

RESONATOR STATUS

2.1 THz



SKINNER

LUA BASED GRAPHICS ENGINE

Ver 03.04.00 - 27 Dicembre 2018



57° / 69°



62°



[ABILENE TEXAS, UNITED STATES]

Introduzione.....	5
Descrizione generale	6
Le Basi.....	7
Caricamento App	8
Autoexec.lua.....	10
UI.lua	11
Thema.lua	13
skncfg.lua	15
sknCss.lua	16
splash.lua	17
loadall.lua	18
SCRIPT Loading	21
Creazione Window	22
Creazione PlugIn.....	23
Spiego i metodi	24
create delete.....	24
show hide	24
enable.....	25
modal.....	25
goto	26
showOnlyMe.....	26
bringTop bringBot	27
color bgcolor.....	27
size	28

pos	28
transp	29
allowMoveX/Y	29
allowResize	30
autoHide	30
blink	31
ani	31
Le classi	32
System	33
Desktop	49
Collection	57
Multimedia	71
Window	74
Label	96
Button	117
iButton	142
ITbutton	165
Image	187
Animated Image	204
CheckBox	218
Progress Bar	237
Multi Image	252
Keyboard	255
Graph	264

Gauge.....	284
Pie.....	299
Draw (NON ATTIVO)	315
Debug	325
CLASSNAME template.....	331

INTRODUZIONE

Dopo tanti anni di programmazione "C" su interfacce grafiche avanzate, di problemi con handle, allocazione, eventi e molto altro, ho pensato ad una soluzione disegnata per chi non conosce nel dettaglio una GUI e che comunque non vuole incorrere in tutte quelle difficoltà note a chi sviluppa Interfacce Utente nei sistemi Embedded.

La scelta di Skinner è quella di avere uno scripting language per la gestione dell'interfaccia grafica e degli eventi ad essa collegati. Nella precedente versione veniva utilizzato un linguaggio proprietario e la componente di scripting configurava l'interfaccia ma necessitava di molto codice "C" a supporto.

Questa versione si avvale del linguaggio LUA e di conseguenza permette di integrare negli script l'intera gestione della GUI.

Una console sempre disponibile consente di interagire in Real-Time con l'interfaccia e di esercitare metodi anche senza scrivere codice ad Hoc.

Questa soluzione consente di modificare l'interfaccia utente senza ricompilare il binario del sistema Embedded, ma di modificare solo il contenuto degli script e immagini presenti sul disco.

**"100 Persone che ogni giorno studiano
e realizzano Servizi e Soluzioni Web su misura per Te"**

CEDAC SOFTWARE s.r.l.

Raul Mungai



MADE IN ITALY.

DESCRIZIONE GENERALE

Skinner consente di creare interfacce grafiche tramite di script LUA.
Skinner rende semplice la costruzione di sofisticate interfacce grafiche senza specifiche competenze di programmazione.

Tramite il linguaggio LUA è possibile implementare la quasi totalità dell'interfaccia senza dover ricorrere alla parte "C".

Skinner consente di creare applicazioni efficienti e di semplice manutenzione.

GUI

La GUI si basa sulla rappresentazione di oggetti (es: Buttons, Label, etc) piazzate all'interno di una Window che ne controlla la presentazione.

Ogni oggetto deve avere una Window che lo contiene e ne controlla la visualizzazione e i relativi eventi. La vista degli oggetti contenuti in una Window dipendono dalla stessa (se viene nascosta la Window, si nascondono anche tutti gli oggetti in essa contenuti)

Ogni Oggetto è creato utilizzando un nome di riferimento (univoco a livello globale), la visibilità degli Oggetti è globale e non legata alla Window di appartenenza.

Qualche consiglio

Per ricercare una specifica Classe è sufficiente digitare la Classe all'interno delle parentesi quadre (es: [sknSys]) per raggiungere la tabella dei membri della classe.

LE BASI

La programmazione e il controllo di Skinner avviene tramite Classi LUA.

Ogni Classe implementa una funzionalità o uno specifico componente, alcune Classi permettono di configurare l'ambiente Skinner, altre ne controllano le funzionalità comuni.

Ogni Classe ha metodi che governano le specificità, sono normalmente sempre presenti tutta una serie di metodi che consentono di:

- Visualizzare / Nascondere
- Aggregare a specifici gruppi
- Gestire il Blink
- Gestire l'Auto Hide
- Gestire le Animazioni
- Valori Custom
- Gestire le Callbacks
- Gestire l'ordine Top-Bot

I principi di base sono quelli di una GUI basata su Finestre ed eventi, è possibile creare e distruggere i componenti in Run-Time, i componenti si identificano tramite una stringa (di lunghezza finita secondo quanto previsto dal BSP).

Ogni modulo carica le risorse necessarie (fonts, immagini, etc) per rendere più atomico lo sviluppo, se una determinata risorsa è già presente verrà ignorata la richiesta di caricamento.

Questa soluzione evita di avere riferimenti incrociati fra scripts e risorse, di difficile lettura e altrettanto complessa manutenzione.

Ogni modulo espone un oggetto con una serie di membri, questo permette di avere un codice più chiaro e più semplice da manutenere.

Segue una semplice descrizione dei passaggi necessari allo sviluppo di una applicazione tramite Skinner.

CARICAMENTO APP

Dopo aver avviato i servizi di sistema, una volta arrivato nel processo utente principale, è necessario caricare gli script LUA necessari all'applicazione.

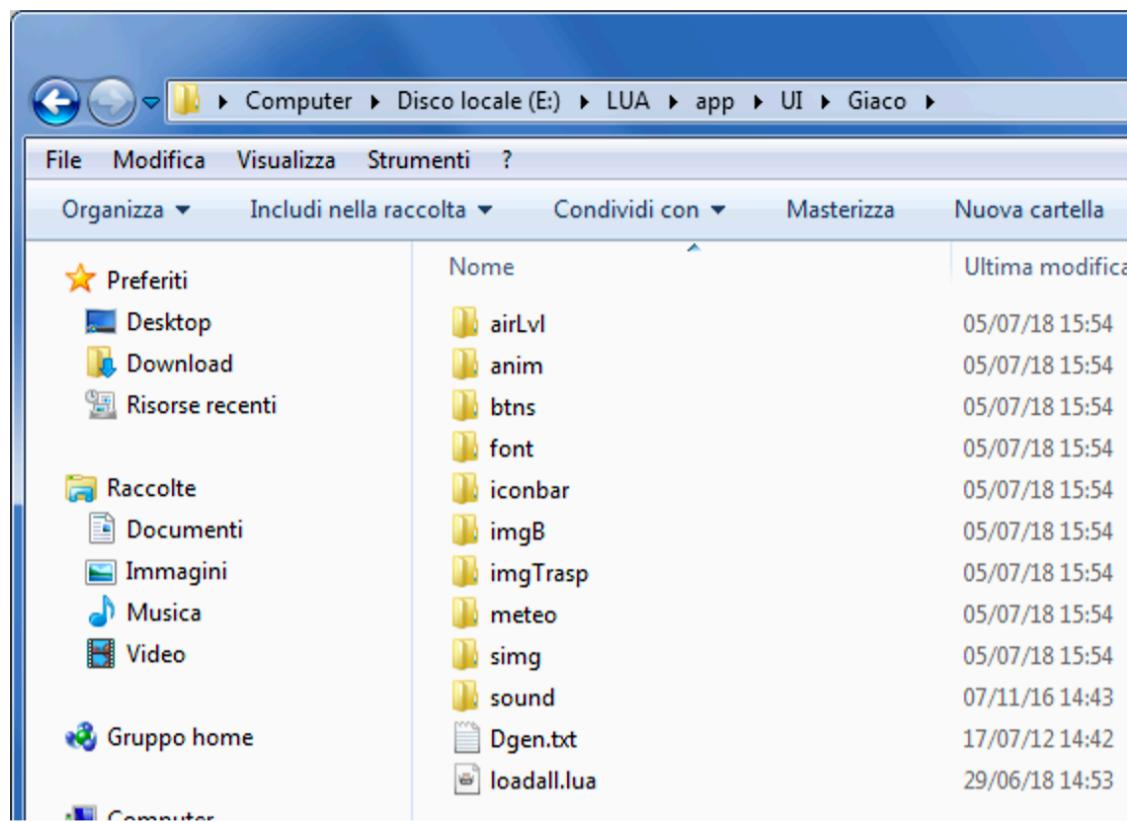
Nella componente "C" verranno implementate le interfacce LUA necessarie all'applicazione e poi registrate tramite un'apposita funzione di sistema (vedi codice di esempio del LoaderApp).

Il LoaderApp tenta di eseguire il file "\LUA\autoexec.lua" per avviare il caricamento dell'applicazione.

Segue una convenzione, non mandatoria, che prevede la seguente struttura:

			note
LUA			
	autoexec		Directory che contiene il codice di avvio
		autoexec.lua	eseguibile per avvio App
	plugin		Funzionalità specifiche
	win		Tutte le Window dell'applicativo
		splash.lua (ex)	Splash screen
		w_home.lua (ex)	Home
		w_cfg.lua (ex)	Configurazione
		...	
	app		Directory che contiene la componente applicativa
		skncfg.lua	Configurazione ambiente
		skncss.lua	Definizione dello stile
		thema.lua	Definizione del Thema
		UI.lua	User Interface
	UI		
		F32B_1_S.txt	Font per Splash screen
		spl0ITA.bmp (ex)	Background image per splash screen
	UI\Giaco		Thema directory (ex)

Segue un esempio puramente indicativo della directory di Thema":



I files Dgen.txt (Dizionario) e loadall.lua (carica tutti i plugin e tutte le finestre) , mette il contenuto delle sotto-directory contengono i vari artefatti grafici (immagini, icone, fonts, etc).

Di fatto è possibile disegnare un'interfaccia che è in grado di selezionare un Thema influenzato da immagini, fonts e colori senza dover modificare gli script applicativi.

AUTOEXEC.LUA

Questo file viene chiamato dal Loder App per avviare il caricamento dell'intera applicazione. In questa componente viene normalmente selezionato lo standard Output e richiamato il successivo script LUA che configura la GUI e in seguito eventuali altre componenti applicative (nell'esempio segue il CCC per la comunicazione Wireless).

Code example:

```
-- Select Virtual device
-- nut.setVdev ("ser", 1, 115200)

--[ [
-- Wait for USB connect
for t=1, 4 do
    io.write ("*")
    os.sleep(1000)
end
-- ] ]

-- true=load artefact and windows creation,
-- false=load management method only

-- Load User Interface
dofile ("\LUA\app\skn\UI.lua")

-- Config and start CCC
-- dofile ("\LUA\app\ccc\ccc.lua")

-- nut cmdline("servSetNum CCC sys\debugLevel 0")
```

UI.LUA

Questo script carica tutte le componenti applicative necessarie.

Nella prima parte è visibile la sezione dove vengono allocate le collezioni di immagini che conterranno le varie immagini utili all'applicazione. E' necessario dimensionare correttamente tali collezioni in modo che possano contenere un adeguato numero di immagini e con una dimensione totale utile a contenere tutte le immagini in modalità raw.

A seguire vengono caricati i componenti che si occupano:

- Thema
- Configurazione
- Stile
- Splash Screen form
- tutte le finestre e plugin (loadall.lua)

Nel caricamento del file "loadall.lua" viene impiegato il riferimento ad una specifica directory tramite il metodo "**thema.getDir()**" che consente di gestire la directory di Thema in modo automatico.

Code example:

```
-- Load User Interface

-- Config package resources
-- Button images
sknColl.addImgColl ("btn", 100, 2000000)
sknColl.addImgColl ("btnP", 100, 2000000)
sknColl.addImgColl ("btnd", 100, 2000000)

-- Button images
sknColl.addImgColl ("icon", 50, 1000000)

-- Large images
sknColl.addImgColl ("BG", 10, 2000000)
sknColl.addImgColl ("IMG", 20, 1000000)

-- Load function utilities
dofile ("\\LUA\\app\\skn\\thema.lua")      -- Thema management

-- Load UI basics modules
dofile ("\\LUA\\app\\skn\\skncfg.lua")      -- Skinner Config

dofile ("\\LUA\\app\\skn\\skncss.lua")        -- CSS functions

-- Splash Form
dofile ("\\LUA\\app\\skn\\win\\splash.lua")

-- Load All UI elements
dofile (thema.getDir() .. "\\loadall.lua")
```

THEMA.LUA

Questo script da il supporto al Thema, implementa tutte le funzioni necessarie a creare un layer che gestisce il Thema senza dover personalizzare il codice a livello più alto.

