SKScheMa Standard

License Agreement....2
SKScheMa General....3
Events....3
  Reminder....3
  Do nothing....4
  Event Properties 4
Scripts....6
  Script Editor....6
  Variables and pseudovariables (macros)....11
  Script execution sequence control....12
  Work with strings (text)....14
  Working with ZIP archives....14
Profiles....16
Support....16
  ©2008, SKKV Software....16

SKScheMa General

SKScheMa is an excellent tool to schedule unattended running of applications, scripts, reminder and much more. SKScheMa not only allows you to launch programs at scheduled times but also to send keypresses to those programs. Note: if you turn off your smartphone (smartphones not have «suspend» mode) scheduled events will started after you switch device on.

Events

'Event' in SKScheMa stands for any action which can be executed by the program. It can be opening a file, playing a sound (a particular case of a file opening), a message with sound alert, device on/off, screen on/off, wireless functions (bluetooth, wi-fi – device-depended), a phone call, sending SMS, launching a script, etc. SKScheMa is oriented to execution of recurrent events (e. g. to occur every Tuesday of January), as a particular case – an event can be configured to occur only once. Events scheduled to execute are displayed in this window. All elements of this list are stored in the system Notification Queue.
You can enable/disable event. If event disabled SKScheMa anyway enable device, but not run any command.

You can create new events, edit the existing events, clone events, save them as a file (profile),load them from a file (profile). You may define what events will are shown in queue, use Tools->Settings->Queue filter.

Reminder

This is a special type of an event. Once it occurs, you will be notified of it with a sound alert (optional – a vibrating sound alert). The text previously set-up by you will be shown at that. An event can be interrupted or postponed at the time specified. Special parameters for the Reminder Mode are below:

Sound file
Here you can choose a *.mp3, *.wav, *.wma or other sound file (if your system has a player for this file type) to play a sound for the Reminder. If you enter name of file like \My Documents\mp3\*.mp3 then SKScheMa will use random mp3 file from folder \My Documents\mp3.

Sound script (optional, only for *.wav files)
This is rules definition for play sound sequence.
This allow you add vibrate and/or LED blink to sound sequence.
Possible commands:
a = activate device
cN = set volume to N in percentage max volume
fN = flash notification LED for N seconds
p = play ringtone. Note that this will play the ringtone all the way through before continuing with the next code.
r = repeat. Note that this should be the last code in your Script string, if used at all.
vN = vibrate for N seconds
wN = wait for N seconds. Note that the device will wait this long before continuing with the next code.
Use [...] button for select script command or predefined scripts.

Repeat sound file
Here you can set to repeat the sound or not (only for wav and other types associated with wmplayer).

Sound volume
Here you can set the sound volume level or choose the system settings.

Vibrate
Here you can enable or disable vibration. Also, you can set the vibration time.

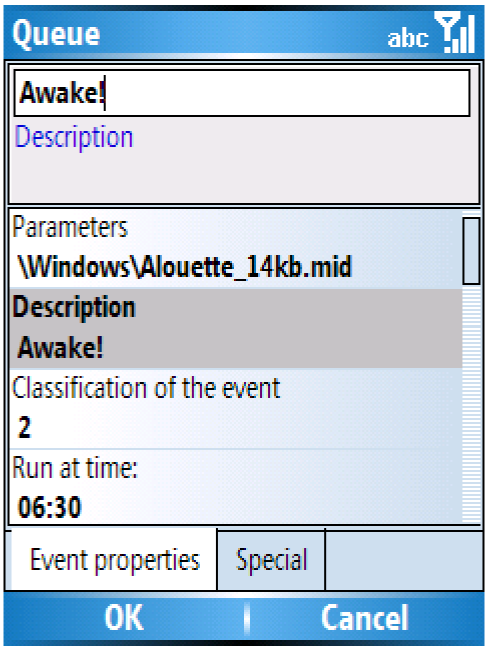
Display reminder window only X seconds
Set the time in seconds to display the reminder window.

Snooze delay (min.)
Set the time in minutes to repeat the reminder (window).

Do nothing

This is a special type of an event. No operations will be performed.. Setting up other parameters of an event you can program some actions periodically: e. g. switch an active device off, change volume level etc.

Event Properties



Awake!

