr.hide <object\_number\_nexp>{, <object\_number\_nexp> ...}

Hides the objects with the specified Object Numbers. If the Object is a Group, all of the Graphical Objects in the Group are hidden. This change will not be visible until the **Gr.render** command is called.

\*\*\*\*\*

Скрывает объекты с указанными номерами объектов. Если Объект является Группой, все Графические Объекты в Группе скрыты. Это изменение не будет видно, пока не будет вызвана команда Gr.render.

##### Gr.show <object\_number\_nexp>{, <object\_number\_nexp> ...}

Shows (unhides) the objects with the specified Object Numbers. If the Object is a Group, all of the Graphical Objects in the Group are shown. This change will not be visible until the **Gr.render** command is called.

\*\*\*\*\*

Показывает (показывает) объекты с указанными номерами объектов. Если объект является группой, отображаются все графические объекты в группе. Это изменение не будет видно, пока не будет вызвана команда Gr.render.

##### Gr.toFront <object\_number\_nexp>

Moves the object with the specified Object Number visually to the Front. That means this element is the last one on the Display List. If the Object is a Group, all of the Graphical Objects in the Group are on top. This change will not be visible until the **Gr.render** command is called.

\*\*\*\*\*

Перемещает объект с указанным номером объекта визуально вперед. Это означает, что этот элемент является последним в списке отображения. Если объект является группой, все графические объекты в группе находятся сверху. Это изменение не будет видно, пока не будет вызвана команда Gr.render.

##### Gr.behind <object\_behind\_nexp>, <object\_number\_nexp>

Moves the object with the specified Object Number <object\_ibehind\_nexp> in behind the object <object\_number\_nexp> visually. That means the element <object\_behind\_nexp> is in front of the one specified by <object\_number\_nexp> on the Display List. If the Object is a Group, all of the Graphical Objects in the Group are on this position also. This change will not be visible until the **Gr.render** command is called.

\*\*\*\*\*

Визуально перемещает объект с указанным номером объекта <object\_ibehind\_nexp> за объектом <object\_number\_nexp>. Это означает, что элемент <object\_behind\_nexp> находится перед элементом, указанным <object\_number\_nexp> в списке отображения. Если объект является группой, все графические объекты в группе также находятся в этой позиции. Это изменение не будет видно, пока не будет вызвана команда Gr.render

##### Gr.inFront <object\_inFront\_nexp>, <of\_object\_nexp>

Moves the object with the specified Object Number <object\_inFront\_nexp> in front of the object <of\_object\_nexp> visually. That means the element <object\_inFront\_nexp> is behind the one specified by <of\_object\_nexp> on the Display List. If the Object is a Group, all of the Graphical Objects in the Group are on this position also. This change will not be visible until the **Gr.render** command is called.

\*\*\*\*\*

Визуально перемещает объект с указанным номером объекта <object\_inFront\_nexp> перед объектом <of\_object\_nexp>. Это означает, что элемент <object\_inFront\_nexp> находится позади элемента, указанного в <of\_object\_nexp> в списке отображения. Если объект является группой, все графические объекты в группе также находятся в этой позиции. Это изменение не будет видно, пока не будет вызвана команда Gr.render.

##### Gr.toBack <object\_number\_nexp>

Moves the object with the specified Object Number visually to the Back. That means this element is the first one on the Display List. If the Object is a Group, all of the Graphical Objects in the Group are at rear. This change will not be visible until the **Gr.render** command is called.

\*\*\*\*\*

Перемещает объект с указанным номером объекта визуально в заднюю часть. Это означает, что этот элемент является первым в списке отображения. Если объект является группой, все графические объекты в группе находятся сзади. Это изменение не будет видно, пока не будет вызвана команда Gr.render.

##### Gr.render

Sets the graphic screen to invalidate. The system will schedule a redraw of ~~This commanddisplays~~all objects that are listed in the current working Display List and are not marked as hidden. It is not necessary to have a **Pause** command after a **Gr.render**. The **Gr.render** command will not complete until the contents of the Display List have been fully displayed. **Gr.render** always waits until the next screen refresh. Most Android devices refresh the screen 60 times per second; your device may be faster or slower. Therefore, if you

execute two consecutive **Gr.render** commands, there will be a delay of 16.7 milliseconds (on most devices) between the two commands.

For smooth animation, try to avoid doing more than 16.7 ms of work between **Gr.render**

commands, to achieve the maximum refresh rate. This is not a lot of time for a BASIC!

program, so you may have to settle for a lower frame rate. However, there is no benefit to trying to render more often than 16.7 ms.

If BASIC! is running in the background (see **Background()** function and **Home** command), **Gr.render** will not execute. It will pause your program until you return BASIC! to the foreground.

Keep in mind, that under some circumstances (software rendering) a graphic object canbe displayed before **Gr.render** is called. Hardware rendering is the default option and can only changed byGR.set.acceleration.

If the program has to do a lot of work in the background before the result should be displayed, do not draw or change any objects until the background work is complete.

\*\*\*\*\*

Устанавливает графический экран для аннулирования. Система запланирует перерисовку объектов данной команды, которые отображаются в текущем рабочем списке отображения и не помечены как скрытые. Нет необходимости иметь команду Pause после Gr.render. Команда Gr.render не будет выполнена, пока содержимое списка отображения не будет полностью отображено. Gr.render всегда ждет до следующего обновления экрана. Большинство устройств Android обновляют экран 60 раз в секунду; Ваше устройство может быть быстрее или медленнее. Поэтому, если вы

Выполните две последовательные команды Gr.render, между двумя командами будет задержка 16,7 миллисекунды (на большинстве устройств).

Для плавной анимации старайтесь избегать больше чем 16,7 мс работы между Gr.render

Команды, чтобы достичь максимальной частоты обновления. Это не много времени для бейсика!

программа, поэтому вам, возможно, придется согласиться на более низкую частоту кадров. Однако нет смысла пытаться рендерить чаще, чем 16,7 мс.

Если бейсик! работает в фоновом режиме (см. функцию Background () и команду Home), Gr.render не будет выполняться. Это приостановит вашу программу, пока вы не вернетесь BASIC! на передний план.

Имейте в виду, что при некоторых обстоятельствах (рендеринг программного обеспечения) графический объект может отображаться до вызова Gr.render. Аппаратный рендеринг является опцией по умолчанию и может быть изменен только с помощью GR.set.acceleration.

Если программе необходимо выполнить большую работу в фоновом режиме, прежде чем отобразить результат, не рисуйте и не изменяйте какие-либо объекты, пока фоновая работа не будет завершена.

See alsoGR.set.acceleration

### Notify <title\_sexp>, <subtitle\_sexp>, <alert\_sexp>, <wait\_lexp>{{{,

**<options\_bundle\_nexp>}, <notification\_id\_nvar>}, <notified\_id\_nvar>}**

This command will cause a Notify object to be placed in the Notify (Status) bar. The Notify object displays the BASIC! app icon and the <alert\_sexp> text. The user taps the Notify object to open the notification window. Your program’s notification displays the <title\_sexp> and <subtitle\_sexp> text.

The code snippet and screenshots shown below demonstrate the placement of the parameter strings.

If <wait\_lexp> is not zero (true), then the execution of the BASIC! program will be suspended until the user taps the Notify object. The optional numeric variable

<notified\_id\_nvar> returns in this case the notification id of the touched notification. If

<notified\_id\_nvar> returns -1, no notification touch was identified. If the value <wait\_lexp> is zero (false), the BASIC! program will continue executing.

The Notify object will be removed when the user taps or slides the object, or when the program exits. See also the bundle keys \_AutoCancel and \_Ongoing later.

\*\*\*\*\*

Эта команда приведет к тому, что объект Notify будет помещен в панель уведомлений (Status). Объект Notify отображает BASIC! значок приложения и текст <alert\_sexp>. Пользователь нажимает на объект «Уведомить», чтобы открыть окно уведомлений. В уведомлении вашей программы отображается текст <title\_sexp> и <subtitle\_sexp>.

Фрагмент кода и снимки экрана, показанные ниже, демонстрируют размещение строк параметров.

Если <wait\_lexp> не равно нулю (true), то выполнение BASIC! Программа будет приостановлена, пока пользователь не коснется объекта Уведомить. Необязательная числовая переменная

В этом случае <messages\_id\_nvar> возвращает идентификатор уведомления о прикосновении к уведомлению. Если

<messages\_id\_nvar> возвращает -1, касание уведомления не обнаружено. Если значение <wait\_lexp> равно нулю (false), BASIC! Программа продолжит выполнение.

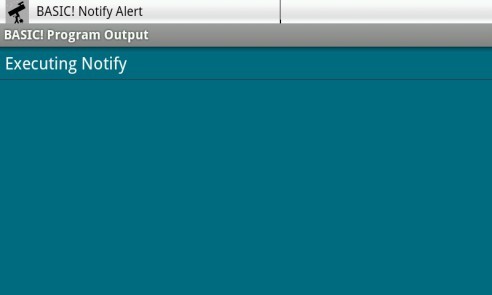
Объект Notify будет удален, когда пользователь касается или перемещает объект, или когда программа выходит. См. Также ключи связки \_AutoCancel и \_ Продолжается позже.