Code example:

```
-- Global Object
thema = {}

-- Thema management (configured by skncfg.lua)
thema.language = "ITA"
thema.thema = ""                      -- thema of application
thema.motivo = ""                      -- Motivo
thema.workDisk = ""                     -- working disk
thema.white = true                     -- use white text

-- ** Global layout vars **

-- icons:
thema.iconSpacyng = 40
thema.iconFont = "F16B"
thema.iconLabelFont = "F16B"

-- Effects
thema.effectON = true
thema.effectMoveStep = 20
thema.effectMoveIterations = 10

-- Look
thema.textColor = sknSys.getColor("black")
thema.bg_color = sknSys.getColor("white")
...
thema.bg_colorPopUp_bar = sknSys.getColor("blue")
thema.bg_colorStat = sknSys.getColor("lightgray")

function thema.getDir()
    return thema.workDisk .. thema.thema
end

function thema.loadDict()
    sknColl.dictLoad ("GEN", thema.language, thema.getDir() .. "\\DGen.txt")
end

...
```


SKNCFG.LUA

Questo script configura l'ambiente applicativo.

Viene avviato Skinner, aggiunto un Dizionario dimensionato per l'applicazione, poi caricato con la lingua di default.

Configurato il Thema (si potrebbe caricare da file questa configurazione per renderla modificabile dalla componente applicativa).

Configurato Skinner per la componente di animazione, auto hide e gesture, rimandando alla componente applicativa come reagire alle gesture.

Code example:

```
-- Skinner config
sknSys.start(false)

sknColl.addDict ("GEN", 400, 300000)

-- Set language, working disk and Thema
--sknColl.dictLoad ("GEN", "ITA", "mmc:\skn\DAppl.txt")
thema.thema = "Giacomo" -- "C102",
thema.motivo = "3"
thema.whiteText(true)
thema.language="ITA"
thema.workDisk = "\LUA\app\UI\
sknSys.tplongTouch (true, 1500, 8000) -- Long touch enabled

-- Animation
sknDesk.anicfg(10, 50, "shift")
sknDesk.autohidecfg(10, 50)
sknSys.tpMove ("gestureUp", "gestureDown", "", "")
```

SKNCSS.LUA

Questo script definisce lo stile.

In questo modulo vengono definite le funzioni per creare Buttons e per configurarne le immagini oltre a Window e piccole utilità analoghe.

Code example:

```
-- Global Object
css = {}

-- Button : new
function css.button_new (name, winName, x, y, image, event, show)
    sknITbtn.create (name, winName, x, y, 0, 0)
    thema.itbtn_set_imgs(name, image)
    sknITbtn.keyReleaseCB (name, event)
    . . .
end

-- Window : new
function css.window_new (winName, show)
    sknWin.create (winName, 0, 0, sknSys.screenXsize(), sknSys.screenYsize())
    . . .
end

-- set button images (png)
function css.loadImgMulti_png (fnameNoExt, name)
    if sknColl.imgSize("btn", name) == 0 then
        sknColl.imgAdd ("btn", thema.getDir() .. fnameNoExt .. ".png", name,
false, true)
        sknColl.imgAdd ("btnP", thema.getDir() .. fnameNoExt .. "_.png", name,
false, true)
        sknColl.imgAdd ("btnd", thema.getDir() .. fnameNoExt .. "-.png", name,
false, true)
    end
end

-- set button images (bmp)
function css.loadImgMulti_bmp (fnameNoExt, name)
    . . .
end

-- set icon images (png)
function css.loadIcon_png (fnameNoExt, name)
    . . .
end

. . .
. . .

-- set small images (bmp)
function css.loadSmallImg_bmp (fnameExt, name)
    . . .
end
```

SPLASH.LUA

Questo script è puramente applicativo e normalmente viene caricato per creare una schermata di avvio che consente di monitorare il caricamento del programma.

Code example:

```
-- Mount fonts
css.font_mount (thema.workDisk .. "\F32B_1_S.txt", "F32B")
-- Background image
if thema.language == "ITA" then
    sknColl.imgAdd ("BG", thema.workDisk .. "spl0ITA.bmp", "splash", false, false)
elseif thema.language == "ENG" then
    .
.
else
    -- undefined
    sknColl.imgAdd ("BG", thema.workDisk .. "spldef.bmp", "splash", false, false)
end

-- Global Object
win_splash = {}

-- Splash screen
function win_splash.create()
    sknWin.create ("splash", 0, 0, sknSys.screenXsize(), sknSys.screenYsize())
    sknWin.bgcolor ("splash", sknSys.getColor("white"))

    ix,iy = sknWin.size("splash")
    sknImg.create ("bgimage", "splash", 0, 0, ix, iy)
    sknImg.image("bgimage", "BG", "splash")
    sknImg.show("bgimage")
    -- Progress bar
    local xwin, ywin = sknWin.size("splash")
    sknProgbar.create ("loadPb", "bgimage", 0, ywin-13, xwin, 13)
    sknProgbar.setlimits ("loadPb", 0, 100, 0)
    sknProgbar.show("loadPb")
    sknWin.show ("splash")
end

function win_splash.destroy()
    --sknLbl.delete("lab1")
    sknImg.delete ("bgimage")
    sknProgbar.delete ("loadPb")
    sknWin.delete ("splash")
end
function win_splash.loadProgress(progValue)
    if __loadArtefact == true then
        sknProgbar.setvalue ("loadPb", progValue)
    end
end

-- Splash screen display
win_splash.create()
```

LOADALL.LUA

Questo script carica tutti i moduli applicativi, plug-in e Window.

Esaminando lo script è evidente che non viene terminato la Splash Screen e non viene avviata la Window principale, cosa che farà l'ultimo modulo di caricamento, infatti tutte le operazioni di caricamento vengono ripetute per ogni processo legato alla GUI e solo l'ultimo (il caricamento del processo di background) chiuderà la Splash Screen e renderà visibile la finestra di Home.

Code example:

```
-- Load plugins
dofile ("\\LUA\\app\\skn\\plugin\\meteo.lua")
win_splash.loadProgress (5)

dofile ("\\LUA\\app\\skn\\plugin\\button.lua")
win_splash.loadProgress (7)

-- Load all windows
dofile ("\\LUA\\app\\skn\\win\\w_home.lua")
win_splash.loadProgress (10)

dofile ("\\LUA\\app\\skn\\win\\w_cfg.lua")
win_splash.loadProgress (50)

dofile ("\\LUA\\app\\skn\\win\\w_gtemp.lua")
win_splash.loadProgress (60)

dofile ("\\LUA\\app\\skn\\win\\w_curHL.lua")
win_splash.loadProgress (70)

dofile ("\\LUA\\app\\skn\\win\\w_meteo.lua")
win_splash.loadProgress (75)

dofile ("\\LUA\\app\\skn\\win\\w_tempSet.lua")
win_splash.loadProgress (80)

-- Execution
win_splash.loadProgress (90)

-- Load Dict GEN
thema.loadDict()
win_splash.loadProgress (100)
```



```

static void _LUA_load_script_all (BOOLEAN shell)           // TRUE=shell
active
{
CHAR * argv[ 5 ];
. . .

//Load Application script
argv[ 0 ] = (CHAR *) "LUA";
argv[ 1 ] = (CHAR *) "-v";
argv[ 2 ] = (CHAR *) "-e_U=false";
argv[ 3 ] = (CHAR *) "\\LUA\\autoexec\\autoexec.lua";
argc = 4L;
CLI_suspendService(TRUE);
LUAsession = LUA_shell (argc, argv, FALSE, NULL, FALSE, NULL);
CLI_suspendService(FALSE);
app_load_display (LUAsession);                         // Register Application functions

if (shell == TRUE)
{
    argv[ 0 ] = (CHAR *) "-v";
    argv[ 1 ] = (CHAR *) "-E";
    argv[ 2 ] = NULL;
    argv[ 3 ] = NULL;
    argc = 2L;
    CLI_suspendService(TRUE);
    LUAsession = LUA_shell (argc, argv, FALSE, LUAsession, FALSE, NULL);
    (void) TERMU_printString((CHAR*) "\r\nEnd LUA session\r\n");
    CLI_suspendService(FALSE);
}
return;
}

```

SCRIPT LOADING

Come descritto in precedenza, è necessario caricare gli script LUA ed eseguirli.

La componente "C" richiama il caricamento tramite appositi scripts collocati nella directory "LUA\\autoexec" in seguito descritti.

Come primo caricamento viene eseguito direttamente "autoexec.lua" (shell e lato applicativo), a seguire le componenti relative ai servizi di Skinner.

Ecco un esempio del codice "C" necessario al caricamento (normalmente integrato nell'applicativo Embedded).

Code example:

Nel codice "C" dopo aver creato i vari LUAspecte (sessioni LUA indipendenti) viene richiamata la funzione "**app_load_display (LUAtmp);**" che rende visibili a quella specifica istanza LUA le funzioni esposte dalla componente Embedded.

CREAZIONE WINDOW

In questa sezione viene fatto un esempio di creazione di una Window.

Code example:

```
-- Load artworks
-----
css.loadImgMulti_png ("\\bt�s\\chart", "chart")
css.loadImgMulti_png ("\\bt�s\\buttonRect", "buttonRect")

css.loadImgMulti_png ("\\simg\\hum", "hum")
css.loadImgMulti_png ("\\simg\\termo", "temp")
css.loadImgMulti_png ("\\simg\\airq", "airq")
css.loadImgMulti_png ("\\simg\\co2", "co2")

css.loadImgMulti_png ("\\bt�s\\cfg", "cfg")

-- Background image
if thema.motivo == "1" then
  css.loadImg_bmp ("\\imgB\\schermo_rotto.bmp", "BGI")
elseif thema.motivo == "2" then
  css.loadImg_bmp ("\\imgB\\acquario.bmp", "BGI")
elseif thema.motivo == "3" then
  css.loadImg_bmp ("\\imgB\\appartamento.bmp", "BGI")
else
  css.loadImg_bmp ("\\imgB\\Giacco.bmp", "BGI")
end
```

CREAZIONE PLUGIN

In questa sezione viene fatto un esempio di creazione di un PlugIn Meteo.

Code example:

```
-- Global Object
meteo = {}

----- METEO -----

-- set meteo image
function meteo.picture (name, meteoImg)
    sknIbtn.imgpress (name, 0, "icon", meteoImg)
    sknIbtn.imgunpress (name, 0, "icon", meteoImg)
    sknIbtn.imgdisabled (name, 0, "icon", meteoImg)
    local keyX, keyY = sknIbtn.size(name)
    return keyX, keyY
end

-- Special Buttons: new (multi format)
function meteo.new (name, winName, x, y, meteoImg, event, show)
    sknIbtn.create (name, winName, x, y, 0, 0)
    local keyX, keyY = meteo.picture(name, meteoImg)
    sknITbtn.keyReleaseCB (name, event)
    if show == true then
        sknIbtn.show (name)
    end
    return keyX, keyY
end

function meteo.show (name)
    sknImg.show(name)
end

function meteo.hide (name)
    sknImg.hide(name)
end

function meteo.destroy (name)
    sknImg.delete(name)
end

-- Load artworks
-----
-- css.loadImgMulti_png ("\\bt�s\\back", "back")

-- icons
css.loadIcon_png ("\\meteo\\meteo_piaoggia", "pioggia")
css.loadIcon_png ("\\meteo\\meteo_sole", "sole")
css.loadIcon_png ("\\meteo\\meteo_soleggiato", "soleggiato")
css.loadIcon_png ("\\meteo\\meteo_temporali", "temporali")

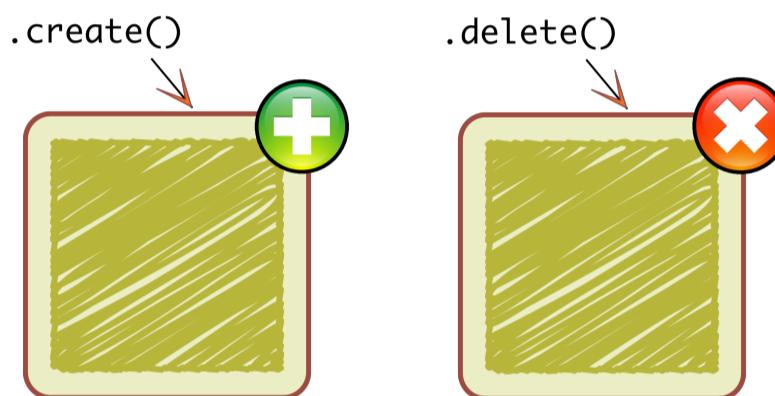
-- Fonts
--css.font_mount (thema.getDir() .. "\\font\\F16B_1_S.txt", "F16B")
```

SPIEGO I METODI

In questo capitolo vengono rappresentati alcuni metodi comuni a quasi tutte le Classi, una descrizione grafica è di supporto alla specifica documentazione allegata ad ogni metodo.