Parameters
\Windows\Alouette_14kb.mid
Description
Awake!
Classification of the event
2
Run at time:
06:30

Here you can plan and set up your events. For any event, basic properties set-up option is available; for some of the events there are also special properties set-up feature (see the corresponding tab).

For fields "Command", "Parameters", "Description" text can include this macros: @time(format) and @date(format) and @dayofweek. SKScheMa will replace it to current date-time with format:

for @time: h Hours with no leading zero for single-digit hours; 12-hour clock hh Hours with leading zero for single-digit hours; 12-hour clock H Hours with no leading zero for single-digit hours; 24-hour clock HH Hours with leading zero for single-digit hours; 24-hour clock m Minutes with no leading zero for single-digit minutes mm Minutes with leading zero for single-digit minutes s Seconds with no leading zero for single-digit seconds ss Seconds with leading zero for single-digit seconds	for @date: d Day of month as digits with no leading zero for single-digit days dd Day of month as digits with leading zero for single-digit days ddd Day of week as a three-letter abbreviation dddd Day of week as its full name M Month as digits with no leading zero for single-digit months MM Month as digits with leading zero for single-digit months MMM Month as a three-letter abbreviation MMMM Month as its full name y Year as last two digits, with a leading zero for years less than 10. The same format as "yy" yy Year as last two digits, with a leading zero for years less than 10 yyyy Year represented by full four digits

you can use commands like:
#msg(Today @date(yyyy/MM/dd) @time(HH:mm))

Command
You can select command from [^] menu.

Parameters
Here you can use parameters like the command line to launch an executable file.

Description
Here you can describe an event. The description will be visible on the SKScheMa window.

Run at time
Here you can set the start time of an event in HH:MM.
If the other time parameters have no settings, the event will occur daily at the time specified. If you need to set the event occurrence time precisely on a specific date, use the following event periodicity parameters:

Sequence script
You can control sequense of event execution like one day - will run, next day too run, next day skip etc.
Enter "r" if you want run and "s" if you want skip.
Example:
you want use wakeup reminder 3 day and after you not want it one day - so use this sequence:
rrrs

Days of week
The event will occur on every specified day of the week.

Days of month
The event will occur on every specified day of the month.

Months
The even will only occur in the specified month(s) of the year.

Years
The event will only occur in the specified year(s).

Run only one time
The event will occur only once. If this option is disabled, the event will occur again.

Other parameters:

Wait until the started process is terminated
SKScheMa will wait until the started process is terminated.

Off screen after start
This option switches off the screen of the device after the start.

Wait before starting the process
SKScheMa will make a pause (you will set a waiting time) before it starts the process.

Wait while X is not accessible
SKScheMa will wait while the target (or another file) is not accessible.

Minimum battery level
SKScheMa checks the battery status before starting an event. If the battery level is below the specified level – no actions will be performed and the power of the device will remain off.

Sound volume
Here you can set the sound volume level or choose the system settings.

Scripts

‘Script’ is a set of commands stored in a file with the extension *.sksc. This allows to perform several actions for one event. A script can also be run from a file directly which is convenient for tasks automation.
In this window, you can create and edit scripts.

Script Editor

This window allows to change/create a script.
From "Script" menu you can:

open script, save and rename current script, create shortcut for script, execute script.
From "Edit" menu you can:
Select and insert command or command parameters.

Command
'script’ is a set of commands stored in a file with the extension *.sksc.
This file have these format:
#command 1(parameter1;...;parameterN) [#sleep(millisecond)] [other options]
#command 2(parameter1;...;parameterN) [#sleep(millisecond)] [other options]
...
#command N(parameter1;...;parameterN) [#sleep(millisecond)] [other options]

Possible to use old format:

#r(command 1) #p(parameters) [#sleep(millisecond)] [other options]
#r(command 2) #p(parameters) [#sleep(millisecond)] [other options]
....
#r(command N) #p(parameters) [#sleep(millisecond)][other options]

You can use this command line for run script files:
:SKSCHM #r(#script) #p(script-file-name.sksc) #onlyrun #VAR1(any text) #VAR2(any text) #VAR3(any text)
With command line You can define three variables - this variables can be accessible from script with names @var1,@var2,@var3.