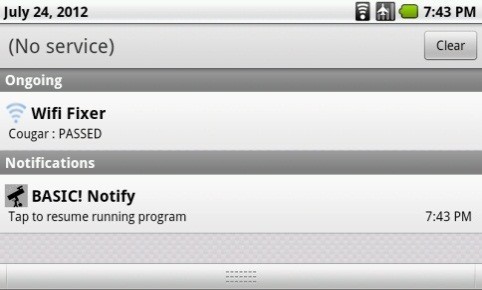
Example:

Print "ExecutingNotify"

Notify "BASIC! Notify", "Tap to resume running program",~ "BASIC! Notify Alert",1

! Execution is suspended and waiting for user to tap the Notify Object Print "Notified"





The optional options bundle <options\_bundle\_nexp> controls the notification:

|  |  |  |
| --- | --- | --- |
| **Key** | **Value** |  |
| **\_SmallIcon** | Small icon file path | Sets the small icon, which will be used to represent the notification in the status bar. The platform template for the expanded view will draw this icon normally in the left.  It is a special bitmap. The visible content is defined only by the alpha channel.<http://romannurik.github.io/>  AndroidAssetStudio/icons- notification.html |
| **\_LargeIcon** | Large icon file path | Add a large icon to the notification content view.<http://romannurik.github.io/>  AndroidAssetStudio/index.html |
| **\_Color** | {Alpha,}Red,Green,Blue (comma delimitedstring) or  \_{Alpha,}ColorName ({comma delim.}string) or  #{hn}hnhnhn (hex. string) | Min. Lollipop 5.0 (API 21) |
| **\_Progress** | Max, progress, animated | Progress bar, Max → maximal (numeric value), progress → between 0 and maximal (numeric value),  animated > 0 → true |
| **\_Sound** | sound file path | Deprecated but  Note: Beginning with Android 8.1 (API level 27), apps cannot make a notification sound more than once per second. If your app posts multiple notifications in one second, they all appear as expected, but only the first notification per second makes a sound.  Beginning with Android 9 (API level  28) system sounds take over. |
| **\_AutoCancel** | 0 or > 0 (numeric) | Makes this notificationautomatically dismissed when the user touches or slidesit.  Default is true (> 0). |

|  |  |  |
| --- | --- | --- |
| **\_Ongoing** | 0 or > 0 (numeric) | Sets whether this is an "ongoing" notification. Ongoing notifications cannot be dismissed by the user, so your application must take care of canceling them. They are typically used to indicate a background task that the user is actively engaged with (e.g., playing music) or is pending in some way and therefore occupying the device (e.g., a file download, sync operation, active network connection).  Default is false (0). |
| **\_ShowWhen** | 0 or > 0 (numeric) | Beginning with Android 7 (API level  26) the notification time has to be set, if you want to display it. Default is false (0). |

The optional numeric variable <notification\_id\_nvar> controls and returns a notification id.

|  |  |
| --- | --- |
| <notification\_id\_nvar> | Control and Result |
| -2 | A new notification id is created by the last nine digits of the current system time in milliseconds. Returns the created Id. |
| -1 | Repeats and returns the last used notification id. |
| 0 | Default, returns 0 |
| Given id | Uses the given id if possible. Otherwise 0 will be used and returned. |

Example

! bundle.put b, "\_SmallIcon", "myIconAlphaChaneled.png" bundle.put b, "\_LargeIcon", "cartman.png"

bundle.put b, "\_Sound", "whee.mp3" maxVal =1000

progress =333

animate = 0

bundle.put b, "\_Progress", int$(maxVal) + "," + int$(progress) + "," + int$(animate) bundle.put b, "\_Color", "255,0,0" % Red

myNnotify\_ID\_1 = -2

Notify "title1", "subtitle", "alert", 0 , b, myNnotify\_ID\_1

? myNnotify\_ID\_1 : ?

bundle.put b, "\_Sound", ""

bundle.put b, "\_Color", "0,255,0" % Green myNnotify\_ID\_2 = -2

Notify "title2", "subtitle", "alert", 0 , b, myNnotify\_ID\_2

? myNnotify\_ID\_2 : ?

bundle.put b, "\_LargeIcon","galaxy.png" bundle.put b, "\_Color", "0,0,255" % Blue myNnotify\_ID\_3 =-2

Notify "title3", "subtitle", "alert", **1** , b, myNnotify\_ID\_3, ret

? myNnotify\_ID\_3 : ?

? ret do until 0

### Notify.cancel {<notification\_id\_nexp>}

Cancels all notifications triggered by your program by default.

Does not overwrite <wait\_lexp> from the command Notify, because in this case the execution of the BASIC! program is suspended until the user taps the Notify object.

If you use the optional <notification\_id\_nexp> with a number > 0 only a notification with this number will be canceled. If <notification\_id\_nexp> is 0 all notifications will be canceled like the default setting.

\*\*\*\*\*

Отменяет все уведомления, запускаемые вашей программой по умолчанию.

Не перезаписывает <wait\_lexp> из команды Notify, потому что в этом случае выполнение BASIC! Программа приостановлена ​​до тех пор, пока пользователь не коснется объекта Уведомить.

Если вы используете необязательный <tification\_id\_nexp> с номером> 0, только уведомление с этим номером будет отменено. Если <tification\_id\_nexp> равно 0, все уведомления будут отменены, как по умолчанию.

### PERMISSIONS

Permissions are special privileges that apps must ask for if they want to access sensitive media on your Android device.

Android devices contain so much personal information, like the exact location, contact data, and cameras that can record the user (permission protection level dangerous). Apps can not just use these unless the user tell them it is okay, because beginning with Android

* 1. "Marshmallow", API level = 23 the permission handling is more defensive. But if your program is running on a lower API-level, no restrictions willhappen.

Permissions with have protection level dangerous **have be requested first** in Basic’s or your application's manifest, by default these are not be granted to your app at first start after the first installation.

*Note, that normal permissions are granted at install time if requested in the manifest.*

The permissions of an app can be checked at any time. If the app is already installed, go to Settings → Apps and locate the app you want to examine. Tap an app and on the About screen, click the Permissions box or scroll down to find the list. Here you can see everything the app asks for. For a programmed link see App.Settings.

File access to Resources, Assets and the Internal file directory should not need any permissions, because these are protected areas.

It is a good practice to check the needed permissions at the first application start directly. To prevent irritations explain your requested permissions in detail, because if you trigger the DEVICE command with a bundle, you need Phone permissions. But if you do not want to make a call, you should describe that you need only device information.

Do not be surprised if the application user later changes one or more of the required permissions.

May be the user denied the permissions for camera and microphone for privacy reasons. So you should take appropriate precautions. Using permission.get is a possibility.

If you want to compile your program as an APK, Permission.checkPath is useful. You can check, is the given file path is a candidate for a file permission request.

*Note, that permissions in conjunction with Internal Directories on an External Directory or on removable SD-cards were not tested until now.*

In conjunction to deliver your program as an APK the conclusion is, ask as soon as possible for all needed permissions. Use assets and the internal directory for easy file permission control, because in this case you do not have to request for permissions.

\*\*\*\*\*

Разрешения - это особые привилегии, которые приложения должны запрашивать, если они хотят получить доступ к конфиденциальной информации на вашем Android-устройстве.

Устройства Android содержат так много личной информации, как точное местоположение, контактные данные и камеры, которые могут записывать пользователя (уровень защиты разрешений опасен). Приложения не могут просто использовать их, если пользователь не скажет им, что все в порядке, потому что, начиная с Android

«Зефир», уровень API = 23, обработка разрешений более оборонительная. Но если ваша программа работает на более низком уровне API, никаких ограничений не будет.

Разрешения с опасным уровнем защиты сначала запрашиваются в Basic или манифесте вашего приложения, по умолчанию они не предоставляются вашему приложению при первом запуске после первой установки.

Обратите внимание, что обычные разрешения предоставляются во время установки, если они запрашиваются в манифесте.

Разрешения приложения могут быть проверены в любое время. Если приложение уже установлено, перейдите в «Настройки» → «Приложения» и найдите приложение, которое вы хотите проверить. Нажмите на приложение и на экране «О программе» щелкните поле «Разрешения» или прокрутите вниз, чтобы найти список. Здесь вы можете увидеть все, что просит приложение. Для запрограммированной ссылки см. App.Settings.

Доступ к файлам ресурсов, ресурсов и внутреннего каталога файлов не должен требовать каких-либо разрешений, поскольку они являются защищенными областями.

Рекомендуется проверять необходимые разрешения при первом запуске приложения. Чтобы избежать раздражения, подробно объясните запрошенные вами разрешения, потому что если вы запускаете команду DEVICE с помощью пакета, вам нужны разрешения для телефона. Но если вы не хотите совершать звонок, вам следует указать, что вам нужна только информация об устройстве.

Не удивляйтесь, если пользователь приложения позже изменит одно или несколько необходимых разрешений.

Может быть, пользователь отказал в разрешении на камеру и микрофон из соображений конфиденциальности. Поэтому вы должны принять соответствующие меры предосторожности. Использование allow.get возможно.

Если вы хотите скомпилировать вашу программу как APK, Permission.checkPath будет полезен. Вы можете проверить, является ли данный путь к файлу кандидатом на запрос разрешения файла.

Обратите внимание, что разрешения в сочетании с внутренними каталогами на внешнем каталоге или на съемных SD-картах до сих пор не тестировались.

В связи с тем, что для доставки вашей программы в виде APK вам нужно как можно скорее запросить все необходимые разрешения. Используйте ресурсы и внутренний каталог для простого контроля прав доступа к файлам, потому что в этом случае вам не нужно запрашивать разрешения.

##### Permission.automatic <auto\_nvar>

OliBasic has an automatic permission detection and request. Which can be switched to OFF by <auto\_nvar> = 0 or to ON by <auto\_nvar> = 1 (default).

If <auto\_nvar> = 2 it is switched to ON but with no file permissions.

It is only an option to switch to OFF in case of massive data file transfer because the system delay. If possible Permission.ignore is the better option.

\*\*\*\*\*

OliBasic имеет автоматическое обнаружение и запрос разрешения. Который может быть выключен с помощью <auto\_nvar> = 0 или включен с помощью <auto\_nvar> = 1 (по умолчанию).

Если <auto\_nvar> = 2, он включен, но без прав доступа к файлу.

Это только возможность переключиться на OFF в случае массивной передачи файла данных из-за системной задержки. Если возможно, Permission.ignore - лучший вариант.