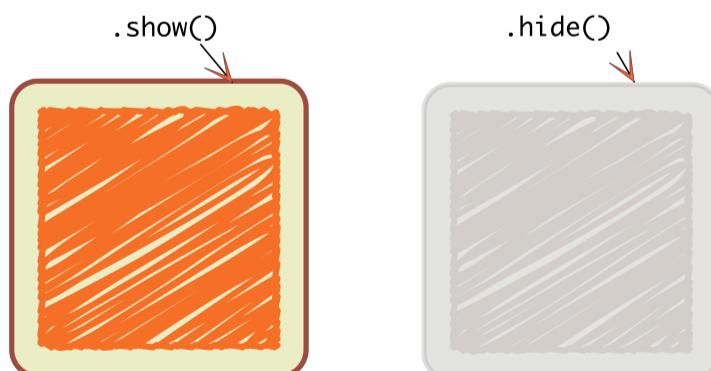
CREATE DELETE

Questi due metodi consentono di creare un oggetto o di distruggerlo.



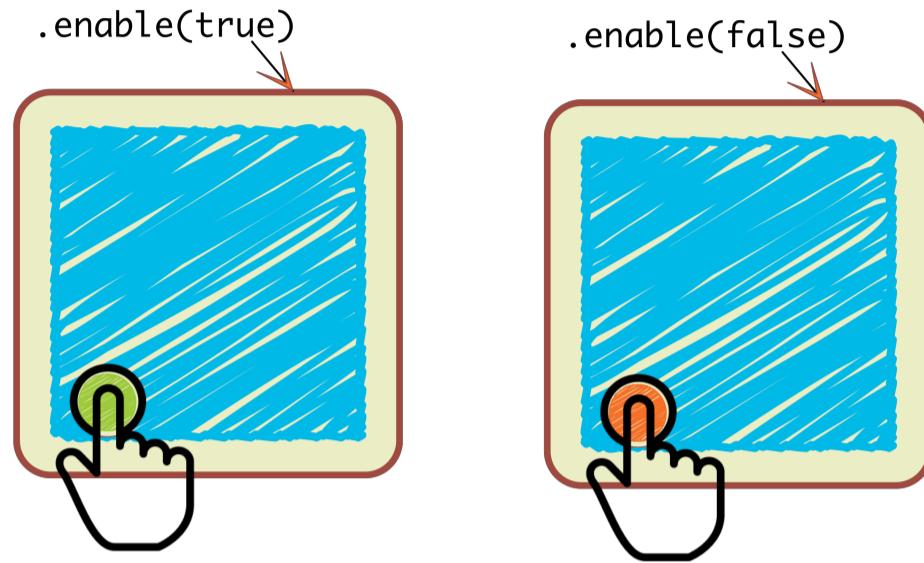
SHOW HIDE

Questi due metodi permettono di rendere visibile e operativo un oggetto oppure nasconderlo e renderlo non operativo.



ENABLE

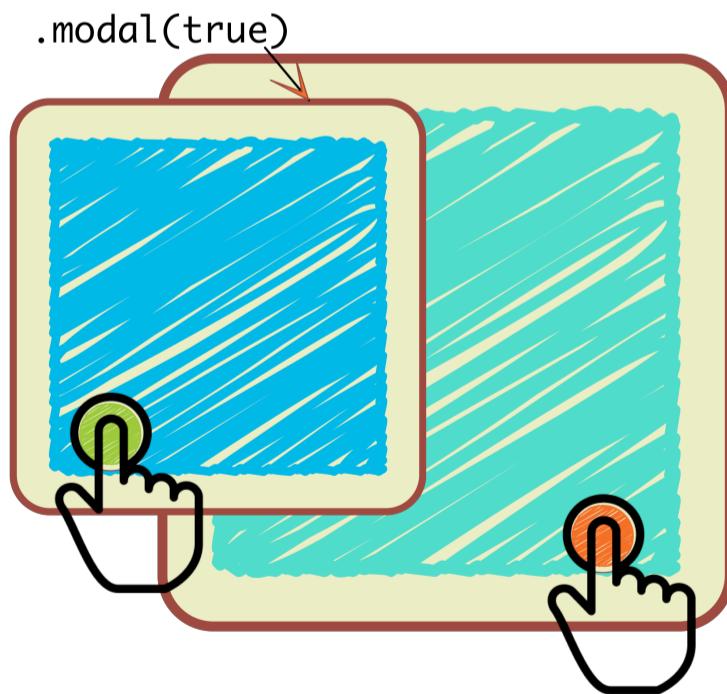
Questo metodo consente di abilitare o disabilitare un oggetto, una volta disabilitato non cambia la presentazione dello stesso, ma non risponde più agli eventi dell'utente.



MODAL

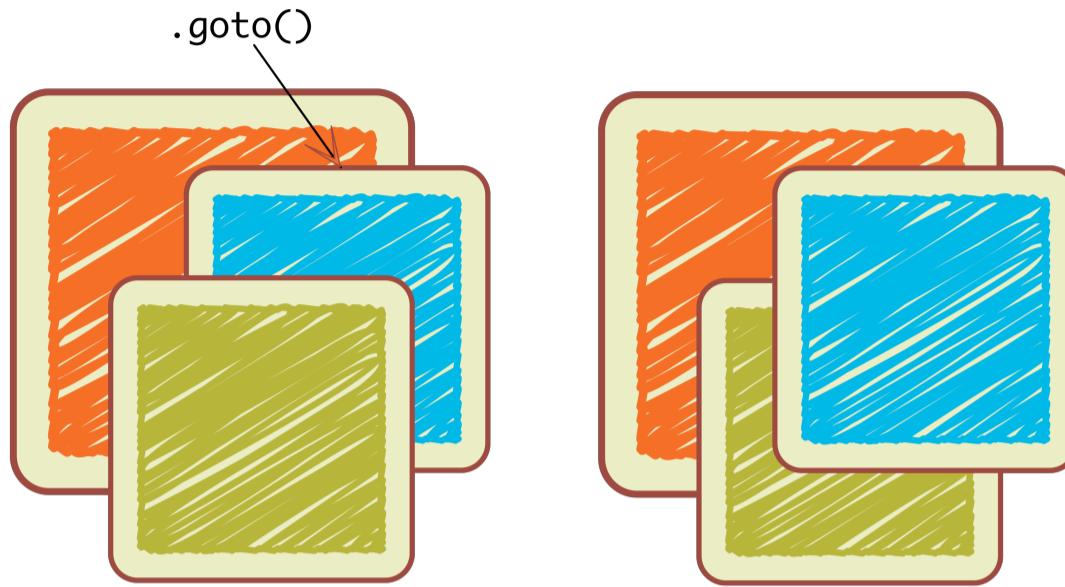
Questo metodo consente di rendere modale la finestra, in questa modalità non è possibile "toccare" le altre finestre anche se visibili.

Questo metodo è di norma disponibile solo per le Window.



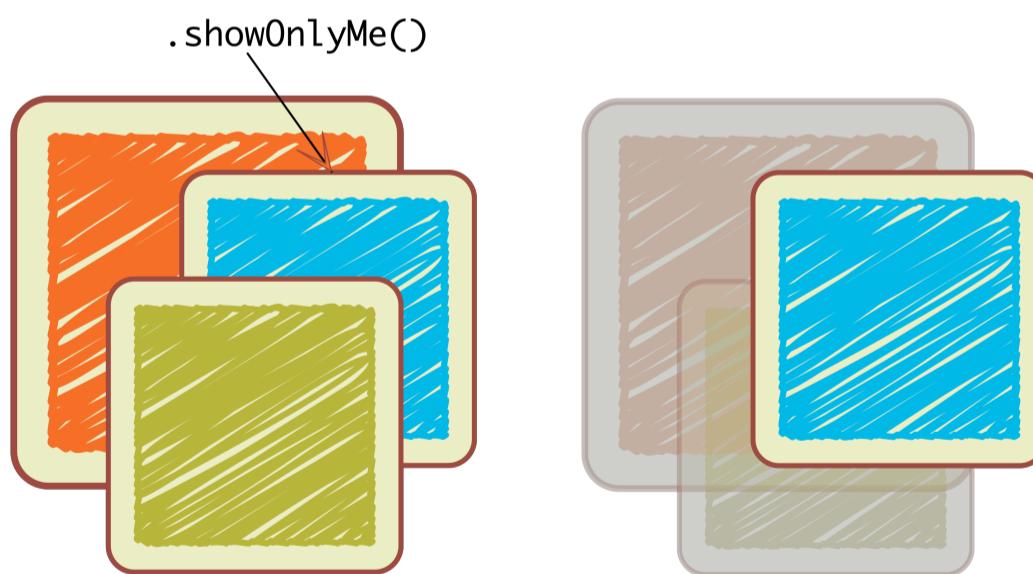
GOTO

Questo metodo porta in primo piano una finestra lasciando comunque visibili le altre.



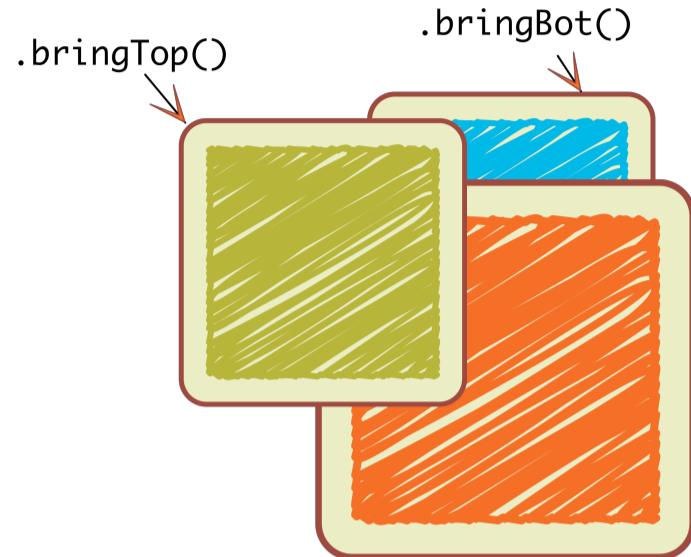
SHOWONLYME

Questo metodo visualizza una finestra nascondendo tutte le altre attualmente visibili.



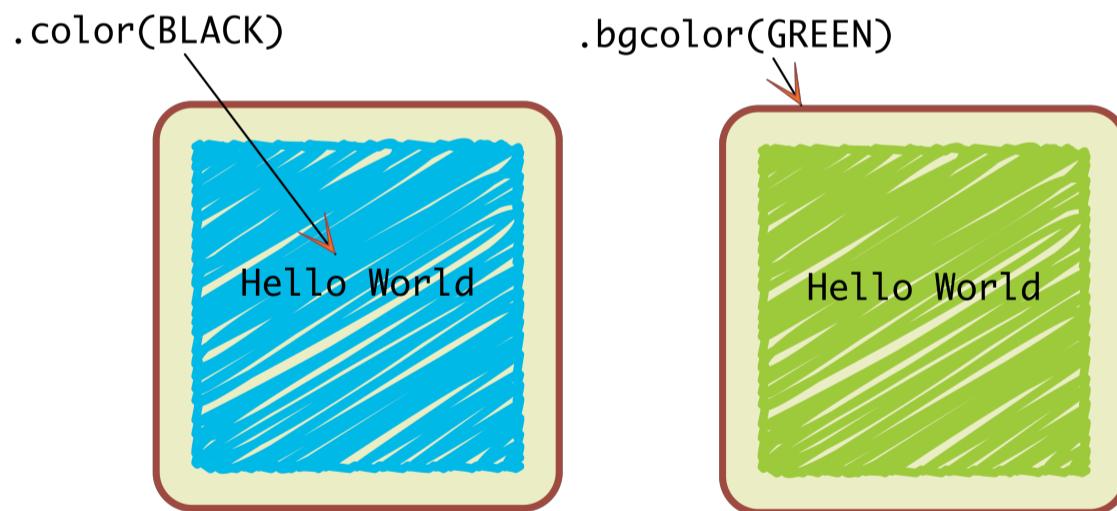
BRINGTOP BRINGBOT

Questi due metodi consentono di portare l'oggetto in primo piano o in fondo.