Script commands:

[*The name of the file to be launched***]**
\Program Files\SKKV Software\SKScheMa\skschema.exe)

#sr (soft reset), you can use this command not only at end of script, but at any line. Execution will is continued after rebooting
#pwr (power off)
#scr (screen off)
#scn (screen oт)

#call (phone call, example #call) #phone(1234567))
#sms (send SMS, example #sms(message text) #phone(1234567))
#**answer** answer
#**hangup** hangup
#**soundvol** set sound volume, from 0 (system) to 15, 16 - mute
#**phonevol** phone sound volume, from 0 to 5

#**script** (start script, example #script(script file))

#**0** (do nothing)
#**wlanon** (WLAN on)
#**wlanoff** (WLAN off)
#**bton** (BT on)
#**btoff** (BT off)

#**findwindow** (this command is searching for a window by these criteria (set in the Parameters box of this command): main window title; main window class; child window title, child window class,without parameters find active window
#**activatewindow** if #findwindow success this command activate obtained window
#**getwndtext** if #findwindow success this command copy to variable title of obtained window, *#getwndtext(4)* - will copy title to Var4
#**getwndclass** if #findwindow success this command copy to variable class name of obtained window, *#getwndclass(4)* - will copy class name to Var4
#**getwndexe** if #findwindow success this command copy to variable exe name of obtained window, *#getwndexe(4)* - will copy executable name to Var4
#**postmessage** if #findwindow success this command can send message to window , *#postmessage(0xMessageCode,0xwParam,0xlParam)*

#**copyfile** copy file *#copyfile(source path\filename;destination path\[filename])*
#copyfile(My Documents.*;\Storage Card\My Documents\)*
#copyfile(My Documents.txt;\Storage Card\My Documents\)*
#copyfile(My Documents\list.txt;\Storage Card\My Documents\)
#copyfile(My Documents\list.txt;\Storage Card\My Documents\copyoflist.txt)

#**move** move file or folder *#move(source path\filename;destination path\filename)* or move folder *#move(source path\folder;destination path\folder)*
#move(My Documents.*;\Storage Card\My Documents\)*
#move(My Documents\My Folder;\Storage Card\My Documents\My Folder)
#move(My Documents\list.txt;\Storage Card\My Documents\copyoflist.txt)

#**delete** delete file or folder *#delete(file)*
#delete(My Documents.bak)*
#delete(\Test Folder)

#**led** on/off LED, *#led(0|1|2)*

#**clearqueue** delete all SKScheMa events from system queue
#**profile** replace all events in queue from profile, *#profile(path\filename.sksp)*
#**addprofile** add events to queue from profile, #addprofile(path\filename.sksp)

#**msg** show message, *#msg(Variable A = #A#) #d(12)*
#**writestr** write text to new file, *#writestr(filename.txt) #d(Variable A = #A#)*
#**appendstr** append text to file, *#appendstr(filename.txt) #d(Variable A = #A#)*

#**gotohome** activate home screen
#**gotodialer** activate dialer screen
#**gotostart** activate start menu

#**yesno** #p(Message text) #d(font size) #rmr(0|1|2 - default answer,0-YES-TRUE,1-NO-FALSE,2-UNKNOWN) #rmt(time limit for show this message,0 - no limit), show message, if user answer YES then sysvar == TRUE, by pressed "action" button or after time (#rmt) will be selected answer from #rmr
#yesno(Are you sure?) #rmr(1) #rmt(10000) - show question, if user not answer - after 10 seconds will be selected "No"
#**okcancel** see #yesno
#**gettext** #p(Text) #d(font size) #rmr(0|1|2 - default answer,0-YES-TRUE,1-NO-FALSE,2-UNKNOWN) #rmt(time limit for show this message,0 - no limit) Wait user input. If user enter text and tap OK - sysvar ==

TRUE and SKScheMa put text to variable Var0 , if user tap Cancel - sysvar == FALSE. by pressed "action" button or after time (#rmt) will be selected answer from #rmr
#**setval** set string value to VARx variable, #setval(3) #d(New value for Var3)

#**rgdelval** delete registry value, #rgdelval(HKLM|HKCU|HKCR;path;value name)
#**rgdelkey** delete registry key, #rgdelkey(HKLM|HKCU|HKCR;path;key name)
#**rgset** set registry value, #rgset(HKLM|HKCU|HKCR;path;value name;DWORD|SZ|MZ;value data)
#**rgget** get registry value, #rgget(HKLM|HKCU|HKCR;path;value name;DWORD|SZ|MZ;variable name)
#**regflush** flush registry from ram to storage