##### Permission.checkPath <nvar>, <checkPath\_sexp>, <dump\_nexp>

If <nvar> = 1 the given file name <checkPath\_sexp> is a candidate for checking the STORAGE permission related to the shared external storage (READ\_EXTERNAL\_STORAGE, WRITE\_EXTERNAL\_STORAGE).

If <dump\_nexp> > 0 detailed information are printed.

\*\*\*\*\*

Если <nvar> = 1, указанное имя файла <checkPath\_sexp> является кандидатом для проверки разрешения STORAGE, связанного с общим внешним хранилищем (READ\_EXTERNAL\_STORAGE, WRITE\_EXTERNAL\_STORAGE).

Если <dump\_nexp>> 0, выводится подробная информация.

##### Permission.ignore <file\_path\_Array$[]>

Dealing with file access permissions requires some care.

Assets and Internal directories should be ignored for permission checking.

File names including "files:///data/", "Android/data/(Application’s package name)" and "asset://" will be ignored by default.

If you deal with relative file paths, it could be useful to add your own exceptions.

\*\*\*\*\*

Работа с разрешениями на доступ к файлам требует некоторой осторожности.

Активы и внутренние каталоги следует игнорировать для проверки разрешений.

Имена файлов, включая «files: /// data /», «Android / data / (имя пакета приложения)» и «asset: //», будут игнорироваться по умолчанию.

Если вы имеете дело с относительными путями к файлам, может быть полезно добавить ваши собственные исключения.

##### Permission.get <granted\_lvar>, <permission\_sexp>

Returns <granted\_lvar> = 1 if the permission given by <permission\_sexp> is granted. Otherwise <granted\_lvar> returns 0.

\*\*\*\*\*

Возвращает <grant\_lvar> = 1, если предоставлено разрешение, данное <license\_sexp>. В противном случае <grant\_lvar> возвращает 0.

|  |  |
| --- | --- |
| **Permissions used by AndroidManifest.xml** | **Level Dangerous** |
| ACCESS\_COARSE\_LOCATION |  |
| ACCESS\_FINE\_LOCATION |  |
| ACCESS\_LOCATION\_EXTRA\_COMMANDS | |
| ACCESS\_MOCK\_LOCATION | |
| ACCESS\_NETWORK\_STATE | |
| ACCESS\_SUPERUSER | |
| ACCESS\_WIFI\_STATE | |
| BLUETOOTH\_ADMIN | |
| BLUETOOTH | |
| CALL\_PHONE |  |
| CAMERA |  |
| INSTALL\_SHORTCUT | |
| INTERNET | |
| KILL\_BACKGROUND\_PROCESSES | |
| READ\_CONTACTS |  |
| READ\_EXTERNAL\_STORAGE |  |
| READ\_PHONE\_STATE |  |
| READ\_SMS |  |
| READ\_USER\_DICTIONARY | |
| RECEIVE\_SMS |  |
| RECORD\_AUDIO |  |
| RUN\_SCRIPT (Only to detect over APP.settings!) |  |
| SEND\_SMS |  |
| UNINSTALL\_SHORTCUT | |
| VIBRATE | |
| WAKE\_LOCK | |
| WRITE\_EXTERNAL\_STORAGE |  |
| WRITE\_SETTINGS | |

*Note, that WRITE\_EXTERNAL\_STORAGE includes the READ\_EXTERNAL\_STORAGE permission.*

##### Permission.request <permission\_sexp> | Array$[]

For Android 6.0 Marshmallow, API level = 23 and later.

Requests permissions to be granted to the running Basic-engine or your application(APK).

\*\*\*\*\*

Для Android 6.0 Marshmallow, уровень API = 23 и выше.

Запрашивает разрешения, которые будут предоставлены работающему движку Basic или вашему приложению (APK).

|  |  |
| --- | --- |
| **Level Dangerous Permission Group** | **Related Permissions** |
| **CALENDAR**  Used for runtime permissions related to user's calendar. | READ\_CALENDAR WRITE\_CALENDAR |
| **CALL\_LOG** (Since Android 9)  This permission group gives control and visibility to apps that need access to sensitive information about phone calls, such as reading phone call records and identifying phone numbers. | PROCESS\_OUTGOING\_CALLS READ\_CALL\_LOG WRITE\_CALL\_LOG |
| **CAMERA**  Used for permissions that are associated with accessing camera or capturing images/ video from the device. | CAMERA |
| **CONTACTS**  Used for runtime permissions related to contacts and profiles on this device. | READ\_CONTACTS WRITE\_CONTACTS GET\_ACCOUNTS |
| **LOCATION**  Used for permissions that allowaccessing the device location. OliBasic supports by default only ACCESS\_FINE\_LOCATION (GPS and Network). ~~If onlyACCESS\_COARSE\_LOCATIONwanted,~~  ~~use Permission.request"ACCESS\_COARSE\_LOCATION" atapp~~  ~~start first. Both together are not allowed.~~ | ~~(ACCESS\_COARSE\_LOCATION)~~ACCESS\_FINE\_LOCATION |
| **MICROPHONE**  Used for permissions that are associated with accessing microphone audio from the device. | RECORD\_AUDIO |
| **PHONE**  Used for permissions that are associated telephony features. | ANSWER\_PHONE\_CALLS READ\_PHONE\_STATE READ\_PHONE\_NUMBERS CALL\_PHONE ADD\_VOICEMAIL USE\_SIP  (Until Android 8.1 PROCESS\_OUTGOING\_CALLS READ\_CALL\_LOG WRITE\_CALL\_LOG ) |
| **SENSORS**  Used for permissions that are associated | BODY\_SENSORS |

|  |  |
| --- | --- |
| with accessing body or environmental sensors. |  |
| **SMS**  Used for runtime permissions related to user's SMS messages. | READ\_SMS SEND\_SMS RECEIVE\_SMS RECEIVE\_WAP\_PUSH RECEIVE\_MMS |
| **STORAGE**  Used for runtime permissions related to the shared external storage. | READ\_EXTERNAL\_STORAGE WRITE\_EXTERNAL\_STORAGE |
| **Random Terminal Scripts** | Only to control over APP.settings |

*Note, that not all table permissions are used in conjunction with Basic.*

Normal permissions are granted at install time if requested in the manifest.

If your program does not have the requested permissions the user will be presented with UI for accepting them.

Note that requesting a permission does not guarantee it will be granted and your program should be able to run without having this permission.

This command may start an activity allowing the user to choose which permissions to grant and which to reject. Hence, you should be prepared that your activity may bepaused and resumed. **Further, granting some permissions may require a restart of you application!**

When checking whether you have a permission you should use Permission.get. Google Reference:

Calling this command for permissions already granted to your app would show UI to the user to decide whether the app can still hold these permissions. This can be useful ifthe way your app uses data guarded by the permissions changessignificantly.

##### But after experiences:

The UI is only created , if the permission is not granted.

*Note, use the single string only, if will request one permission.*

*A second one or more without a pause will be ignored. Use instead an array.*

*\*\*\*\*\**

*Обратите внимание, что не все разрешения для таблиц используются вместе с Basic.*

*Обычные разрешения предоставляются во время установки, если это требуется в манифесте.*

*Если ваша программа не имеет запрошенных разрешений, пользователю будет представлен пользовательский интерфейс для их принятия.*

*Обратите внимание, что запрос разрешения не гарантирует его получение, и ваша программа должна быть в состоянии работать без этого разрешения.*

*Эта команда может запустить действие, позволяющее пользователю выбирать, какие разрешения предоставлять, а какие отклонять. Следовательно, вы должны быть готовы к тому, что ваша деятельность может быть приостановлена ​​и возобновлена. Кроме того, предоставление некоторых разрешений может потребовать перезапуска вашего приложения!*

*При проверке наличия разрешения вы должны использовать Permission.get. Справочник Google:*

*Вызов этой команды для разрешений, уже предоставленных вашему приложению, покажет пользователю пользовательский интерфейс, чтобы решить, может ли приложение по-прежнему хранить эти разрешения. Это может быть полезно, если способ, которым ваше приложение использует данные, защищенные разрешениями, изменяется значительно.*

*Но после переживаний:*

*Пользовательский интерфейс создается, только если разрешение не предоставлено.*

*Обратите внимание, используйте только одну строку, если будет запрашивать одно разрешение.*

*Второй или более без паузы будут игнорироваться. Используйте вместо этого массив.*

Example:

Permission.request “READ\_PHONE\_STATE”

See also:

Permission.get, App.settings

##### IDEAS

Nicolas’s Stuff

**Program.info <nexp>|<nvar>**

Returns a Bundle that reports information about the currently running program. If you provide a variable that is not a valid Bundle pointer, the command creates a new Bundle and returns the Bundle pointer in your variable. Otherwise it writes into the Bundle your variable or expression points to.

The bundle keys and possible values are in the table below:

\*\*\*\*\*

Возвращает Bundle, который сообщает информацию о запущенной в данный момент программе. Если вы предоставляете переменную, которая не является допустимым указателем Bundle, команда создает новый Bundle и возвращает указатель Bundle в вашей переменной. В противном случае он записывает в Bundle вашу переменную или выражение, на которое указывает.