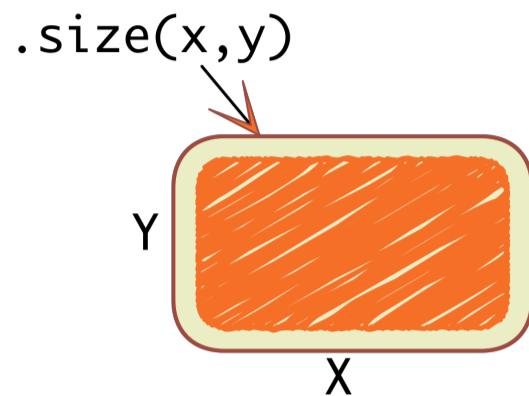
COLOR BGCOLOR

Questo metodo controlla il colore di sfondo e di primo piano di una window.



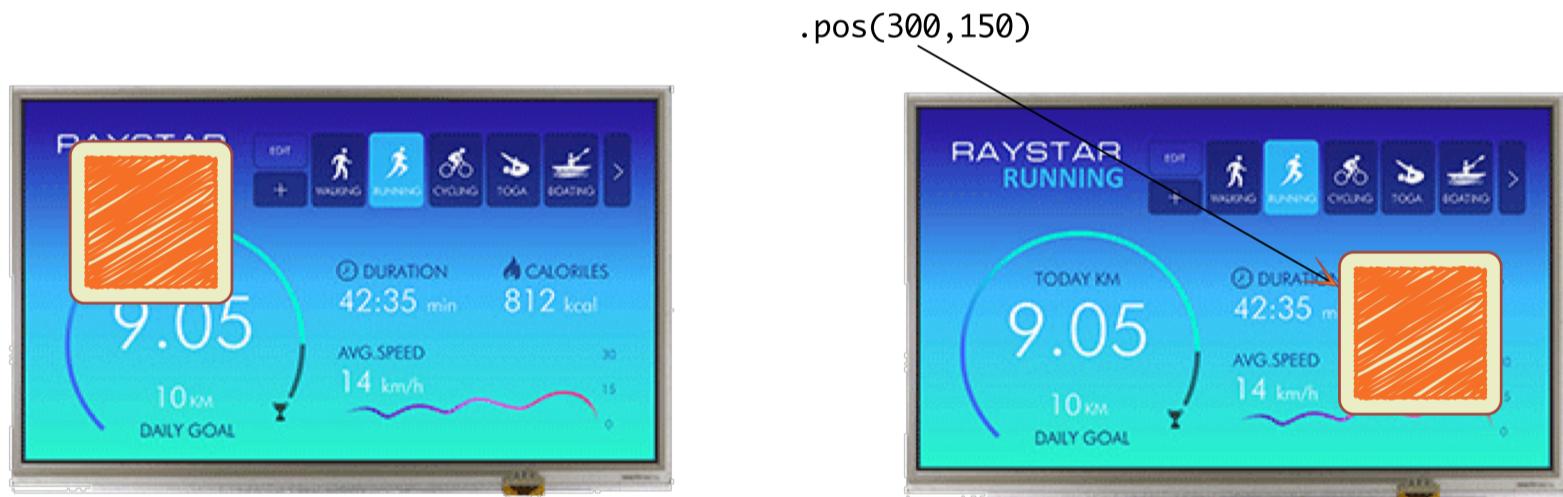
S I Z E

Questo metodo controlla la dimensione di un oggetto.



P O S

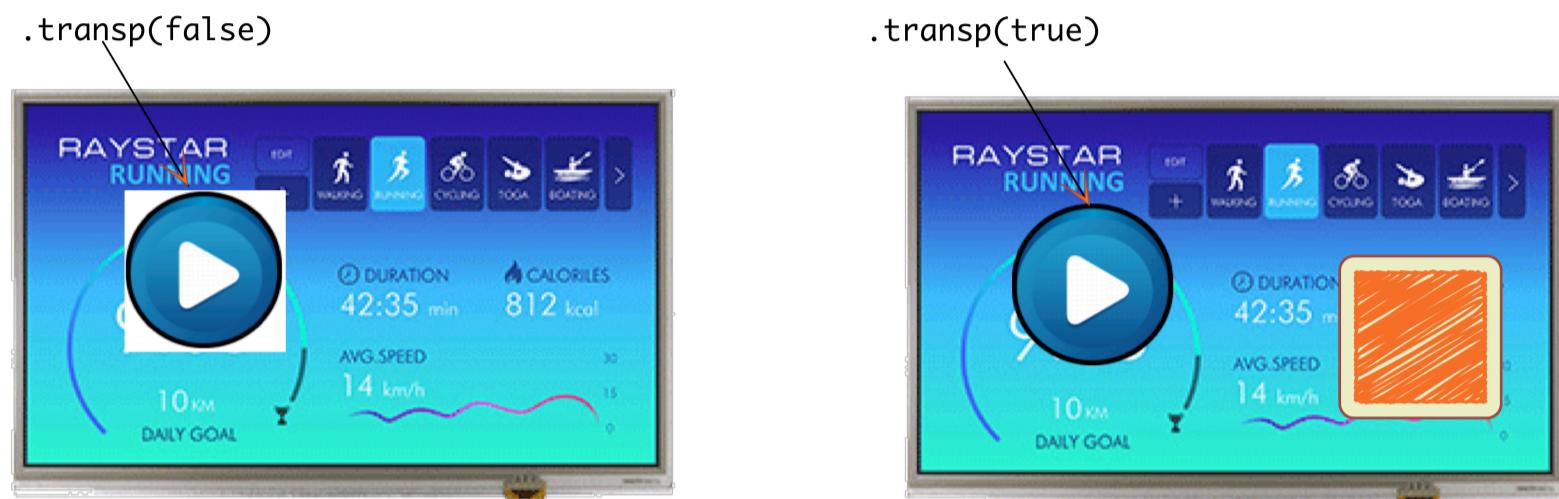
Questo metodo controlla la posizione dell'oggetto all'interno dell'area video.



TRANSP

Questo metodo controlla la trasparenza di un oggetto, quando questo non è trasparente, durante il redraw non ricarica lo sfondo che lo ospita.

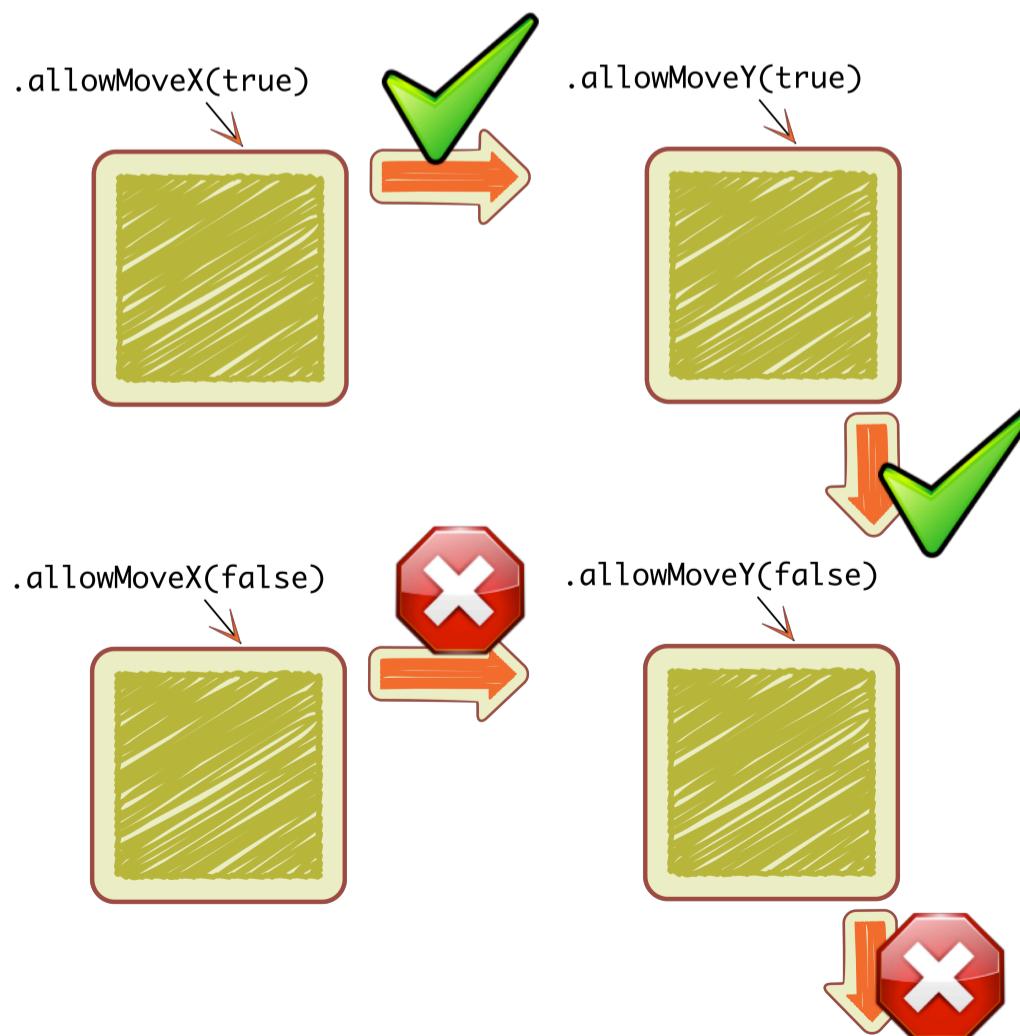
Un oggetto trasparente, durante il refresh obbliga il proprio sfondo al ridisegno prima di eseguire il proprio refresh.



ALLOWMOVEX/Y

Questo metodo permette o nega lo spostamento negli assi X e Y effettuati direttamente da Skinner.

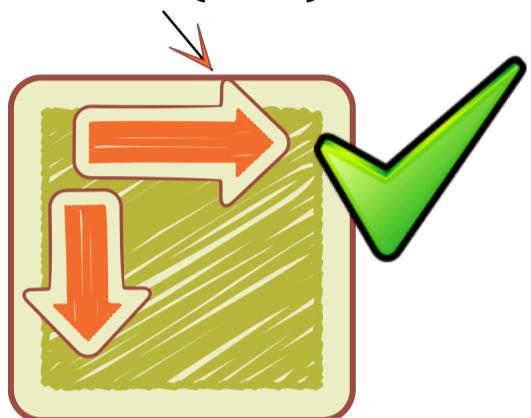
Per default lo spostamento è abilitato.



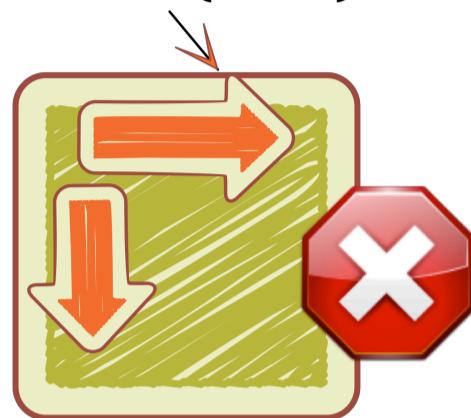
ALLOWRESIZE

Questo metodo permette o nega il ridimensionamento effettuato direttamente da Skinner.
Per default il resize è abilitato.

.allowResize(true)



.allowResize(false)



AUTOHIDE

Questo metodo abilita la funzionalità di Auto Hide, che mette in Hide automaticamente un oggetto dopo uno specifico tempo di inattività dell'utente (nessuna evento sul Touch screen o simili).

Questa modalità consente di nascondere automaticamente degli oggetti durante l'inattività dell'utente, poi ripresentandoli appena intercettata una nuova attività.

E' possibile configurare per ogni singolo oggetto anche la modalità di Auto Hide secondo il seguente schema:

Modo	Descrizione
left	Scompare verso sinistra
right	Scompare verso destra
up	Scompare verso l'alto
down	Scompare verso il basso
hide	Scompare senza alcun movimento

BLINK

Questo metodo abilita la funzionalità di Blink, creando un effetto Blink tramite l'Hide e lo Show dell'oggetto stesso.

La configurazione è normalmente globale, ma se necessario è possibile crearne una custom per lo specifico oggetto.