#**getsoundvol** get sound volume, #getsoundvol(variable name)
#**getphonevol** get phone sound volume, #getphonevol(variable name)
#**getbattery** get battery level, #getbattery(variable name)
#**waitcusor** #p(ON|off) show/hide wait simbol

#**url2file** download a file, r(#url2file(URL;path on PDA)
r(#url2file(http://s-k-tools.com/SKScheMa/SKScheMa.zip;\My Documents\SKScheMa.zip)

#**filesize** get filesize, #filesize(file name;variable name)
#**crc32** calculate CRC32 for file, #crc32(file name;variable name)
#**getdate** get current date , #getdate(date format;string variable number)
#**gettime** get current time , #gettime(time format;string variable number)
#getdate(dd/MM/yyy;5)
#gettime(hh:mm;6)
#msg(Now @var5 @var6) #d(9)

#**getmute** get "mute" state, #getmute(numeric variable name), very device dependent
#**setmute** set "mute" state, #setmute(ON|OFF|1|0), very device dependent

#**sendkey** (press a key)
The following macros can be used in the line:
Enter (action) **R**
Back **Z**
Left **<**
Right **>**
Up **^**
Down **!**
Hardware buttons (only 15 buttons possible) **A1-AF**
Soft keys (only 2 buttons possible) **K1-K2**
Send (green button) **P1**
End (red button) **P0**
[any string] - copy "any string" to the clipboard.
Other: **1-9**, **# , ***

#puttext put text to current input window
#puttext(This is text)

Sample (send ussd message):
#gotodialer()
*#sendkey(*100#) #sleep(1000)*
#sendkey(P1) #sleep(1000)

#**startsound** play sound in bacground with rules defined in "script". #startsound(sound file;script)
script (like phone script) commands:
a = activate device
cN = set volume to N in percentage max volume

fN = flash notification LED for N seconds
p = play ringtone. Note that this will play the ringtone all the way through before continuing with the next code.
r = repeat. Note that this should be the last code in your Script string, if used at all.
vN = vibrate for N seconds
wN = wait for N seconds. Note that the device will wait this long before continuing with the next code.
#**stopsound** stop sound started only from current script.
#**stopallsounds** stop sound started by any other script(s).

sample (play and vibrate 20 seconds):
#startsound(\Windows\Windows Default.wav;av1pr)
#stopsound() #sleep(20000)

#**wrecord** record wav file. #wrecord(path\file.wav;time (ms.))
record to \My Documents\yymmddhhmm.wav, 6 second
#wrecord)
record to \Storage Card\My Documents\yymmddhhmm.wav, 10 second
#wrecord(\Storage Card\My Documents\;10000)
record to \Storage Card\My Documents\myrecord.wav, 20 second
#wrecord(\Storage Card\My Documents\myrecords.wav;20000)

#**clipget** get text from clipboard #clipget(variable)
#**clipset** put text to clipboard #clipset(text)
#**speakerphone** enable/disable speakerphone mode #speakerphone(ON|1|off|0)

#phoneprofile select phone profile, #phoneprofile(N), where N:
0 — Normal
1 — Silent
2 — Car
3 — Headset
4 — Loud
5 — Meeting
6 — Speakerphone

With all command can be used the parameter **#sleep**(millisecond), this parameter make timeout BEFORE command will run.

Some command example:

Work with registry:

```
#msg(Attempt to get value HKCU\software\sk\testreg\dwvalue)
#rgget(HKCU;software\sk\testreg;dwvalue;DWORD;testdw)
#iftrue(ok)
#msg(Value HKCU\software\sk\testreg\dwvalue not exist)
#label(ok)
#msg(Attempt to set value HKCU\software\sk\testreg\dwvalue -> 1024)
#rgset(HKCU;software\sk\testreg;dwvalue;DWORD;1024)
#msg(Attempt 2 to get value HKCU\software\sk\testreg\dwvalue)
#rgget(HKCU;software\sk\testreg;dwvalue;DWORD;testdw)
#ifunknown(bad)
#iffalse(bad)
#msg(Value HKCU\software\sk\testreg\dwvalue == #testdw#)
#label(bad)
#msg(Attempt to get value HKCU\software\sk\testreg\szvalue)
#rgget(HKCU;software\sk\testreg;szvalue;SZ;var2)
#iftrue(ok1)
#msg(Value HKCU\software\sk\testreg\szvalue not exist)
#label(ok1)
#msg(Attempt to set value HKCU\software\sk\testreg\szvalue)
#rgset(HKCU;software\sk\testreg;szvalue;SZ;This is SZ value)
#msg(Attempt 2 to get value HKCU\software\sk\testreg\szvalue)
#rgget(HKCU;software\sk\testreg;szvalue;SZ;var3)
#ifunknown(bad1)
#iffalse(bad1)
#msg(Value HKCU\software\sk\testreg\szvalue == @var3)
#label(bad1)
#msg(Now need delete key HKCU\software\sk\testreg)
#rgdelkey(HKCU;software\sk;testreg)
```