Ключи комплекта и возможные значения приведены в таблице ниже:

|  |  |  |
| --- | --- | --- |
| **Key** | **Type** | **Value** |
| **\_BasPath** | String | Full path + name of the program currently being executed.  The path is relative to BASIC!’s "source/" directory. See also File.root |
| **\_BasName** | String | Name of the program currently being executed. |
| **\_SysPath** | String | Full path to the BASIC!’s private file storage directory. The path is relative to BASIC!’s "data/" directory.  See also File.root |
| **\_UserApk** | Numeric (Logical) | Returns 1.0 (true) if the current program is being run from a standalone user-built APK (Appendix D).  Returns 0.0 (false) if the program is being from the BASIC! Editor or a Launcher Shortcut (Appendix C). |
| **\_LauncherStart** | Numeric (Logical) | Returns 1.0 (true) if the current program is being run from a Launcher Shortcut (Appendix C) or an intent.  Returns 0.0 (false) in other cases. |
| **\_PackageName** | String | package\_id |
| **\_MyProcessId** | String | my\_process\_id |
| **\_TotalUserRam** | Numeric | The total memory accessible by the kernel. This is basically the RAM size of the device, not including below-kernel fixed allocations like DMA buffers, RAM for the baseband CPU, etc. |
| **\_AvailableRam** | Numeric | The available memory on the system. This number should not be considered absolute: dueto the nature of the kernel, a significant portion of this memory is actually in use and needed for the overall system to runwell. |
| **\_ThresholdRam** | Numeric | The threshold of available memory at which we consider memory to be low and start killing background services and other non-extraneous processes. |
| **\_HeapLimit** | Numeric | Return the approximate per-application memory class of the current device. This gives you an idea of how hard a memory limit you should impose on your application to let the overall system work best. |
| **\_NativeHeap** | Numeric | Returns the (dynamic) size of the native heap. |
| **\_NativeHeapAllocated** | Numeric | Returns the amount of allocated memory in the native heap. |
| **\_NativeHeapFree** | Numeric | Returns the amount of free memory in the native heap. |

For example, assume:

* + - You are using the default <pref basedrive>
    - You downloaded a file called "my\_program.bas" to the standard AndroidDownload directory.
    - You used the BASIC! Editor to load and run the downloaded program.

|  |  |
| --- | --- |
| **Key** | **Value** |
| **\_BasPath** | ../../Download/my\_program.bas |
| **\_BasName** | my\_program.bas |
| **\_SysPath** | ../../../../../data/data/com.rfo.basic |
| **\_UserApk** | 0.0 |
| **\_PackageName** | **com.rfo.basic** |
| **\_MyProcessId** | **21615** |
| **\_TotalUserRam** | **2759.0** in megabytes. |
| **\_AvailableRam** | **1370.0** in megabytes. |
| **\_ThresholdRam** | **216.0** in megabytes. |
| **\_HeapLimit** | **192.0** in megabytes. |
| **\_NativeHeap** | **15.89...** in megabytes. |
| **\_NativeHeapAllocated** | **12.19...** in megabytes. |
| **\_NativeHeapFree** | **3.69...** in megabytes. |

Then the returned values would be asfollows:

\*\*\*\*\*

Например, предположим:

Вы используете <pref basedrive> по умолчанию>

Вы загрузили файл my\_program.bas в стандартный каталог AndroidDownload.

Вы использовали бейсик! Редактор для загрузки и запуска загруженной программы. Тогда возвращаемые значения будут следующими:

**SysPath**: BASIC! normally keeps programs and data in its *base directory* (see **Working with Files**, later in this manual). The base directory is in public storage space (external). BASIC! programs also have access to a private (internal) storage area. Your program can create a sub directory within the SysPath directory and store private files there. Note that if you uninstall BASIC!, any files in private storage will be deleted.

SysPath is of particular interest to you if you build a BASIC! program as an application in a standalone apk, as described in Appendix D. See **Modifications to the AndroidManifest.xml File** **Permissions**.

You can extent the heap size by inserting android:largeHeap="true"; in the APK’s AndroidManifest.xml File.

\*\*\*\*\*

SysPath: ОСНОВНОЙ! обычно хранит программы и данные в своем базовом каталоге (см. «Работа с файлами» далее в этом руководстве). Базовый каталог находится в публичном хранилище (внешнем). BASIC! Программы также имеют доступ к частному (внутреннему) хранилищу. Ваша программа может создать подкаталог в каталоге SysPath и хранить там личные файлы. Обратите внимание, что если вы удалите BASIC !, все файлы в личном хранилище будут удалены.

SysPath представляет особый интерес для вас, если вы создаете бейсик! запрограммируйте как приложение в отдельном apk, как описано в Приложении D. См. Изменения в файле AndroidManifest.xml erPermissions.

Вы можете увеличить размер кучи, вставив android: largeHeap = "true"; в файле APKManifest.xml APK.

### Zip Commands now in RFO-Basic 1.91 included

ZIP File I/O

The ZIP file I/O commands work with compressed files. ZIP is an archive file format that stores multiple directories and files, using a method of lossless data compression to save file space.

Use **Zip.dir** to get an array containing the names of all of the directories and files in an archive. Use the file names with **Zip.read** to extract files from the archive. **Zip.read** can not extract a directory. Use **Zip.write** to put files in a new archive. You can overwrite an existing ZIP file, but you cannot replace or add entries.

\*\*\*\*\*

Команды ввода / вывода ZIP-файла работают со сжатыми файлами. ZIP - это формат файла архива, в котором хранятся несколько каталогов и файлов, используя метод сжатия данных без потерь для экономии места на диске.

Используйте Zip.dir, чтобы получить массив, содержащий имена всех каталогов и файлов в архиве. Используйте имена файлов с Zip.read для извлечения файлов из архива. Zip.read не может извлечь каталог. Используйте Zip.write, чтобы поместить файлы в новый архив. Вы можете перезаписать существующий ZIP-файл, но не можете заменить или добавить записи.

### Zip.dir <path\_sexp>, Array$[] ~~{,<dirmark\_sexp>}~~{,<timeStamp\_nexp>}

Returns the names of the files including the internal path beginning with ZIP root ~~anddirectories~~inside the ZIP file which is located at <path\_sexp>. The path is relative to "<pref base drive>/rfo-basic/data/".

The names are placed into Array$[]. The array is sorted alphabetically ~~with the directoriesat the top of the list~~. If the array exists, it is overwritten, otherwise a new array is created. The result is always a one-dimensional array.

~~A directory is identified by a marker appended to its name. The default marker is the string"(d)". You can change the marker with the optional directory mark parameter~~

~~<dirmark\_sexp>. If you do not want directories to be marked, set <dirmark\_sexp> to anempty string, "".~~

If the directory is empty, Zip.dir returns an array with one item and a string with one space (" ") in it.

\*\*\*\*\*

Возвращает имена файлов, включая внутренний путь, начинающийся с корня ZIP и директории внутри файла ZIP, который находится в <path\_sexp>. Путь относительно "<pref base drive> / rfo-basic / data /".

Имена помещаются в массив $ []. Массив отсортирован в алфавитном порядке с каталогами в верхней части списка. Если массив существует, он перезаписывается, в противном случае создается новый массив. Результатом всегда является одномерный массив.

Каталог идентифицируется маркером, добавленным к его имени. Маркер по умолчанию - это строка "(d)". Вы можете изменить маркер с помощью дополнительного параметра метки каталога

<Dirmark\_sexp>. Если вы не хотите, чтобы каталоги были помечены, установите для <dirmark\_sexp> пустую строку "".

Если каталог пуст, Zip.dir возвращает массив с одним элементом и строку с одним пробелом ("").

Options of <timeStamp\_nexp>:

* 0 no time stamp (default), sortedalphabetically
* 1 with time stamp as time in milliseconds + ":" + file name, butunsorted
* 2 with time stamp as time in milliseconds + ":" + file name, sorted in ascendingorder
* 3 with time stamp as time in milliseconds + ":" + file name, sorted in descendingorder

GitHub#246

### Zip.open {r|w|a}, <file\_table\_nvar>, <path\_sexp>

The ZIP file specified by the path string expression <path\_sexp> is opened. The path is relative to "<pref base drive>/rfo-basic/data/".

The first parameter is a single character that sets the I/O mode for this file:

\*\*\*\*\*

ZIP-файл, указанный в строковом выражении <path\_sexp>, открыт. Путь относительно "<pref base drive> / rfo-basic / data /".

Первый параметр - это один символ, который устанавливает режим ввода-вывода для этого файла:

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Mode** | **Notes** |
| r | read | File exists: Reads from the start of the file. File does not exist: Error (see below). |
| w = a | write | File exists: Writes from the start of the file. Writes over any existing data.  File does not exist: Creates a new file. Writes from the start of the file. |

Note: unlike **Text.open** and **Byte.open**, **Zip.open** does not support an append mode.

A file table number is placed into the numeric variable <file\_table\_nvar>. This value is for use in subsequent **Zip.read**, **Zip.write**, or **Zip.close** commands.

If there was an error opening the ZIP file, <file\_table\_nvar> is set to -1 with details available from the **GETERROR$()** function.

Zip.open has no access to **assets** and **res**ources in conjunction with APKs. If you want to read a ZIP that is in assets, you must first copy it from assets to an internal or externalfile system. Then you can open this new file withZip.Open.

\*\*\*\*\*

Примечание: в отличие от Text.open и Byte.open, Zip.open не поддерживает режим добавления.

Номер таблицы файлов помещается в числовую переменную <file\_table\_nvar>. Это значение предназначено для использования в последующих командах Zip.read, Zip.write или Zip.close.

Если при открытии ZIP-файла произошла ошибка, для <file\_table\_nvar> устанавливается значение -1 с подробностями, доступными из функции GETERROR $ ().

Zip.open не имеет доступа к активам и ресурсам в сочетании с APK. Если вы хотите прочитать ZIP-архив с активами, вы должны сначала скопировать его из ресурсов во внутреннюю или внешнюю файловую систему. Затем вы можете открыть этот новый файл с помощью Zip.Open.

See also Byte.copy

# Good to know

**Something about AndoidManifest.xml**

If you want only encoded IP connections like Https instead of Http change android:usesCleartextTraffic to false.