ANI

Questo metodo abilita l'animazione dell'oggetto. Ogni oggetto può essere configurato con una propria modalità di animazione.

Modo	Descrizione
shift	spostamento laterale
squeeze	collassata
squeezecentral	collassata centrale
global	Riflette la configurazione Globale
none	nessuna animazione

LE CLASSI

In questo capitolo sono raggruppate tutte le classi.

Classe	Tipologia	Descrizione
sknSys	Sistema	Gestione servizi essenziali e componenti comuni
sknDesk	Screen	Gestione dello schermo
sknColl	Collection	Gestione delle liste di immagini e dizionari
sknMmedia	MMedia	Gestione del Multi Media
sknWin	Window	Gestione Window
sknLbl	Widget	Gestione Label
sknBtn	Widget	Gestione Button
sknIbtn	Wdget	Gestione Image Button
sknITbtn	Wdget	Gestione Text and Image Button
sknImg	Widget	Gestione della presentazione immagini
sknAnimate	Widget	Gestione delle immagini animate
sknChkbox	Widget	Gestione delle CheckBox
sknProgbar	Widget	Gestione delle Progress Bar
sknMimage	Collection	Gestione delle Multi Image
sknKbrd	Object	Gestione della Keyboard
sknGraph	Object	Gestione Grafici cartesiani
sknGauge	Object	Gestione del Gauge
sknPie	Object	Gestione Grafici a Torta
sknDraw	Utility	Elementari utilità di disegno e Memory Devices
sknDbg	Debug	Analisi dei componenti attivi e semplice debug

SYSTEM

La Classe **sknSys** consente di configurare i servizi essenziali di Skinner e di servire le funzionalità comuni.

Membro	Descrizione
Avvio-Arresto	
sknSys.start()	Inizializza Skinner ed avvia i servizi di base
skn.stop()	Ferma i servizi di Skinner
skn.restart()	Riavvia i servizi di Skinner
skn.exit()	Termina Skinner
Configurazione on fly	
skn.userTimeout()	Configura il valore di user timeout
Esecuzione Asincrono	
skn.asyncFunct()	Richiede l'esecuzione di una funzione in modo asincrono
Gruppi	
sknSys.group_new()	Crea un gruppo
sknSys.group_delete()	Elimina un gruppo
sknSys.group_hide()	Nascondi un gruppo
sknSys.group_show()	Rendi visibile un gruppo
sknSys.group_list()	Visualizza la lista dei gruppi
Special modes	
sknSys.gesture()	Abilita le gesture
sknSys.ecomode()	Abilita la modalità Eco
sknSys.ecoSettings()	Configura la modalità eco mode
Info	
sknSys.version()	Riporta la versione di Skinner
Screen	
sknSys.screenXsize()	Ritorna la dimensione orizzontale del display in pixels
sknSys.screenYsize()	Ritorna la dimensione verticale del display in pixels
Touchpanel	
sknSys.tpMove()	Registra gli eventi di moving per il Touchpanel nelle quattro principali direzioni
sknSys.tpFinemove()	Registra gli eventi di fine moving per il Touchpanel nelle quattro principali direzioni
sknSys.tpSens()	Regola la sensibilità della gesture detection
sknSys.tplongTouch()	Configura il Long Touch
sknSys.tplongTouchEv()	Configura il Long Touch event

sknSys.tpLongTshow()	Configura il Long Touch auto show
sknSys.tpLivemove()	Configura l'evento relativo al Live move
Get special values	
sknSys.getColor ()	Converte un colore numerico o Ascii in un valore di colore
Management	
sknSys.getMaxObj()	Riporta il numero massimo di oggetti utilizzabili contemporaneamente
sknSys.getWidgetType()	Ritorna il tipo di widget relativo all'oggetto selezionato
sknSys.getParent()	Ritorna l'oggetto Padre
sknSys.getParentIndex()	Ritorna l'oggetto Padre relativo all'indice

Classe [sknSys]

sknSys.start()

Avvia Skinner, attiva i servizi di Background e i servizi base che rendono operativo Skinner.

sknSys.start (MemDeviceOn)	
Avvia il supporto skinner con o senza memory devices. Questo comando deve essere utilizzato prima di qualsiasi operazione su Skinner.	
param	memDeviceOn true=Abilita I Memory Devices
return	1=successo, 0=errore
call	sknSys.start (true)

Code example:

```
sknSys.start(false)  
sknColl.addDict ("GEN", 400, 300000)
```

Note:

La gestione dei Memory Devices gestita globalmente non è ancora completamente funzionante.

sknSys.stop()

Arresta i servizi di Skinner.

sknSys.stop()	
	Sospende il supporto Skinner. Questo comando viene utilizzato per sospendere tutte le attività di Skinner.
param	
return	1=successo, 0=errore
call	sknSys.stop()

sknSys.restart()

Riavvia Skinner.

sknSys.	
	Riavvia il supporto Skinner. Questo comando forza un riavvio completo di Skinner
param	
return	1=successo, 0=errore
call	sknSys.restart()

sknSys.exit()

Termina Skinner.

sknSys.	
	Arresta il supporto Skinner. Dopo questo comando non è più possibile invocare comandi Skinner.
param	
return	1=successo, 0=errore
call	sknSys.exit()

sknSys.userTimeout()

Configure lo User Timeout, tempo dopo il quale viene chiamata la funzione **usrActivity()**.

sknSys.userTimeout(TOsec)	
Configura il valore di user timeout. L'user timeout è il tempo di inattività verso la user interface, quindi quando l'utente non opera nessun tocco sul display, dopo il tempo configurato viene generato un evento di Timeout "usrTimeout()". Ad ogni evento dell'utente è possibile generare un evento "usrActivity()". E' sufficiente esporre le relative funzioni come pubbliche, queste verranno automaticamente chiamate in occasione dell'evento.	
param	TOsec
	Il valore può assumere valori da 1 a 32000, il valore 0 indica che non verranno generati eventi di timeout.
return	1=successo, 0=errore
call	sknSys.userTimeout(300)

sknSys.asyncFunct()

Richiede l'esecuzione di una funzione asincrona.

sknSys.asyncFunct("functName", "functStrParam")		
Richiede l'esecuzione di una funzione asincrona. La funzione verrà eseguita non appena la coda delle funzioni asincrone.		
param	functName	Nome della funzione da eseguire
	functStrParam	Parametro della funzione o ""
return	1=successo, 0=errore	
call	sknSys.asyncFunct("myFunct", "varSpecific")	

sknSys.group_new()

Aggiunge un gruppo.

I gruppi consentono di eseguire operazioni massive riferendosi al nome del gruppo (ed eventualmente sotto gruppo). Viene utilizzato normalmente per nascondere o visualizzare un gruppo di oggetti all'interno di una Form, un insieme di finestre, etc.

sknSys.group_new("groupName")

	Aggiunge un gruppo a cui associare degli oggetti	
param	groupName	deve essere un nome valido non già utilizzato nei gruppi.
return	1=successo, 0=errore	
call	sknSys.group_new("groupName")	

sknSys.group_delete()

Elimina un gruppo.

sknSys.group_delete("groupName")

	Elimina un gruppo a cui associare degli oggetti	
param	groupName	Nome di un gruppo esistente
return	1=successo, 0=errore	
call	sknSys.group_delete("groupName")	

sknSys.group_hide()

Nasconde un gruppo.

sknSys.group_hide("groupName", "subGroupName")

	Nasconde un gruppo o un sotto gruppo associato ad uno specifico gruppo	
param	groupName	Nome di un gruppo esistente
	subGroupName	Nome di un sotto gruppo o "" per qualsiasi
return	1=successo, 0=errore	
call	sknSys.group_hide("grpInfo", "subgSecure")	

sknSys.group_show()

Rende visibile un gruppo.

sknSys.group_show("groupName", "subGroupName")		
	Rende visibile un gruppo o un sotto gruppo associato ad uno specifico gruppo	
param	groupName	Nome di un gruppo esistente
	subGroupName	Nome di un sotto gruppo o "" per qualsiasi
return	1=successo, 0=errore	
call	sknSys.group_show("grplInfo", "subgSecure")	

sknSys.group_list()

Rende visibile un gruppo.

sknSys.group_list()	
	Rende visibile un gruppo o un sotto gruppo associato ad uno specifico gruppo
param	
return	Stampa la lista dei gruppi seguita da 1=successo, 0=errore
call	sknSys.group_list()

Esempio di Output del comando da console:

```
> sknSys.group_list()
Group list (active 45 of 50)

groupA
groupB
groupD
groupSub1
groupSub2
1
>
```

sknSys.gesture()

Abilita le gesture

sknSys.gesture(gestureOn)

	Abilita le gesture
param	gestureOn
	true=Gesture abilitate
return	1=successo, 0=errore
call	sknSys.gesture(true)

sknSys.version()

Riporta la versione di Skinner.

Il valore riportato in decimale, se convertito in esadecimale riporta la seguente notazione:

0x00VVRRBB (VV=versione, RR=release, BB=beta)

sknSys.version()

	Riporta la versione di Skinner
param	
return	Valore numerico (decimale) della versione codificato in esadecimale come segue: 0x00VVRRBB (VV=versione, RR=release, BB=beta)
call	print(sknSys.version())

sknSys.ecoSettings()

Configura la modalità eco mode.

sknSys.ecoSettings(winBGcolor, labelBGcolor, buttonBGcolor, fineMove)		
Configura la modalità eco mode. E' possibile configurare l'aspetto della modalità eco nei colori degli artefatti disegnati in modalità eco.		
param	winBGcolor	Colore di sfondo delle Finestre
	labelBGcolor	Colore di sfondo delle Label (numerico o stringa per colore predefinito)
	buttonBGcolor	Colore di sfondo dei Buttons
	fineMove	true=eco mode durante il fine move, false=eco mode disabilitato durante il fine move
return	1=successo, 0=errore	
call	sknSys.ecoSettings("black", sknSys.getColor("white"), "yellow", true)	

sknSys.screenXsize()

Ritorna la dimensione orizzontale del display in pixels.

sknSys.screenXsize()	
	Ritorna la dimensione orizzontale del display in pixels.
param	
return	Numero di pixels dell'asse X
call	Print (sknSys.screenXsize())

sknSys.screenYsize()

Ritorna la dimensione verticale del display in pixels.

sknSys.screenYsize()	
	Ritorna la dimensione verticale del display in pixels.
param	
return	Numero di pixels dell'asse Y
call	Print (sknSys.screenYsize())

sknSys.tpMove()

Registra gli eventi di moving per il Touchpanel nelle quattro principali direzioni.

sknSys.tpMove("upEvent", "dwnEvent", "leftEvent", "rightEvent")		
	Registra gli eventi di moving per il Touchpanel nelle quattro principali direzioni. Scorrendo nell'intera area del display (Sx, Dx, Up, Down) possono essere generati eventi relativi al movimento.	
param	upEvent	Nome della funzione da chiamare nello scorrimento verso l'alto (" per nessuno)
	dwnEvent	Nome della funzione da chiamare nello scorrimento verso il basso (" per nessuno)
	leftEvent	Nome della funzione da chiamare nello scorrimento verso sinistra (" per nessuno)
	rightEvent	Nome della funzione da chiamare nello scorrimento verso destra (" per nessuno)
return	1=successo, 0=errore	
call	sknSys.tpMove("sxmove", "dxmove", "", "")	

sknSys.tpFinemove()

Registra gli eventi di fine moving per il Touchpanel nelle quattro principali direzioni.

sknSys.tpFinemove("upEvent", "dwnEvent", "leftEvent", "srightEvent")		
		Registra gli eventi di fine moving per il Touchpanel nelle quattro principali direzioni. Scorrendo nell'intera area del display (Sx, Dx, Up, Down) possono essere generati eventi relativi al movimento.
param	upEvent	Nome della funzione da chiamare nello scorrimento verso l'alto("") per nessuno)
	dwnEvent	Nome della funzione da chiamare nello scorrimento verso il basso ("") per nessuno)
	leftEvent	Nome della funzione da chiamare nello scorrimento verso sinistra("") per nessuno)
	srightEvent	Nome della funzione da chiamare nello scorrimento verso destra("") per nessuno)
return	1=successo, 0=errore	
call	sknSys.tpFinemove("sxmove", "dxmove", "", "")	

sknSys.tpSens()

Regola la sensibilità della gesture detection.