#gettext sample:

```
#gettext(Test 1) #d(9)
#iffalse(end)
#unknown(end)
#msg(You enter: @var0) #d(9)
#label(end)
```

Variables and pseudovariables (macros)

Macros (pseudovariables) is a record like @Name which is replaced with some value if a script is executed. A script may contain the following macros: @time, @date, @var0, ..., @var9 and other - user defined.

You can use variables in SKScheMa scripts. There are two types of variables - numeric and line (string).
Numeric variables can contain only numbers. They may have arbitrary names. Usage of up to 50 different numeric variables is allowed.
To work with numeric variables, use the #EVAL command.
to set a value: #eval(a=1)
to change a variable value: #eval(a=a+1)
a variable value can be modified into a character string to appear in a message
#msg("A" value = #A#)
for comparing variables #chkcondition is used
Simple comparison sample
#chkcondition(a!=4)
#IFTRUE(start)
comparison of the current date and the specified date
#eval(data=@Date(yyyyMMdd)) #chkcondition(data>=20060701)

There can only be 50 string variables (macros) altogether.
To access those, @VarName macro are used.
Exist 10 system variables with names @Var0 - @Var9.
Every variable may contain a string of up to 256 symbols.
Variables can be set in the command line #VARx(text) (x from 0 to 9), using command #setvar - (#setvar(variable name|x) #d(text)) (x from 0 to 9) and/or using command #rgget.

Script execution sequence control

Commmand #LABEL defines a label. Label is a script line associated with a certain name. This name always makes it possible to find the line. When two identical labels exist, the one which is closer to the beginning of the script is always found.
#LABEL(label name)

Command #GOTO causes a control transfer to the script line defined by a certain label.
#GOTO(label name)

Command #EVAL allow you to set/change a variables.
example:
#eval(a=1)
#label(start)
#msg("A" value = #A#)
#eval(a=a+1)
#chkcondition(a!=4)
#IFTRUE(start)

The following check-up commands can be used in a script:

radio subsystems status	#chkwlan, #chkbt, #chkgsm
files or folders existence	#chkfile, #chkfolder
SKScheMa start status (true if SKScheMa event is reason of power on device)	#chkstart
Device power on AC line? (true for yes, false for not, unknown if unknown state)	#chkac
SIP status	#chksip
String1 include any substring from String2? (result - TRUE) Or String1 is substring of String2? (result FALSE) Othervise - result UNKNOWN String2 can include separator ";" #chkstrings(String1) #d(String2) #chkstrings(12345) #d(1234;aaaaaa;bbbbbbb) - result TRUE #chkstrings(aaaaaaaaaa) #d(1234;aaaaaa;bbbbbbb) - result TRUE #chkstrings(aaa) #d(1234;aaaaaa;bbbbbbb) - result FALSE #chkstrings(ccccc) #d(1234;aaaaaa;bbbbbbb) - result UNKNOWN	#chkstrings

You can use #chkcondition for check variables:
#chkcondition(a==10)
You can use #chkconnection for check connection:
#chkconnection(`USB)
You can use #chkprocess for check process:
#chkprocess(tmail.exe)

After these commands have been executed, a special internal variable (for script his name is SYSVAR and possible this: #eval(myvar=sysvar))
takes one of the 3 status: TRUE (on, active, existing), FALSE (off, non-existing), UNKNOWN (not possible to define a status). This variable keeps its status until the following working (not management control one) command is executed. Commands which do not change their status TRUE-FALSE always return the variable into UNKNOWN.