# Something about touch events

Android triggers touch events at a distance of at least 16 to 18 milliseconds.

|  |  |  |
| --- | --- | --- |
| **Standard Values**  Some of them can be changed by the user in global preferences.  All device independent pixels (DIPs) are in conjunction to the Android standard 160 DPI resolution. | | |
| Tap Timeout | 100 | Duration in milliseconds we will wait to see if a touch event is a tap or a scroll. If the user does not move within this interval, it is considered to be a tap. |
| Double Tap Timeout | 300 | Duration in milliseconds between the first tap's up event and the second tap's down event for an interaction to be considered a double-tap. |
| Long Press Timeout | 400 | Duration in milliseconds before a press turns into a long press |
| Maximum Fling Velocity | 8000 | Maximum velocity to initiate a fling, as measured in DIPs per second. |
| Minimum Fling Velocity | 50 | Minimum velocity to initiate a fling, as measured in DIPs per second. |
| Touch Slop | 8 | Distance in DIPs a touch can wander before we think the user is scrolling |

**NaN (Not a Number) or Infinity values (IEEE 754)**

These are floating point values in our case from type Double.

If you try SQR (-1) or 0/0 you get a NaN, because it is an undefined operation. But if you try 1/0 you get an Infinit**y**, because it is an infinit**e** result.

Or you get these values per definition like VAL("Infinity"), VAL("-Infinity") andVAL("NaN").

ThisdiffersfromolderBasicimplementationsandinsomepartsalsofromRFO-Basic1.91.

That should not have a negative impact on older code. Hopefully in this development stage, you will only get a runtime error if you want to handle these values and functions, where input values must be of the type Integer or Long.

Functions which use BigDecimal inside do not deal with NaN (Not a Number) or Infinity values. Examples: BigD(decimal) command group, List.join

Functions which use <lexp> and <nexp> interpret these values as 0.

For detection use Is\_NaN and Is\_Infinit**e**. The last one returns -1 if the value is negative.

!NaN Example DIM x[1]

x[1] = 1

ARRAY.STD\_DEV nanV, x[ ] % Returns NaN in **all** Basic! versions.

IF nanV THEN ? "true" : ELSE ? "false" % → **true** But that is **wrong, see below**

IF Is\_Number("NaN") THEN ? "true" : ELSE ? "false" % → true IF Is\_NaN(nanV) THEN ? "true" : ELSE ? "false" % → true

IF nanV = nanV THEN ? "true" : ELSE ? "false" % → false IF nanV <> nanV THEN ? "true" : ELSE ? "false" % →true IF nanV < 0 THEN ? "true" : ELSE ? "false" % →false

IF nanV > 0 THEN ? "true" : ELSE ? "false" % →false

!Infinity Example

!infiV = -1/0 %Returns -Infinity.

infiV = val("-Infinity") % Returns -Infinit**y** in **all** Basic! versions.

? infiV

IF infiV THEN ? "true" : ELSE ? "false" % → true

IF Is\_Number("-Infinity") THEN ? "true" : ELSE ? "false" % → true IF Is\_Infinit**e**(infiV) THEN ? "true" : ELSE ? "false" % → true

?Is\_Infinit**e**(infiV) % →-1

IF infiV = infiV THEN ? "true" : ELSE ? "false" % → true IF infiV <> infiV THEN ? "true" : ELSE ? "false" % →false IF infiV < 0 THEN ? "true" : ELSE ? "false" % →true

IF infiV > 0 THEN ? "true" : ELSE ? "false" % → false

**Use instaed of**

IF nanV THEN ? "true" : ELSE ? "false" % → **true** But that is **wrong**

IF **!Is\_NaN(nanV) &** nanV THEN ? "true" : ELSE ? "false" % → false That is **right**

WHILE nanV % → **true** But that is **wrong**

WHILE **!Is\_NaN(nanV) &** nanV % → false That is **right**

UNTIL nanV % → **true** But that is **wrong**

UNTIL **!Is\_NaN(nanV) &** nanV % → false That is **right**

**Comparison with NaN**

A comparison with a NaN always returns an *unordered result* even when comparing with itself. The comparison predicates are either signaling or non-signaling on quiet NaN operands; the signaling versions signal the invalid operation exception for such comparisons. The equality and inequality predicates are non-signaling so *x* = *x* returning false can be used to test if *x* is a quiet NaN. The other standard comparison predicates are all signaling if they receive a NaN operand, the standard also provides non-signaling versions of these other predicates. The predicate *isNaN(x)* determines if a value is a NaN and never signals an exception, even if *x* is a signaling NaN.

Comparison between NaN and any floating-point value *x* (including NaN and ±∞)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Comparison | NaN ≥ *x* | NaN ≤ *x* | NaN > *x* | NaN < *x* | NaN = *x* | NaN ≠ *x* |
| Result | Always **False** | Always **False** | Always **False** | Always **False** | Always **False** | Always **True** |

See also

<https://docs.oracle.com/cd/E19957-01/806-3568/ncg_intro.html#110>

https://docs.oracle.com/cd/E19957-01/806-3568/ncg\_goldberg.html

# Fixes

##### GR.TOUCH2 and GR.BOUNDED.TOUCH2 <touched> returns 0.0 instead of 1.0 if finger 1 is lifted up. When the second finger already touches the display, a second touch of the first finger initiates a rebound of the second finger without raising it! Fixed

**Calling Basic! by a modern file browser fails if it uses a document path beginning with "content://".**

**Fixed**

**WIFI.INFO Android issue returns wrong results for SSID, BSSID and MACaddress. Fixed**

**GR.BITMAP.CROP Crash, if wished bounds are outside the given.** GitHub**#268 Fixed**

**"&=" was dropping into "|=" op immediately after, giving wrong result.** Humpty**#0243 Fixed**

**GR.ARC, GR.MODIFY Some devices do not ignore NaN or Infinity values. Now you get a runtime error, if Gr.Arc (angels) or get NaN or Infinity values.** GitHub**#267 Fixed**

**GR.TOUCH <touched> returns sometimes 1.0 instead of 0.0 in special cases. Fixed**

**GR.ARRAY.TOUCH <count\_nvar> returns 1.0 instead of 0.0. Fixed**

**GR.LIST.TOUCH <count\_nvar> returns only 0.0. Fixed**

**GR.CAMERA.\_ \_ \_SHOOT commands return no valid bitmappointer. Fixed**

**CONSOLE.SAVE in graphics or HTML mode returns an empty file. Fixed by a runtime error message**

**EMAIL.SEND the last four optional arguments are working as a group instead of single arguments**

**Fixed**

**SENSORS.READ crashes on Android 8 if a sensor returns only one or two arguments** GitHub**#265**

**Fixed**

**Ctrl A, C, V, X**

**Fixed**

**Byte.write.buffer can only write one time beforeByte.close Fixed**

**Split returns the same as Split.all Fixed**

**GrabURL reads cache** GitHub**#264 Fixed**

**FILE.EXISTS fails, if a variable and a String are part of the secondargument Fixed**

**ROUND fails often, because imprecise double import** GitHub**#262 Fixed**

**BIGD.FROMDOUBLE imprecise double import Fixed**

**BIN$ Input of a NaN (Not a Number) or Infinity Value returns 0** GitHub**#260 Fixedwithareturnof"NaN"**

**INT Input of a NaN (Not a Number) or Infinity Value returns 0.0** GitHub**#259 Fixedwithareturnof"NaN"**

**INT$ Input of a NaN (Not a Number) or Infinity Value returns 0** GitHub**#259 Fixedwithareturnof"NaN"**

**HEX$ Input of a NaN (Not a Number) or Infinity Value returns 0** GitHub**#259 Fixedwithareturnof"NaN"**

**OFT$ Input of a NaN (Not a Number) or Infinity Value returns 0** GitHub**#259 Fixedwithareturnof"NaN"**

**BIN$ Input of a NaN (Not a Number) or Infinity Value returns 0** GitHub**#259 Fixedwithareturnof"NaN"**

**USING$ Input of a NaN (Not a Number) or Infinity Value for arguments needing Integer or Long values returns 0** GitHub**#258**

**Fixed, Double values return NaN, Integer values throw runtime errors**

**ROUND Input of a NaN (Not a Number) or Infinity Value returns 0.0 or crashes if the scale is specified.** GitHub**#257**

**Does not round if only the scale is specified. Fixed**

**The Editor stops until REPLACE ALL in Search is used with empty search string**

GitHub**#256 Fixed**

##### BYTE.WRITE.BUFFER is very slow GitHub#255 Improved

**GR.CAMERA. \_ \_ \_SHOOT The file name extension from "image.png" is wrong in a case of jpeg compression** GitHub**#254**

**Fixed**

**STT.RESULTS list variable does not reset at program start** GitHub**#253**

**VOLKEYS.ON / .OFF initialization and logic backward referenced** GitHub**#251 Fixed**

**SENSORS.LIST names not all 20 standard sensor types.** GitHub**#250 Fixed**

**BIGD.ROUND returns 0 Fixed**

**BUNDLE.PUT needs expressions in brackets like (n+1) or(h$+p$) Fixed**

**BUNDLE.GET detects no number array Fixed**

**ZIP.DIR …, <dirmark\_sexp> is not used** GitHub**#246 Fixed by deleting parameter**

**BUNDLE.PUT sends wrong error message if array does not exists Fixed**

**Carriage Return characters in program code are misinterpreted** GitHub**#243 Fixed by ignoring**

**APP.START Package Name and Component Name** GitHub**#240**

**Fixed,** butthereisstillaninternaldesignissuethroughsendingunneededdata,whichtriggers sometimesunexpectedresults.TodaywithAPP.SARyouareonthesafeside.