sknSys.tpSens(Xsens, YDiagSens)		
		Regola la sensibilità della gesture detection. Permette di rendere più o meno sensibile la gesture detection, di fatto è possibile decidere per quanti punti si deve effettuare la gesture prima che questa venga rilevata.
param	Xsens	Sensibilità dell'asse X (1 – 100) :default = 30
	Ysens	Sensibilità dell'asse Y (1- 100): default = 15
return	1=successo, 0=errore	
call	sknSys.tpSens(30, 15)	

sknSys.tplongTouch()

Configura il Long Touch.

sknSys.tplongTouch(enable, touchTime_ms, longTouchTimeout_ms)		
	Configura il Long Touch. E' possibile configurare il Long touch, non solo nella abilitazione ma anche nella programmazione del tempo di long touch e del tempo di long touch timeout.	
param	Enable	true=Long touch abilitato, false=Long touch disabilitato
	touchTime_ms	Tempo di touch per essere definito un Long Touch (100 – 32000 millisecondi)
	longTouchTimeout_ms	Tempo di touch per essere definito un Long Touch Timeout (100 – 32000 millisecondi)
return	1=successo, 0=errore	
call	sknSys.tplongTouch(true, 3000, 10000)	

sknSys.tplongTouchEv()

Configura il Long Touch event.

sknSys.tplongTouchEv("eventName")		
	Configura il Long Touch event. Registra il nome della funzione da chiamare al rilevamento di un Long Touch.	
param	eventName	nome di una funzione pubblica
return	1=successo, 0=errore	
call	sknSys.tplongTouchEv("cbLongTouch")	

sknSys.tpLongTshow()

Configura il Long Touch auto show.

sknSys.tpLongTshow(longTouchAutoShow)		
	Configura il Long Touch auto show. E' possibile configurare l'auto show dopo un long touch o dopo qualsiasi azione sull'interfaccia dell'utente.	
param	longTouchAutoShow	true=Auto show dopo un long touch, false=Auto show dopo una azione utente
return	1=successo, 0=errore	
call	sknSys.tpLongTouchTshow(true)	

sknSys.tpLivemove()

Configura l'evento relativo al Live move.

Trascinamento del puntatore nello schermo.

sknSys.tpLivemove(eventLive)		
	durante il move viene chiamata la relativa funzione con un parametro string che contiene X,Y (Es: "123,23") Per determinare la posizione corrente del Topuch è possibile anche utilizzare all'interno dell'evento il metodo "sknDesk.currCoords()"	
param	eventLive	nome di una funzione pubblica
return	1=successo, 0=errore	
call	sknSys.tpLivemove("moveObject")	

sknSys.getColor()

Converte un colore numerico o Ascii in un valore di colore.

sknSys.getColor (color)

Converte un colore numerico o Ascii in un valore di colore.
Al fine di facilitare l'impiego di colori predefiniti (tipo: "white", "black", ...) questa funzione è in grado di ritornare il colore numerico di un qualsiasi colore predefinito o eventualmente un codice Hex del colore stesso (es: 0x00FF00FF)
I colori predefiniti sono:
"blue", "green", "red", "cyan", "magenta", "yellow", "lightblue", "lightgreen", "lightred"
"lightcyan", "lightmagenta", "lightyellow", "lightgray", "darkblue", "darkgreen", "darkred"
"darkcyan", "darkmagenta", "darkyellow", "darkgray", "white", "gray", "black"
"brown", "transparent"

param	color	un qualsiasi colore predefinito tipo "white" o un codice numerico di colore espresso in decimale o esadecimale (0x)
-------	-------	---

return Valore numerico del colore corrispondente (RGB)

call sknSys.getColor("white")

...

sknSys.getColor(43000)

sknSys.getColor(0x00F0F0F0)

sknSys.getMaxObj()

Riporta il numero massimo di oggetti utilizzabili contemporaneamente.

sknSys.getMaxObj()	
	Riporta il numero massimo di oggetti utilizzabili contemporaneamente
param	
return	Numero di oggetti supportati
call	sknSys.getMaxObj()

sknSys.getWidgetType()

Ritorna il tipo di widget relativo all'oggetto selezionato.

sknSys.getWidgetType("widgetName")	
	Ritorna il tipo di widget relativo all'oggetto selezionato
param	widgetName
	Nome dell'oggetto da analizzare
return	Tipo di oggetto o "" se non trovato
call	Print (sknSys.getWidgetType("myWidget"))

sknSys.getParent()

Ritorna l'oggetto Madre.

sknSys.getParent("widgetName")	
	Ritorna l'oggetto Padre
param	widgetName
	Nome dell'oggetto figlio
return	Nome dell'oggetto Madre o "" se non trovato
call	Print (getParent("myWidget"))

sknSys.getParentIndex()

Ritorna l'oggetto Padre relativo all'indice.

sknSys.getParentIndex(objectIndex)	
	Ritorna l'oggetto Padre relativo all'indice
param	objectIndex Indice dell'oggetto (1-N)
return	Nome dell'oggetto Padre o "" se non trovato
call	Print (sknSys.getParentIndex(12))

DESKTOP

La classe “sknDesk” controlla le funzionalità generiche dello screen.

Membro	Descrizione
Configurazione	
sknDesk.calibrate()	Forza la calibrazione del touchpanel
sknDesk.activate()	Attiva la gestione dei servizi relativi al Desktop
sknDesk.colors()	Configura i colori del desktop
sknDesk.cursoron()	Configura la presenza del cursore
sknDesk.memdevon()	Abilita l'uso dei memory devices
sknDesk.touchon()	Abilita l'uso del touchpanel
sknDesk.currCoords()	Riporta le coordinate correnti
Anti Aliasing	
sknDesk.AAenable()	Abilita Anti Aliasing
sknDesk.AAhires()	Abilita Anti Aliasing ad alta risoluzione
sknDesk.AAfactor()	Configura il fattore di Anti Aliasing
Core	
sknDesk.guiexecon()	Abilita il refresh del display
Animazione	
sknDesk.autohidecfg()	Autohide configuration (Set/Get)
sknDesk.anicfg()	Amination configuration (Set/Get)
sknDesk.blinkcfg()	Configurazione della funzionalità di Blink (Set/Get)
Markers	
sknDesk.arrow()	Configurazione del puntatore
sknDesk.posMarkColors()	Configurazione dei colori markers (Set/Get)
sknDesk.posMarkDims()	Configurazione della dimensione markers (Set/Get)

Classe [sknDesk]

sknDesk.calibrate()

Forza la calibrazione del Touch Panel.

Comando non disponibile nelle macchine Host.

sknDesk.calibrate()	
	Forza la calibrazione del touchpanel. Una volta eseguito questo comando eseguire il reboot del sistema per entrare in modalità calibrazione.
param	
return	1=successo, 0=errore
call	sknDesk.calibrate()

sknDesk.activate()

Attiva la gestione dei servizi relativi al Desktop.

sknDesk.activate()	
	Attiva la gestione dei servizi relativi al Desktop. E' necessario attivare il Desktop per ottenere i servizi di refresh del display e la lettura del Touchpanel.
param	
return	1=successo, 0=errore
call	sknDesk.activate()

sknDesk.colors()

Configura i colori del desktop.

sknDesk.colors(colorFG, colorBG)		
Configura i colori del desktop. Esegue la configurazione del colore di sfondo e primo piano del Desktop.		
param	colorFG	colore di primo piano (numerico o stringa per colore predefinito)
	colorBG	colore di sfondo (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknDesk.colors("black","transparent")	

sknDesk.cursoron ()

Configura la presenza del cursore.

sknDesk.cursoron (cursorOn)		
	Configura la presenza del cursore. Permette la visualizzazione opzionale del cursore.	
param	cursorOn	true=cursore visibile, false=cursore non attivo
return	1=successo, 0=errore	
call	sknDesk.cursoron (true)	

sknDesk.memdevon()

Abilita l'uso dei Memory Devices.

sknDesk.memdevon (memdevOn)		
	Abilita l'uso dei Memory Devices	
param		
return	1=successo, 0=errore	
call	sknDesk.memdevon(true)	

sknDesk.touchon()

Abilita l'uso del touchpanel.

sknDesk.touchon (touchOn)		
	Abilita l'uso del touchpanel.	
param	touchOn	true=touchpanel abilitato, false=touchpanel non abilitato
return	1=successo, 0=errore	
call	sknDesk.touchon(true)	

sknDesk.currCoords()

Riporta le coordinate correnti.

sknDesk.currCoords()	
	Riporta le coordinate correnti
param	
return	X, Y correnti
call	x,y = sknDesk.currCoords()

sknDesk.AAenable()

Abilita Anti Aliasing.

sknDesk.AAenable(enable)	
	Abilita Anti Aliasing
param	enable
	true=Anti Aliasing abilitato
return	1=successo, 0=errore
call	sknDesk.AAenable(false)

sknDesk.AAhires()

Abilita Anti Aliasing ad alta risoluzione.

sknDesk.AAhires(enable)	
	Abilita Anti Aliasing ad alta risoluzione.
param	enable
	true=abilitato l'Anti Aliasing ad alta risoluzione
return	1=successo, 0=errore
call	sknDesk.AAhires(true)

sknDesk.AAfactor()

Configura o legge il fattore di Anti Aliasing.

sknDesk.AAfactor(AAfactor)		
	Configura o legge il fattore di Anti Aliasing.	
param	AAfactor	Fattore Anti Aliasing (1-6)
return	1=successo, 0=errore, fattore corrente di Anti Aliasing	
call	sknDesk.AAfactor(3) ... print (sknDesk.AAfactor())	

sknDesk.guiexecOn()

Abilita il refresh del display.

sknDesk.guiexecOn(guiExecOn)		
	Abilita il refresh del display. Questa funzionalità si occupa dell'intero render del display e di alcune funzionalità Core della Graphics Interface.	
param	guiExecOn	true=GUI refresh abilitata, false=Disabilitato il refresh della GUI
return	1=successo, 0=errore	
call	sknDesk.guiexecOn(true)	

sknDesk.autohidcfg ()

Autohide configuration (Set/Get).

Funzionalità condizionata dalla profilazione **SKN_HIDDEN_OBJ_EN == 1**.

sknDesk.autohidcfg (numStep, stepDelay)		
Autohide configuration (Set/Get). Imposta o ritorna la configurazione dell'auto hide (numero di step e delay di ogni step). Chiamando la funzione senza parametri, viene ritornata la configurazione corrente.		
param	numStep	numero di step per Show / Hide
	stepDelay	delay per ogni step (millisecondi)
return	1=successo, 0=errore o numStep, stepDelay	
call	sknDesk.autohidcfg (3, 200) ... print (sknDesk.autohidcfg ())	

sknDesk.anicfg()

Amination configuration (Set/Get).

Funzionalità condizionata dalla profilazione **SKN_ANIMATION_ACTIVE == 1**.

sknDesk.anicfg(numStep, stepDelay, "aniMode")		
Amination configuration (Set/Get). Imposta o ritorna la configurazione dell'auto hide (numero di step e delay di ogni step). Chiamando la funzione senza parametri, viene ritornata la configurazione corrente.		
Elenco delle modalità: "shift" = spostamento laterale "squeeze" = collassata "squeezcentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione		
param	numStep	numero di step per Show / Hide
	stepDelay	delay per ogni step (millisecondi)
	aniMode	animation mode {"shift", "squeeze", "squeezcentral", "global", "none"}
return	1=successo, 0=errore o numStep, stepDelay	
call	sknDesk.anicfg(4, 230, "shift") ... print (sknDesk.anicfg())	

sknDesk.blinkcfg()

Configurazione della funzionalità di Blink (Set/Get).