Window search command #findwindow sets this variable value as well depending upon search results.
Files-operation commands like #copyfile, #move, #delete sets this variable value as well depending upon operation results.

Commands #IFTRUE, #IFFALSE, #IFUNKNOWN, #IF cause a control transfer to a label depending upon a status of the internal variable.
#IFTRUE(label name)
#IFFALSE(label name)
#IFUNKNOWN(label name)
#IF(label name for TRUE;label name for FALSE;label name for UNKNOWN)

Sample (file exist or not):

```
#chkfile(\windows\pword.exe)
#ifunknown(unk2)
#iftrue(on3)
#msg(File not exist)
#goto(end3)
#label(on3)
#msg(File Exist)
#goto(end3)
#label(unk2)
#msg(Cannot check file!)
#label(end3)
```

Work with strings (text)

To compare variables or strings, use the #chkstrings command
#chkstrings(@var5) #d(123456;aaaaaa;bbbbbbb)

For get string variable length use #length command:
#length(@variable name|text;numeric variable name)

For get substring position use #pos command:
#pos(substring;string;numeric variable name)

For extract substring from string use #substr command:
#substr(string;position;length;result string variable name)

Sample (#length, #pos, #substr):

```
#setvar(Variable) #d(0123456789)
#length(@Variable;len)
#pos(@Variable;345;pos)
#eval(len1=len-pos+1)
#substr(@Variable;#pos#;#len1#;NewVariable)
#eval(res=SYSVAR)
#substr(@Variable;#pos#;3;NewVariable2)
#eval(res2=SYSVAR)
#msg(String @Variable#RLength - #len# #R "345" position - #pos# #RResult #res# #res2# #RSubstring1
@NewVariable#RSubstring2 @NewVariable2)
```

Sample (adding string to string):

```
#setvar(Variable1) #d(01234)
#setvar(Variable2) #d(56789)
#setvar(Variable3) #d(@Variable1@Variable2)
#msg(@Variable3)
```

Working with ZIP archives

SKScheMa allows to perform basic operations with zip-archives

To create a new, empty archive
#zcreate(archive name) #d(password)
A password is optional.
If there existed a file with the same name - it will be deleted.

To open an existing archive
#zopen(archive name) #d(password)
A password is optional.

To close
#zclose()
closes a new or open archive.

In one script, you can only work with one archive at any point of time.

A working cycle is only limited with commands between #zopen/#zcreate and #zclose.
For creating new archives you can use the command
#zadd(\path\mask)
to add files into an archive.
After an archive is closed (#zclose() you will not be able to modify its contents with SKScheMa, use other programs for this.
To define how to add files, the following commands are used
#zusepath(ON|off)
this command defines if it is required to save the full path in the name of the file to be added
#zsetrecursion(ON|off)
this command defines if it is required to search in subfolders during a file search.

For existing archives the command
#zextract(\path\mask)
extracts files into the folder which had been set earlier with the command
#zsetextractpath(extract path)
The command
#zusepath(ON|off)
defines if it is required to extract files according to the full path in its name

#zusepath(0|1|STD|UTF8)
defines mode for save/restore file names in zip archives

How to create a new archive - example
```
#zusepath(ON)
#zsetrecursion(ON)
#waitcursor(ON)
#zcreate(\Storage Card\MyZip.zip)
#zadd(\My Documents\*.txt)
#zadd(\My Documents\*.pxl)
#zadd(\My Documents\*.doc)
#zadd(\My Documents\*.bmp)
#zclose()
#waitcursor(OFF)
```

How to extract archive - example
```
#zusepath(ON)
#zsetextractpath(\Storage Card\Temp\)
#waitcursor(ON)
#zopen(\Storage Card\MyZip.zip)
#zextract(*.*)
#zclose()
#waitcursor(OFF)
```

Profiles

This is a list of the events which will not occur but have been saved for further tasks. All elements of this list are stored in files with the extension *.sksp. It may be convenient if, for example, you use one set of events at work and the other one – during your vacation. Both a profile and a separate even from this profile can be added to Queue.

Support

Please note: some functions device dependent and, possible, can not work on your device.
If you have a problem or you have any suggestions, please feel free to contact us: support@s-k-tools.com

E-mail: support@s-k-tools.com
http://www.s-k-tools.com