##### PRINT command before CLS : PRINT "Text" returns empty Screen GitHub#239 Fixed

**Editor SAVE File name fails if ".bAs" file extension contains uppercase characters # Fixed to “.bas” in any case**

**Some strange behaviors interacting with other BASIC! instances mainly on lame devices** GitHub**#29, #209, #238**

**Fixed**

**SOCKET.CLIENT.READ.FILE result** GitHub**#237**

**Fixed**

**SOCKET.CLIENT.WRITE.BYTES Fails after first call. At the second call the socket is disconected.** GitHub**#237**

**Fixed**

**SOCKET.CLIENT.WRITE.FILE result** GitHub**#237**

**Fixed**

**GrabURL long time stability Improved**

**FRAC() does not work correctly with big numbers like 4.5678879E8** GitHub**#234 Fixed *Be aware, with Double you get maximal only 15 correct digits.***

##### DEBUG.DUMP.BUNDLE returns an error

**RUN activate some enhancements also for APKmode Fixed**

**FILE.ROOT sometimes returns null instead of"" Fixed**

**BUNDLE.SAVE and BUNDLE.LOAD sometimes a key is missed Fixed**

**RUN "" crashes BASIC!** GitHub **#216 See #218 below**

##### Sometimes SELECT crashes, if to many data in List or Array The intent transfer memory issue solved

**Global Value Backdoor, if interrupt happens FixedwithCommandGroupsLOCALSandGLOBALS, Improved with the support of nestedfunctions**

**Other enhancements**

**Add fullscreen support for HTML5 videos** GitHub **#224 SuggestionborrowedfromNicolasM[ougino](https://github.com/mougino)**

**Prevent BASIC! halt in case of unhandled Intent** GitHub **#221 SuggestionborrowedfromNicolasM[ougino](https://github.com/mougino)**

**Prevent user APK crashing b/c of bad/missing permission** GitHub**#220 SuggestionborrowedfromNicolasM[ougino](https://github.com/mougino)**

**Add support for an APK to register file extension(s)** GitHub **#219 Nicolas M[ougino](https://github.com/mougino) suggested a command COMMAND$(),**

**but this is also retData in**

**Bundle.in <recAction\_sexp>, <retData\_svar>, <retBundleIndex\_nvar>**

**Android’sdataExtraacceptsonlyURIswiththequeryseparator("?"),fragmentseparator("#'"). Simplecommandlineoptionslike-s,-helpetc.arenotallowed.**

**UseDECODE$withthetype"URL"andtheQualifier"charset".**

Workaround:

FN.DEF Command$()

Bundle.in recAction$, retData$, retBundleIndex FN.RTN DECODE$ ("URL","charset", retData$)

FN.END

ftn$ = Command$()

**RUN command to restart current program** GitHub **#218 An idea from Nicolas Mougino, but use a little different way.**

**Command PROGRAM.INFO** GitHub **#217**

**A Suggestion from Nicolas Mougino~~, but use a some different values~~. See also: File.root and App.installed**

##### Launching in Editor Mode

**At launching with a given program path and an Intent Extra with a key named "\_BASIC!" and a String expression "\_Editor" the BASIC! program starts in the Editor mode.**

##### Launching the Editor at an exact position

**At launching the Editor with a given program path and an Intent Extra with a key named "\_BASIC!" and a String expression "\_Editor?start=<nexp>?end=<nexp>" the BASIC! opens the Editor optionally marking the given area.**

##### Broadcast on Runtime Error

On Runtime Error a Broadcast is send.

To receive this Broadcast with BASIC! use OnBroadcast:

action$ = BasicEnginePackageName$ + ".broadcast.ERROR" BROADCAST.IN retAction$, retData$, retBundleIndexF

IF retAction$ = action$

BUNDLE.GB retBundleIndexF, "\_Error", retBundleIndex BUNDLE.GET retBundleIndex, "\_RuntimeError", runtimeError$ PRINT "\_RuntimeError: "; runtimeError$

BUNDLE.GET retBundleIndex, "\_ErrorText", errorText$ PRINT "\_ErrorText: "; errorText$

BUNDLE.GET retBundleIndex, "\_ErrorLog", errorLog$ PRINT "\_ErrorLog: "; errorLog$

BUNDLE.GET retBundleIndex, "\_LineNumber", lineNumber$ PRINT "\_LineNumber: "; lineNumber$

BUNDLE.GET retBundleIndex, "\_LostContext", lostContext$ PRINT "\_LostContext: "; lostContext$

BUNDLE.GET retBundleIndex, "\_PackageName", packageName$ PRINT "\_PackageName: "; packageName$

ENDIF

Broadcast.resume

##### Overwriting bitmaps with valid pointers

**Now, bitmaps with a valid pointer will be overwritten.**

**Until OliBasicXXI, some GR commands always generate new entries in a Java list containing all the bitmaps. The list index therefore increases to very high values.**

1. **Thus the speed slowsdown.**
2. **The bitmap had to be deleted before the pointer was usedagain.**

##### Console.Save in graphics or HTML mode

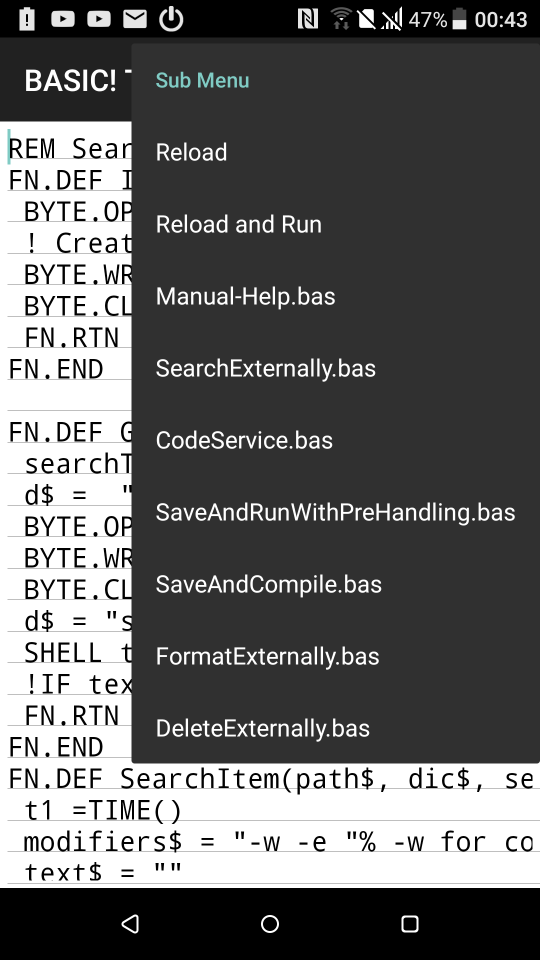
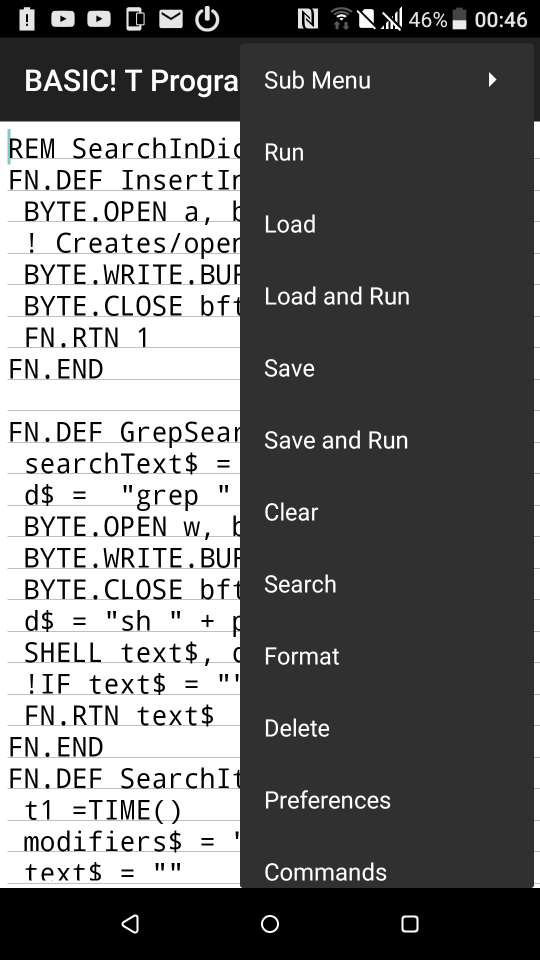
**Now, Console.Save returns an error if the program is still in graphics or HTML mode. Use Gr.close and HTML.close before!**

##### Automatic word completions or -corrections GitHub #247

Automatic word completions or -corrections are **switched to off**, because you get in trouble with the value name "thi" that returns maybe "think" after inserting a space character.

**Editor enhancements** GitHub **#**

The Editor supports a sub menu and keycodes for special commands.



The entry SUB MENU ICON on action bar in Preferences/Menu\_Items\_on\_Action\_bar, supports direct access to the SubMenu.

|  |  |  |  |
| --- | --- | --- | --- |
| **Menu Action**, bas files in source/**Service\_Programs/** | **Menu** | **Fn Keys** | Key Combination |
| Manual-Help.bas | Sub Menu | F1 | Ctrl + Q (Query Help) |
| Commands |  | F2 | Alt + Ctrl + Q |
| Search |  | F3 | Ctrl + F (Find) |
| SearchExternally.bas | Sub Menu | Ctrl + F3 | Alt + Ctrl + F |
| Load |  | F4 | Ctrl + O (Open) |
| Reload | Sub Menu | Ctrl + F4 | Ctrl + R (Reload) |
| Save and Run |  | F5 | Ctrl + G (Go) |
| SaveAndRunWithPreHandling.bas | Sub Menu | Ctrl + F5 | Alt + Ctrl + G |
| Save |  | F6 | Ctrl + S (Save) |
| SaveAndCompile.bas | Sub Menu | Ctrl + F6 | Alt + Ctrl + S |
| Run |  | F7 | Ctrl + 7 (/) |
| CodeService.bas | Sub Menu | Ctrl + F7 | Alt + Ctrl +7 |
| Format |  | F8 | Ctrl + 3 (§) |
| FormatExternally.bas | Sub Menu | Ctrl + F8 | Alt + Ctrl +3 |
| Clear |  | F9 | Ctrl + N (New) |
| Delete |  | F10 | Ctrl + D (Delete) |
| DirectoriesFiles.bas | Sub Menu | Ctrl + F10 | Alt + Ctrl +D |
| Previous | Sub Menu | F11 | Ctrl + P (Previous) |
| Preferences |  | Ctrl + F11 | Alt + Ctrl + P (Preferences) |
| Load and Run |  | F12 | Alt + Ctrl +O |
| Reload and Run | Sub Menu | Ctrl + F12 | Alt + Ctrl + R |

If a Sub Menu item starts with "Reload" the last loaded/saved Basic! program code will be reloaded. It is independent under which circumstances Basic! was be closed.