Funzionalità condizionata dalla profilazione **SKN_BLINK_OBJ_EN == 1**.

sknDesk.blinkcfg(blinkDelay)		
Configurazione della funzionalità di Blink (Set/Get). Imposta o ritorna la configurazione del blink (delay di ogni stato). Chiamando la funzione senza parametri, viene ritornata la configurazione corrente.		
param	blinkDelay	tempo di Show / Hide in millisecondi
return	1=successo, 0=errore o blinkDelay corrente	
call	sknDesk.blinkcfg(200) ... print (blinkcfg())	

sknDesk.arrow()

Configurazione del puntatore.

sknDesk.arrow("imgColl", "imgName")		
Configurazione del puntatore. Imposta il tipo di puntatore / cursore.		
param	imgColl	nome della collezione di immagini
	imgName	nome dell'immagine
return	1=successo, 0=errore	
call	bsknDesk.arrow("curs", "arrowBig")	

sknDesk.markColors()

Configurazione dei colori markers (Set/Get).

sknDesk.markColors(barColor, markColor)		
Configurazione dei colori markers (Set/Get). Imposta o ritorna i colori dei markers (colore della barra e colore del marker). Chiamando la funzione senza parametri, viene ritornata la configurazione corrente.		
param	barColor	colore della barra (numerico o stringa per colore predefinito)
	markColor	colore del marker (numerico o stringa per colore predefinito)
return	1=successo, 0=errore o barColor, markColor	
call	barColor, markColor = sknDesk.markColors() ... sknDesk.markColors("white", 0x00FF)	

sknDesk.markDims()

Configurazione della dimensione markers (Set/Get).

sknDesk.markDims(barWidth, markWidth)		
Configurazione della dimensione markers (Set/Get). Imposta o ritorna la dimensione dei markers (colore della barra e colore del marker). Chiamando la funzione senza parametri, viene ritornata la configurazione corrente.		
param	barWidth	dimensione della barra
	markWidth	dimensione del marker
return	1=successo, 0=errore o barWidth, markWidth	
call	barWidth, markWidth = sknDesk.markDims() ... sknDesk.markDims(20, 40)	

COLLECTION

La Classe **sknColl** controlla liste e collezioni tipo Image list e Dictionary.

Membro	Descrizione
Collezione immagini	
sknColl.addImgColl()	Aggiunge una immagine presente su file in una collezione.
sknColl.imgDestroy ()	Rimuove una collezione di immagini.
sknColl.imgAdd ()	Aggiunge una immagine presente su file in una collezione.
sknColl.imgCollList()	Stampa la lista delle collezioni.
sknColl.imgInfos()	Stampa su terminale le informazioni di una collezione.
sknColl.imgSize()	Get image size
sknColl.maxImages()	Numero massimo di immagini per la collezione
sknColl.imagename	Image name da indice di una collezione
Dizionari	
sknColl.addDict()	Aggiungi un dizionario
sknColl.dictLoad()	Carica un dizionario
sknColl.dictGetPhrase()	Recupera una frase dal dizionario
Ram fonts	
sknColl.font_mount()	Carica un Ram font da file.
sknColl.fontUnmount()	Elimina tutti i Ram font caricati.
sknColl.font_infos()	Stampa su terminale le informazioni sui font caricati.
sknColl.font_exists()	Verifica la presenza di un font.
Sound	
sknColl.soundNew()	Aggiungi un nuovo suono.
sknColl.soundAddSample()	Aggiungi una nota ad un suono.
sknColl.soundPlay()	Esegui il Play di un suono.
sknColl.soundDelete()	Elimina un suono.
sknColl.soundList()	Stampa su terminale l'elenco dei suoni.
sknColl.soundInfo()	Stampa su terminale le informazioni relative ad uno specifico suono.

Classe [sknColl]

sknColl.addImgColl()

Aggiunge una nuova collezione di immagini.

Le immagini saranno salvate in modalità raw senza spazi, questo riduce sensibilmente lo spazio di memoria utilizzata, per contro è impossibile rimuovere una singola immagine dalla collezione.

sknColl.addImgColl("imgCollName", maxImages, bufferSize)

	Aggiunge una nuova collezione di immagini.	
param	Collname	nome della collezione di immagini
	maxImages	numero massimo di immagini supportate
	bufferSize	dimensione del buffer
return	1=successo, 0=errore	
call	sknColl.addImgColl("btn", 100, 200000)	

sknColl.maxImages()

Ritorna il numero di immagini di una collezione.

sknColl.maxImages("imgCollName")

	Ritorna il numero di immagini contenute in una specifica collezione di immagini.	
param	Collname	nome della collezione di immagini
return	numero di immagini, 0=errore	
call	maxImg = sknColl.maxImages("btn")	

sknColl.imageName()

Ritorna il nome dell'immagine relativa all'indice di una specifica collezione.

sknColl.imageName("imgCollName", index)

	Ritorna il nome dell'immagine relativa all'indice di una specifica collezione di immagini.	
param	Collname	nome della collezione di immagini
	index	Indice (0-N)
return	nome immagine, ""=errore	
call	myImgName = sknColl.imageName("btn", 12)	

sknColl.imgDestroy()

Rimuove una collezione di immagini.

sknColl.imgDestroy("collectName")		
	Rimuove una collezione di immagini.	
param	collectName	nome della collezione di immagini
return	1=successo, 0=errore	
call	sknColl.imgDestroy("btn")	

sknColl.imgAdd()

Aggiunge una immagine presente su file in una collezione.

sknColl.imgAdd("collectName", "imageFileName", "imagename", fastBmp, transparent)		
	Aggiunge una immagine presente su file in una collezione.	
param	collectName	nome della collezione di immagini
	imageFileName	image file name (percorso completo)
	Imagename	nome da assegnare all'immagine
	fastBmp	true=immagine fast, false=normal imageArrayName
	Transparent	true=immagine trasparente, false=immagine non trasparente
return	1=successo, 0=errore	
call	sknColl.imgAdd("btn", "mmc:\\images\\buttons\\home.bmp", "home", false, false)	

sknColl.imgCollList()

Stampa la lista delle collezioni.

sknColl.imgCollList()	
	Stampa la lista delle collezioni.
param	
return	1=successo, 0=errore
call	sknColl.imgCollList()

Esempio di Output del comando da console:

```
>sknColl.imgCollList()

Image collection list (active 6 of 10)

btn
btnP
btnD
icon
BG
IMG
1
>
```

sknColl.imgInfos()

Stampa su terminale le informazioni di una collezione.

sknColl.imgInfos("collectName", deepInfos)		
	Elenca le informazioni relative alla collezione selezionata.	
param	collectName	nome della collezione di immagini
	deepInfos	true=Visualizza dettagli, false=non visualizzare i dettagli
return	1=successo, 0=errore	
call	sknColl.imgInfos("btn", true)	

Esempio di Output del comando da console:

```
> sknColl.imgInfos("btn", true)

Collection index : 0 allowed 100 images
                  storage = 2000000 byte,   free size = 1636717
byte, free image entries = 83

+ image name : "chart"
  X size : 48
  Y size : 48
  details : TRANSP
  raw size : 9223

+ image name : "buttonRect"
  X size : 213
  Y size : 55
  details : TRANSP
  raw size : 46867

+ image name : "hum"
  X size : 48
  Y size : 48
  details : TRANSP
  raw size : 9223

+ image name : "temp"
  X size : 48
  Y size : 48
  details : TRANSP
  raw size : 9223

+ image name : "airq"
  X size : 48
  Y size : 40
  details : TRANSP
  raw size : 7687
```

sknColl.imgSize()

Get image size.

sknColl.imgSize("collectName", "imagename")		
	Get image size	
param	collectName	Nome della collezione
	Imagename	Nome dell'immagine
return	X, Y image size or 0=errore	
call	x,y = sknColl.imgSize("btn", "homeBtn")	

sknColl.addDict()

Aggiungi un dizionario.

E' necessario creare una serie di dizionari per poterli utilizzare nelle varie lingue supportate. In Skinner è consigliato utilizzare i messaggi sotto forma di ID (il cardinale del dizionario) che viene risolto da una apposita funzione e ritorna la stringa coretta secondo la lingua corrente.

Di norma i dizionari sono divisi per categoria, ad esempio:

Sistema, calendario, utente... etc

sknColl.addDict("dictName", maxStrings, bufferSize)		
	Aggiungi un dizionario	
param	dictName	Nome del dizionario
	maxStrings	Numero massimo di stringhe supportate
	bufferSize	Dimensione del dizionario in byte
return	1=successo, 0=errore	
call	sknColl.addDict("calendar", 200, 8192)	

sknColl.dictLoad()

Carica un dizionario.

Il caricamento del dizionario permette di selezionare la lingua corrente.

La stringa che determina la lingua è riportata all'interno del dizionario stesso.

sknColl.dictLoad("dictionaryName", "language", "fileDictionary")

	Carica un dizionario con uno specifico file di messaggi. Carica da file l'elenco dei messaggi secondo la lingua selezionata. Per modificare la lingua è sufficiente eseguire nuovamente questa funzione con una differente lingua.	
param	dixtionaryName	Nome del dizionario
	language	identificativo della lingua (deve coincidere con una di quelle presenti sul file)
	fileDictionary	Nome del file contenente il dizionario (percorso completo)
return	1=successo, 0=errore	
call	sknColl.dictLoad ("GEN", thema.language, thema.getDir() .. "\\\\"DGen.txt")	

```
//Multi language file for generic usage

#language ITA
//Date
0, "Gennaio"
1, "Febbraio"
2, "Marzo"
3, "Aprile"
4, "Maggio"
5, "Giugno"
6, "Luglio"
7, "Agosto"
8, "Settembre"
9, "Ottobre"
10, "Novembre"
11, "Dicembre"

12, "Domenica"
13, "Lunedì"
14, "Martedì"
15, "Mercoledì"
16, "Giovedì"
17, "Venerdì"
18, "Sabato"

#language EN
//Date
0, "January"
1, "February"
2, "March"
3, "April"
4, "May"
5, "June"
6, "Jult"
7, "August"
8, "September"
```

Esempio di dizionario:

sknColl.dictGetPhrase()

Recupera una frase dal dizionario

sknColl.dictGetPhrase("dictionaryName", phraseID)		
	Ritorna il messaggio presente su un dizionario tramite phraseID secondo la lingua selezionata. Prima di utilizzare questo metodo, è necessario aver caricato il corrispondente dizionario tramite: //dictLoad ("dictionaryName", "language", "fileDictionary")	
param	DictionaryName	Nome del dizionario
	phraseID	Indice del messaggio da riportare (0-N)
return	Messaggio relativo all'ID selezionato o "" se non trovata	
call	print(sknColl.dictGetPhrase("MYdict", 54))	

sknColl.font_mount()

Carica un Ram font da file.

Funzionalità condizionata dalla profilazione FONT_RAM_BASED == 1

sknColl.font_mount("RAMfont_file_name", "fontName")		
	Carica un Ram font da file.	
param	RAMfont_file_name	nome del file (percorso completo)
	fontName	nome da assegnare al font caricato
return	1=successo, 0=errore	
call	sknColl.ramFont_mount("mmc:\\fnt\\arial14.tx", "Arial14")	

sknColl.font_unmount()

Elimina tutti i Ram font caricati.

Funzionalità condizionata dalla profilazione FONT_RAM_BASED == 1

sknColl.font_unmount()		
	Elimina tutti i Ram font caricati.	
param		
return	1=successo, 0=errore	
call	sknColl.font_unmount()	

sknColl.font_infos()

Stampa su terminale le informazioni sui font caricati.

Funzionalità condizionata dalla profilazione FONT_RAM_BASED == 1

sknColl.font_infos(details)		
	Stampa su terminale le informazioni sui font caricati.	
param	Details	true=visualizza dettagli, false=non visualizzare i dettagli
return	1=successo, 0=errore	
call	sknColl.font_infos(true)	

Esempio di Output del comando da console:

```
> sknColl.font_infos(true)

Font list (installed 3 of 100)

F32B
F16B
F24B
1
>
```

sknColl.font_exists()

Verifica la presenza di un font.

