

REXXOID

Running Rexx on Android

Julian Reindorf

26th International Rexx
Symposium, 2015



CONTENTS

- Rexxoid introduction
- Nutshell examples
- Limitations

REXXOID

- Available on Google Play Store
- Execute Rexx Code
- Send commands to Android shell
- Read out aloud
- No documentation

HELLO WORLD

```
1 SAY "Hello World";
```

OPENING THE BROWSER

```
1 SAY "Which url would you like to visit?"  
2 PARSE PULL url  
3 "am start -a android.intent.action.VIEW  
  --user 0  
  -d http:" || url
```

SENDING KEYEVENTS

```
1 DO 5
2   "input keyevent 24"
3   "sleep 0.6"
4 END
5 DO 5
6   "input keyevent 25"
7   "sleep 0.6"
8 END
```

COMMAND LINE

```
1 DO WHILE TRANSLATE(input) <> 'stop'  
2   PARSE PULL input  
3   input  
4 END  
5 SAY "*Exited*"   
6 EXIT
```

USING THE SPEAKER

```
1 CALL CHAROUT "Speaker:lang=en", "I am the voice of REXXoid"  
2 CALL CHAROUT "Speaker:lang=de", "Und ich kann auch andere  
Sprachen, wie Deutsch"  
3 CALL CHAROUT "Speaker:lang=fr", "Au revoir"
```


READING AND WRITING FILES

```
1 path = "/sdcard/"
2 name = "foo.txt"
3 CALL CHAROUT path || name, "Written to file at" DATE() TIME()
4 CALL CHAROUT path || name, "Some interesting text"
6 text = CHARIN(path || name, 1, 1000)
7 SAY text
```

CREATING EMAILS

```
1 text = ""
2 DO i=1 TO 100
3   text = text i
4 END
5 "am start -a android.intent.action.SEND
  --user 0
  -t 'text/plain'
  -e to 'john.doe@gmail.com'
  -e android.intent.extra.SUBJECT '1 to 100!'
  -e android.intent.extra.TEXT '"text"'"
```

OPENING GOOGLE MAPS

```
1 SAY "What do you want to see?"
2 input = LINEIN()
3 location=""
4 DO WHILE LENGTH(input) > 0
5     PARSE VAR input char +1 input
6     location=location || "%" || C2X(char)
7 END
8 "am start -a android.intent.action.VIEW
  --user 0
  -d 'geo:0,0?q="location"'
```

SETTING ALERTS

```
1 SAY "*Alert Creation*"
2 SAY "Hour?"
3 PULL hour
4 SAY "Minute?"
5 PULL minute
6 SAY "Message?"
7 PULL message
8 "am start -a android.intent.action.SET_ALARM
  --user 0
  --ei android.intent.extra.alarm.HOUR " hour "
  --ei android.intent.extra.alarm.MINUTES " minute "
  -e android.intent.extra.alarm.MESSAGE ' " message "'
  --ez android.intent.extra.alarm.VIBRATE false
  --ez android.intent.extra.alarm.SKIP_UI true"
9 SAY "Done!"
```

ADDING CALENDAR EVENTS

```
1 NUMERIC DIGITS 18
2 SAY "Title?"
3 PULL title
4 SAY "Date? (e.g. 12 Feb 2012)"
5 PARSE PULL inputDate
6 SAY "Time? (e.g. 12:30)"
7 PULL inputTime
8 PARSE VAR inputTime inputHours ":" inputMinutes
9 SAY "Duration? (in hours)"
10 PULL durationHours
11 daysSinceEpoch = DATE("B", inputDate) - DATE("B", "01 Jan 1970")
12 millisSinceEpoch = daysSinceEpoch*24*60*60*1000
13 millisSinceEpoch = millisSinceEpoch + (inputHours*60+inputMinutes)*60*1000
14 millisSinceEpoch = millisSinceEpoch - 60*60*1000
15 "am start -a android.intent.action.INSERT
    --user 0
    -d Events.CONTENT_URI
    -t vnd.android.cursor.dir/event
    --es title '" title "'
    --el beginTime " millisSinceEpoch "
    --el endTime " millisSinceEpoch+durationHours*60*60*1000
```

LIMITATIONS

- Execute commands after starting an activity

```
1 "am start -a android.intent.action.VIEW
  --user 0
  -d 'geo:0,0?q=vienna' "
2 "sleep 10"
3 "am start -a android.intent.action.VIEW
  --user 0
  -d 'geo:0,0?z=16' "
```

Thanks for your attention!

Enjoy the rest of the symposium :)

julian.reindorf@gmail.com