If a Sub Menu item ends with ".bas" it is linked to an equal named Basic! program in the source sub folder Service\_Programs.

If a Basic program in source/Service-Programs called, the current program file name and the selection/cursor will be send like a RUN command.

With Android versions < 6 you have to put the selection into the clipboard, if the selection dialog hides the menu.

Unfortunately does some devices not support hardware keyboard function keys.

In this case, you can use the menu key and the up and down keys as well as the enter key.

Some devices have problems with repeated pressing of the buttons. Try another keyboard or / and an active (power supply) USB hub. Maybe you can insert batteries into your keyboard.

A keyboard or mouse with a Back key is recommended.

|  |  |  |
| --- | --- | --- |
| Navigating the program code / document | | |
| **To** | **Press** |  |
| Select text | Shift + LEFT/RIGHT ARROW  or Shift + UP/DOWN ARROW |  |
| Move cursor right by one word | Ctrl + RIGHT ARROW |  |
| Move cursor left by one word | Ctrl + LEFT ARROW |  |
| Move cursor to beginning of the document | Alt+ UP ARROW |  |
| Move cursor to end of the document | Alt + DOWN ARROW |  |
| Move cursor to the beginning of current line | Alt + LEFT ARROW |  |
| Move cursor to the end of the current line | Alt + RIGHT ARROW |  |
| Select word to the left | Shift + Ctrl + LEFT ARROW |  |
| Select word to the right | Shift + Ctrl + RIGHT ARROW |  |
| Select from current position to beginning of the document | Shift + Alt + UP ARROW |  |
| Select from current position to end of the document | Shift + Alt + DOWN ARROW |  |
| Select from current position to beginning of the line | Shift + Alt + LEFT ARROW |  |
| Select from current position to end of the line | Shift + Alt + RIGHT ARROW |  |

|  |  |  |
| --- | --- | --- |
| Editing (and formatting) the program code / document | | |
| **To** | **Press** |  |
| Undo the last action | Ctrl + Z | Not supported in the Editor |
| Repeat the last action | Ctrl + Y | Not supported in the Editor |
| Cut selected content | Ctrl + X |  |
| Copy selected content | Ctrl + C |  |
| Paste copied or cut content | Ctrl + V |  |
| Select all | Ctrl + A |  |
| Bold selected content | Ctrl + B | Not supported in the Editor |
| Italicize selected content | Ctrl + I | Not supported in the Editor |
| Underline selected content | Ctrl + U | Not supported in the Editor |
|  |  |  |
| Text to speech commands | | |
| **To** | **Press** |  |
| TextToSpeech Selection | Ctrl + 0 | Starts at cursor position to the next line end  or  the whole selection |
| TextToSpeech Clipboard | Alt + Shift + 0 |  |
| TextToSpeech Stop | Alt + Ctrl + 0 |  |

##### Working with Service Programs

With the SubMenu, Hot Keys and FunctionKeys from above you are able to load and execute Service Programs coded in Basic!.

If a Sub Menu item ends with ".bas" it is linked to an equal named Basic! program in the source sub folder Service\_Programs.

If a Basic program in source/Service-Programs is called, the current program file name and the selection/cursor will be send like the RUN command.

The Service\_Programs directory is direct behind the source dirctory.

. . . . . /rfo-basic/source/Service Programs/ The structure is:

Service Programs

CodeService.bas cs\_data

data Manual\_Help.bas mh\_data

data

. ..

. ..

See also RUN

Example:

The blue line is inserted like the RUN command. The path part “/storage/emulated/0” is device depended. The variable ##$ is used by the function GetBasObjectValues().

##$="/storage/emulated/0/rfo-basic/source/myProgramName.bas?start=0?end=0? package=com.rfo.basicOli"

REM Example for CodeService.bas

FN.DEF ReLaunch(basEngine$, basProgramPath$, mode, mStart, mEnd) eMode$ = ""

IF mode > 0 eMode$ = "\_Editor"

IF mStart > -1 & mEnd > -1 THEN eMode$ = eMode$ + "?start=" + INT$(mStart) + "?end=" + INT$(mEnd)

ENDIF

LIST.CREATE S, commandListPointer LIST.ADD commandListPointer~

"new Intent(Intent.ACTION\_MAIN);" ~

"setData("+ CHR$(34) + basProgramPath$ + CHR$(34) +");" ~

"new ComponentName("+ CHR$(34) + basEngine$ + CHR$(34) + ","+CHR$(34)+ basEngine$ + ".Basic" + CHR$(34)+");" ~

"addCategory(Intent.CATEGORY\_DEFAULT);" ~

"putExtra("+ CHR$(34) + "\_BASIC!" + CHR$(34) + ","+CHR$(34)+ eMode$ + CHR$(34)+");" ~

%Starts program in Editor mode, if eMode$ = "\_Editor"! "addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP);" ~ "addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);" ~ "addFlags(Intent.FLAG\_ACTIVITY\_MULTIPLE\_TASK);" ~

"addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK);" ~ "EOCL"

BUNDLE.PL appVarPointer,"\_CommandList",commandListPointer APP.SAR appVarPointer

FN.END

FN.DEF InitReLaunch(objectValues)

BUNDLE.GET objectValues, "basProgramPath", basProgramPath$ BUNDLE.GET objectValues, "startSelection", startSelection BUNDLE.GET objectValues, "endSelection", endSelection

! basEngine$ ="com.rfo.basicOli" % Or your favorite BASIC! Engine BUNDLE.GET objectValues, "packageID", basEngine$

mode = 1

CALL ReLaunch(basEngine$, basProgramPath$, mode, startSelection, endSelection) FN.END

FN.DEF GetBasObjectValues (objectValues)

GLOBALS.FNIMP ##$ % On start a line started with ##$ is added by Basic!. See also the RUN command.

SPLIT.ALL spRes$[], ##$, "\\?" % ? is a control delimiter, \\ Regular Exp. specific, because ? sign ARRAY.LENGTH al, spRes$[]

IF al > 0

BUNDLE.PUT objectValues, "basProgramPath", spRes$[1] BUNDLE.PUT objectValues, "startSelection", VAL(MID$(spRes$[2], 7)) BUNDLE.PUT objectValues, "endSelection", VAL(MID$(spRes$[3], 5)) BUNDLE.PUT objectValues, "packageID", VAL(MID$(spRes$[4], 9)) ENDIF

FN.END

!\*\*\* Main Part \*\*\* BUNDLE.CREATE objectValues GetBasObjectValues(objectValues)

!\* The Selection Parameters

!\* If startSelection = endSelection THEN it is the cursor position

!\* The position values start with 0 (before the first sign) BUNDLE.GET objectValues, "startSelection", startSelection BUNDLE.GET objectValues, "endSelection", endSelection

? startSelection

? endSelection

CLIPBOARD.GET mClipData$ % Get automatically created text from the selected Basic source contents

? mClipData$

!\* Insert your code here! sel = -6000

DIALOG.MESSAGE "Do your code here!", "In 6 seconds we will return!", sel CLIPBOARD.PUT mClipData$

!\* Set the new cursor/selection position(s)

BUNDLE.PUT objectValues, "startSelection", startSelection BUNDLE.PUT objectValues, "endSelection", endSelection

InitReLaunch(objectValues) EXIT

##### Color Table

Predefined Colors based on official **HTML Color Names** of the World Wide Web Consortium (W3C)

|  |  |  |  |
| --- | --- | --- | --- |
| **HTML Color Name** | **Basic! Color Code** | **Basic! Color Name** | **HTML Color Code Hex** |
| AliceBlue **with Alpha** | **255,**240,248,255 | **\_255,**AliceBlue | #**FF**F0F8FF |
| AliceBlue | 240,248,255 | \_AliceBlue | #F0F8FF |
| AntiqueWhite | 250,235,215 | \_AntiqueWhite | #FAEBD7 |
| Aqua | 0,255,255 | \_Aqua | #00FFFF |
| Aquamarine | 127,255,212 | \_Aquamarine | #7FFFD4 |
| Azure | 240,255,255 | \_Azure | #F0FFFF |
| Beige | 245,245,220 | \_Beige | #F5F5DC |
| Bisque | 255,228,196 | \_Bisque | #FFE4C4 |
| Black | 0,0,0 | \_Black | #000000 |
|  |  |  | or #000 |
|  |  |  | or #**F**000 |
| BlanchedAlmond | 255,235,205 | \_BlanchedAlmond | #FFEBCD |
| Blue | 0,0,255 | \_Blue | #0000FF |
| BlueViolet | 138,43,226 | \_BlueViolet | #8A2BE2 |
| Brown | 165,42,42 | \_Brown | #A52A2A |
| BurlyWood | 222,184,135 | \_BurlyWood | #DEB887 |
| CadetBlue | 95,158,160 | \_CadetBlue | #5F9EA0 |
| Chartreuse | 127,255,0 | \_Chartreuse | #7FFF00 |
| Chocolate | 210,105,30 | \_Chocolate | #D2691E |
| Coral | 255,127,80 | \_Coral | #FF7F50 |
| CornflowerBlue | 100,149,237 | \_CornflowerBlue | #6495ED |
| Cornsilk | 255,248,220 | \_Cornsilk | #FFF8DC |
| Crimson | 220,20,60 | \_Crimson | #DC143C |
| Cyan | 0,255,255 | \_Cyan | #00FFFF |
| DarkBlue | 0,0,139 | \_DarkBlue | #00008B |
| DarkCyan | 0,139,139 | \_DarkCyan | #008B8B |
| DarkGoldenRod | 184,134,11 | \_DarkGoldenRod | #B8860B |
| DarkGray | 169,169,169 | \_DarkGray | #A9A9A9 |
| DarkGreen | 0,100,0 | \_DarkGreen | #006400 |
| DarkKhaki | 189,183,107 | \_DarkKhaki | #BDB76B |
| DarkMagenta | 139,0,139 | \_DarkMagenta | #8B008B |
| DarkOliveGreen | 85,107,47 | \_DarkOliveGreen | #556B2F |
| DarkOrange | 255,140,0 | \_DarkOrange | #FF8C00 |
| DarkOrchid | 153,50,204 | \_DarkOrchid | #9932CC |
| DarkRed | 139,0,0 | \_DarkRed | #8B0000 |
| DarkSalmon | 233,150,122 | \_DarkSalmon | #E9967A |
| DarkSeaGreen | 143,188,143 | \_DarkSeaGreen | #8FBC8F |
| DarkSlateBlue | 72,61,139 | \_DarkSlateBlue | #483D8B |
| DarkSlateGray | 47,79,79 | \_DarkSlateGray | #2F4F4F |
| DarkTurquoise | 0,206,209 | \_DarkTurquoise | #00CED1 |
| DarkViolet | 148,0,211 | \_DarkViolet | #9400D3 |
| DeepPink | 255,20,147 | \_DeepPink | #FF1493 |
| DeepSkyBlue | 0,191,255 | \_DeepSkyBlue | #00BFFF |
| DimGray | 105,105,105 | \_DimGray | #696969 |
| DodgerBlue | 30,144,255 | \_DodgerBlue | #1E90FF |
| FireBrick | 178,34,34 | \_FireBrick | #B22222 |
| FloralWhite | 255,250,240 | \_FloralWhite | #FFFAF0 |
| ForestGreen | 34,139,34 | \_ForestGreen | #228B22 |
| Fuchsia | 255,0,255 | \_Fuchsia | #FF00FF |

|  |  |  |  |
| --- | --- | --- | --- |
| **HTML Color Name** | **Basic! Color Code** | **Basic! Color Name** | **HTML Color Code Hex** |
| Gainsboro | 220,220,220 | \_Gainsboro | #DCDCDC |
| GhostWhite | 248,248,255 | \_GhostWhite | #F8F8FF |
| Gold | 255,215,0 | \_Gold | #FFD700 |
| GoldenRod | 218,165,32 | \_GoldenRod | #DAA520 |
| Gray | 128,128,128 | \_Gray | #808080 |
| Green | 0,128,0 | \_Green | #008000 |
| GreenYellow | 173,255,47 | \_GreenYellow | #ADFF2F |
| HoneyDew | 240,255,240 | \_HoneyDew | #F0FFF0 |
| HotPink | 255,105,180 | \_HotPink | #FF69B4 |
| IndianRed | 205,92,92 | \_IndianRed | #CD5C5C |
| Indigo | 75,0,130 | \_Indigo | #4B0082 |
| Ivory | 255,255,240 | \_Ivory | #FFFFF0 |
| Khaki | 240,230,140 | \_Khaki | #F0E68C |
| Lavender | 230,230,250 | \_Lavender | #E6E6FA |
| LavenderBlush | 255,240,245 | \_LavenderBlush | #FFF0F5 |
| LawnGreen | 124,252,0 | \_LawnGreen | #7CFC00 |
| LemonChiffon | 255,250,205 | \_LemonChiffon | #FFFACD |
| LightBlue | 173,216,230 | \_LightBlue | #ADD8E6 |
| LightCoral | 240,128,128 | \_LightCoral | #F08080 |
| LightCyan | 224,255,255 | \_LightCyan | #E0FFFF |
| LightGoldenRodYellow | 250,250,210 | \_LightGoldenRodYellow | #FAFAD2 |
| LightGray | 211,211,211 | \_LightGray | #D3D3D3 |
| LightGreen | 144,238,144 | \_LightGreen | #90EE90 |
| LightPink | 255,182,193 | \_LightPink | #FFB6C1 |
| LightSalmon | 255,160,122 | \_LightSalmon | #FFA07A |
| LightSeaGreen | 32,178,170 | \_LightSeaGreen | #20B2AA |
| LightSkyBlue | 135,206,250 | \_LightSkyBlue | #87CEFA |
| LightSlateGray | 119,136,153 | \_LightSlateGray | #778899 |
| LightSteelBlue | 176,196,222 | \_LightSteelBlue | #B0C4DE |
| LightYellow | 255,255,224 | \_LightYellow | #FFFFE0 |
| Lime | 0,255,0 | \_Lime | #00FF00 |
| LimeGreen | 50,205,50 | \_LimeGreen | #32CD32 |
| Linen | 250,240,230 | \_Linen | #FAF0E6 |
| Magenta | 255,0,255 | \_Magenta | #FF00FF |
| Maroon | 128,0,0 | \_Maroon | #800000 |
| MediumAquaMarine | 102,205,170 | \_MediumAquaMarine | #66CDAA |
| MediumBlue | 0,0,205 | \_MediumBlue | #0000CD |
| MediumOrchid | 186,85,211 | \_MediumOrchid | #BA55D3 |
| MediumPurple | 147,112,219 | \_MediumPurple | #9370DB |
| MediumSeaGreen | 60,179,113 | \_MediumSeaGreen | #3CB371 |
| MediumSlateBlue | 123,104,238 | \_MediumSlateBlue | #7B68EE |
| MediumSpringGreen | 0,250,154 | \_MediumSpringGreen | #00FA9A |
| MediumTurquoise | 72,209,204 | \_MediumTurquoise | #48D1CC |
| MediumVioletRed | 199,21,133 | \_MediumVioletRed | #C71585 |
| MidnightBlue | 25,25,112 | \_MidnightBlue | #191970 |
| MintCream | 245,255,250 | \_MintCream | #F5FFFA |
| MistyRose | 255,228,225 | \_MistyRose | #FFE4E1 |
| Moccasin | 255,228,181 | \_Moccasin | #FFE4B5 |
| NavajoWhite | 255,222,173 | \_NavajoWhite | #FFDEAD |
| Navy | 0,0,128 | \_Navy | #000080 |
| OldLace | 253,245,230 | \_OldLace | #FDF5E6 |
| Olive | 128,128,0 | \_Olive | #808000 |
| OliveDrab | 107,142,35 | \_OliveDrab | #6B8E23 |
| Orange | 255,165,0 | \_Orange | #FFA500 |
| OrangeRed | 255,69,0 | \_OrangeRed | #FF4500 |

|  |  |  |  |
| --- | --- | --- | --- |
| **HTML Color Name** | **Basic! Color Code** | **Basic! Color Name** | **HTML Color Code Hex** |
| Orchid | 218,112,214 | \_Orchid | #DA70D6 |
| PaleGoldenRod | 238,232,170 | \_PaleGoldenRod | #EEE8AA |
| PaleGreen | 152,251,152 | \_PaleGreen | #98FB98 |
| PaleTurquoise | 175,238,238 | \_PaleTurquoise | #AFEEEE |
| PaleVioletRed | 219,112,147 | \_PaleVioletRed | #DB7093 |
| PapayaWhip | 255,239,213 | \_PapayaWhip | #FFEFD5 |
| PeachPuff | 255,218,185 | \_PeachPuff | #FFDAB9 |
| Peru | 205,133,63 | \_Peru | #CD853F |
| Pink | 255,192,203 | \_Pink | #FFC0CB |
| Plum | 221,160,221 | \_Plum | #DDA0DD |
| PowderBlue | 176,224,230 | \_PowderBlue | #B0E0E6 |
| Purple | 128,0,128 | \_Purple | #800080 |
| RebeccaPurple | 102,51,153 | \_RebeccaPurple | #663399 |
| Red | 255,0,0 | \_Red | #FF0000 |
| RosyBrown | 188,143,143 | \_RosyBrown | #BC8F8F |
| RoyalBlue | 65,105,225 | \_RoyalBlue | #4169E1 |
| SaddleBrown | 139,69,19 | \_SaddleBrown | #8B4513 |
| Salmon | 250,128,114 | \_Salmon | #FA8072 |
| SandyBrown | 244,164,96 | \_SandyBrown | #F4A460 |
| SeaGreen | 46,139,87 | \_SeaGreen | #2E8B57 |
| SeaShell | 255,245,238 | \_SeaShell | #FFF5EE |
| Sienna | 160,82,45 | \_Sienna | #A0522D |
| Silver | 192,192,192 | \_Silver | #C0C0C0 |
| SkyBlue | 135,206,235 | \_SkyBlue | #87CEEB |
| SlateBlue | 106,90,205 | \_SlateBlue | #6A5ACD |
| SlateGrey | 112,128,144 | \_SlateGrey | #708090 |
| Snow | 255,250,250 | \_Snow | #FFFAFA |
| SpringGreen | 0,255,127 | \_SpringGreen | #00FF7F |
| SteelBlue | 70,130,180 | \_SteelBlue | #4682B4 |
| Tan | 210,180,140 | \_Tan | #D2B48C |
| Teal | 0,128,128 | \_Teal | #008080 |
| Thistle | 216,191,216 | \_Thistle | #D8BFD8 |
| Tomato | 255,99,71 | \_Tomato | #FF6347 |
| Turquoise | 64,224,208 | \_Turquoise | #40E0D0 |
| Violet | 238,130,238 | \_Violet | #EE82EE |
| Wheat | 245,222,179 | \_Wheat | #F5DEB3 |
| White | 255,255,255 | \_White | #FFFFFF |
| WhiteSmoke | 245,245,245 | \_WhiteSmoke | #F5F5F5 |
| Yellow | 255,255,0 | \_Yellow | #FFFF00 |
| YellowGreen | 154,205,50 | \_YellowGreen | #9ACD32 |