Funzionalità condizionata dalla profilazione FONT_RAM_BASED == 1

sknColl.font_exists("fontName")		
	Verifica la presenza di un font	
param	fontName	Nome del font
return	true=font presente, false=non presente	
call	fontPresent = sknColl.font_exists("Arial14")	

sknColl.soundNew()

Aggiungi un nuovo suono.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknColl.soundNew("soundEffectName")		
Aggiungi un nuovo suono. Questa operazione censisce un nuovo suono, le varie componenti verranno aggiunte in seguito tramite soundAddSample().		
param	soundEffectName	un nome non utilizzato nei suoni
return	1=successo, 0=errore	
call	sknColl.soundNew("pilong")	

sknColl.soundAddsample()

Aggiungi una nota ad un suono.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknColl.soundAddsample("soundEffectName", "Note", octave, units)		
Aggiungi una nota ad un suono. Questa operazione aggiunge una nuova nota ad un suono, la nota viene aggiunta in coda a quelle esistenti.		
param	soundEffectName	un nome di un suono
	Note	una nota { "DO"=do, "DOD"=do diesis, "RE"=re, "RED",re diesis, "MI"=mi, "FA"=fa, "FAD" = fa diesis, "SOL"=sol, "SOLD"=sol diesis, "LA"=la, "LAD"=la diesis, "PAUSE"=pausa}
	Octave	Ottava
	Units	durata in unità
return	1=successo, 0=errore	
call	sknColl.soundAddSample("pilong", "DO", 2, 24)	

sknColl.soundPlay()

Esegui il Play di un suono.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknColl.soundPlay("soundEffectName")

	Esegui il Play di un suono.	
param	soundEffectName	un nome di un suono
return	1=successo, 0=errore	
call	sknColl.soundPlay("pilong")	

sknColl.soundDelete()

Elimina un suono.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknColl.soundDelete("soundEffectName")

	Elimina un suono.	
param	soundEffectName	un nome di un suono
return	1=successo, 0=errore	
call	sknColl.soundDelete("pilong")	

sknColl.soundList()

Stampa su terminale l'elenco dei suoni.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknColl.soundList()

	Stampa su terminale l'elenco dei suoni.	
param		
return	1=successo, 0=errore	
call	sknColl.soundList()	

sknColl.soundInfo()

Stampa su terminale le informazioni relative ad uno specifico suono.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknColl.soundInfo("soundEffectName", details)	
	Stampa su terminale le informazioni relative ad uno specifico suono.
param	soundEffectName un nome di un suono
	details true=stampa dettagli, false=non stampa i dettagli
return	1=successo, 0=errore
call	sknColl.soundInfo("pilong", true)

MULTIMEDIA

La Classe “sknMmedia” implementa il supporto per il Multimedia.

Membro	Descrizione
Multimedia	
sknMmedia.playAVIfile()	Riproduce il video di AVI.
sknMmedia.AVIpause()	Sospende la riproduzione del videoAVI.
sknMmedia.AVIplay()	Riprende una riproduzione del video AVI precedentemente messo in pausa.
sknMMediaAVIrewind()	Rewind del filmato AVI precedentemente messo in pausa o in riproduzione.
sknMMediaAVIstop()	Termina la riproduzione del filmato AVI precedentemente messo in pausa o in riproduzione.

Classe [sknMmedia]

sknMmedia.playAVIfile()

Riproduce il video di AVI.

sknMmedia.playAVIfile("filename", x, y, maxX, maxY, repeat)		
Riproduce solo la parte video di un AVI video con formato RGB 24 bit. Viene visualizzata una finestra con coordinate di partenza definite dove la dimensione della finestra è definita dalla dimensione nativa del filmato e dai limiti imposti.		
param	filename	AVI file name (percorso completo)
	x	corner X Up-Sx della finestra
	Y	corner Y Up-Sx della finestra
	maxX	max X size della finestra
	maxY	max Y size della finestra
	repeat	true=Ripetizione automatica del filmato, false=una sola riproduzione
return	1=successo, 0=errore	
call	sknMmedia.playAVIfile("mmc:\AVI\intro.avi", 10, 10, 400, 200)	

sknMmedia.AVIpause()

Sospende la riproduzione del video AVI.

sknMmedia.AVIpause()	
	Sospnde la riproduzione del video AVI.
param	
return	1=successo, 0=errore
call	sknMmedia.AVIpause()

sknMmedia.AVIplay()

Riprende una riproduzione del video AVI precedentemente messo in pausa.

sknMmedia.AVIplay()	
	Riprende una riproduzione del video AVI precedentemente messo in pausa.
param	
return	1=successo, 0=errore
call	sknMmedia.AVIplay()

sknMmedia.AVIrewind()

Rewind del filmato AVI precedentemente messo in pausa o in riproduzione.

sknMmedia.AVIrewind()	
	Rewind del filmato AVI precedentemente messo in pausa o in riproduzione.
param	
return	1=successo, 0=errore
call	sknMmedia.AVIrewind()

sknMmedia.AVIstop()

Termina la riproduzione del filmato AVI precedentemente messo in pausa o in riproduzione.

sknMmedia.AVIstop()	
	Termina la riproduzione del filmato AVI precedentemente messo in pausa o in riproduzione.
param	
return	1=successo, 0=errore
call	sknMmedia.AVIstop()

WINDOW

La Classe **sknWin** gestisce le Window.

Membro	Descrizione
Control	
sknWin.create()	Crea una Window
sknWin.delete()	Distrugge una Window
sknWin.enable()	Abilita la Window
sknWin.modal()	Rende modale la Window
sknWin.currwin()	Ritorna la window corrente
View	
sknWin.show()	Visualizza una Window
sknWin.showRestorable()	Visualizza una Window e permette il ritorno automatico
sknWin.showCaller()	Visualizza la Window che ha chiamato quella corrente
sknWin.showOnlyMe()	Visualizza una Window e nasconde tutte le altre
sknWin.hide()	Nascondi una Window
sknWin.goto()	Visualizza una Window portandola in primo piano con il focus
Proprietà	
sknWin.size()	Modifica le dimensioni di una Window
sknWin.color()	Modifica il colore dell'area di una Window
sknWin.bgcolor()	Modifica il colore di sfondo di una Window
sknWin.pos()	Modifica la posizione di una Window
sknWin.transp()	Controlla la trasparenza di una Window
Permessi	
sknWin.allowMoveX()	Controlla l'abilitazione al movimento su asse X di una Window
sknWin.allowMoveY()	Controlla l'abilitazione al movimento su asse Y di una Window
sknWin.allowResize()	Controlla l'abilitazione alla modifica della dimensione di una Window
Preview	
sknWin.miniature()	Configura l'immagine miniatura di una Window
sknWin.layout()	Configura l'immagine di layout di una Window
Visibilità	
sknWin.bringTop()	Porta la finestra in primo piano
sknWin.bringBot()	Porta la finestra in sfondo
sknWin.refresh()	Forza il ridisegno di una window
Aggregazione	

sknWin.auxData()	Controlla l'AuxData di una Window
sknWin.auxStr()	Controlla l'AuxString di una Window
sknWin.group()	Controlla l'appartenenza di un gruppo di una Window
sknWin.subGroup()	Controlla l'appartenenza di un sotto gruppo di una Window
Auto Hide	
sknWin.autoHide()	Controlla la funzionalità di auto hide di una Window
sknWin.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di una Window
sknWin.autoHideMode()	Controlla il tipo ti hide per la funzionalità di auto hide di una Window
Blink	
sknWin.blinkEn()	Controlla la funzionalità di blink di una Window
sknWin.blinkCustEn()	Controlla la funzionalità di blink custom o globale di una Window
Auto Hide Callback	
sknWin.autoHideCBshow()	Controlla la funzione di callback hide di una Window
sknWin.autoHideCBhide()	Controlla la funzione di callback show di una Window
Animazione	
sknWin.aniEn()	Controlla la funzionalità di animation di una Window
sknWin.aniMode()	Controlla la modalità di animation di una Window
Child	
sknWin.autoFocusChild()	Controlla la funzionalità di autofocus per le child window
sknWin.bringToTopChild()	Controlla la funzionalità di brig to top per le child window
Callbacks	
sknWin.longTouchCB()	Controlla il nome della funzione di callback per un long touch
sknWin.paintCB()	Controlla il nome della funzione di callback per il paint della window
sknWin.gestureCB()	Controlla il nome della funzione di callback per le gesture della window
sknWin.keyPressCB()	Controlla il nome della funzione di callback per il Key press della window
sknWin.keyReleaseCB()	Controlla il nome della funzione di callback per il Key release della window

Classe [sknWin]

sknWin.create()

Crea una Window.

sknWin.create ("winName", x, y, Xsize, Ysize)		
	Crea una Window.	
param	winName	Nome della window
	x	origine asse x
	Y	origine asse y
	Xsize	larghezza della window
	Ysize	altezza della window
return	1=successo, 0=errore	
call	sknWin.create("home", 0, 0, 480, 272)	

sknWin.show()

Visualizza una Window.

sknWin.show ("winName")		
	Visualizza una Window.	
param	winName	nome della window
return	1=successo, 0=errore	
call	sknWin.show("home")	

sknWin.showCaller()

Visualizza la Window precedentemente visualizzata con il metodo **sknWin.showRestorable()**.

sknWin.show()		
	Visualizza la Window precedentemente visualizzata con il metodo sknWin.showRestorable() .	
param	winName	nome della window
return	1=successo, 0=errore	
call	sknWin.show("home")	

sknWin.showRestorable ()

Visualizza una Window e ne permette il ritorno automatico tramite la funzione sknWin.showCaller.

sknWin.showRestorable ("winName")

Visualizza una Window e ne registra il nome in una lista di 30 elementi per poterla visualizzare dalla precedente finestra visualizzata con lo stesso metodo.
Utilizzare **sknWin.showCaller()** per tornare alla precedente window (visualizzata con questo metodo).

param winName nome della window

return 1=successo, 0=errore

call sknWin.showRestorable("home")

sknWin.showOnlyMe ()

Visualizza una Window e nasconde tutte le altre.

sknWin.showOnlyMe ("winName")

Visualizza una Window e nasconde tutte le altre.

param winName Nome della window

return 1=successo, 0=errore

call sknWin.showOnlyMe("home")

sknWin.hide ()

Nasconde una Window.

sknWin.hide ("winName")

Nasconde una Window.

param winName Nome della window

return 1=successo, 0=errore

call sknWin.hide("home")

sknWin.goto ()

Visualizza una Window portandola in primo piano con il focus nascondendone un'altra.

sknWin.goto ("winName", "gotoWinName")		
	Visualizza una Window portandola in primo piano con il focus.	
param	winName	Nome della window corrente (da nascondere)
	gotoWinName	Nome della window da mostrare
return	1=successo, 0=errore	
call	sknWin.goto("home", "swWindow")	

sknWin.size ()

Modifica o ritorna le dimensioni di una Window.

sknWin.size ("winName", x, y)		
	Modifica o ritorna le dimensioni di una Window. Se si omette x e y la funzione riporta quelli correnti.	
param	winName	Nome della window
	x	larghezza (opt)
	Y	altezza (opt)
return	Size x, y o 1=successo, 0=errore per Set	
call	sknWin.size("home", 200, 100) ... x,y = sknWin.size("home")	

sknWin.color ()

Modifica o ritorna il colore dell'area di una Window.

sknWin.color ("winName", color)		
	Modifica o ritorna il colore dell'area di una Window. Se si omette il colore la funzione riporta quello corrente.	
param	winName	Nome della window
	Color	colore dell'area attiva della window (opt) (numerico o stringa per colore predefinito)
return	Color o 1=successo, 0=errore per Set	
call	sknWin.color("home","white") ... color = sknWin.color("home")	

sknWin.bgcolor ()

Modifica il colore di sfondo di una Window.

sknWin.bgcolor ("winName", color)		
	Modifica il colore di sfondo di una Window. Se si omette il colore la funzione riporta quello corrente.	
param	winName	Nome della window
	Color	colore di sfondo dell'area attiva della window (opt) (numerico o stringa per colore predefinito)
return	Color o 1=successo, 0=errore per Set	
call	sknWin.bgcolor("home","blue") ... bgColor = sknWin.bgcolor("home")	

sknWin.pos ()

Modifica o ritorna la posizione di una Window.

sknWin.pos ("winName", x, y)		
Modifica o ritorna la posizione di una Window. Se si omettono x e y la funzione riporta la posizione corrente.		
param	winName	Nome della window
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknWin.pos("home", 10, 50) ... print ("\\r\\nmy pos=" .. sknWin.pos("home"))	

sknWin.trasp ()

Controlla la trasparenza di una Window.

sknWin.trasp ("winName", transp)		
Controlla la trasparenza di una Window. Se non viene valorizzato il campo transp, la funzione ritorna il valore corrente di trasparenza della window selezionata.		
param	winName	Nome della window
	Transp	true=window trasparente, false>window opaca (opt)
return	Transparency o 1=successo, 0=errore per Set	
call	sknWin.transp("home", true) ... print ("\\r\\nmy transp=" .. sknWin.transp("home"))	

sknWin.enable ()

Controlla l'abilitazione di una Window.

sknWin.enable ("winName", enable)		
	Controlla l'abilitazione di una Window. Se non viene valorizzato il campo enable, la funzione ritorna il valore corrente di abilitazione della window selezionata.	
param	winName	Nome della window
	enable	true=window abilitata, false=window disabilitata (opt)
return	Enable o 1=successo, 0=errore per Set	
call	<pre>sknWin.enable("home", true) ... print ("\r\nmy enabled=" .. sknWin.enable("home"))</pre>	

sknWin.delete ()

Distrugge una Window.

Questo metodo non rimuove gli oggetti figli della finestra.

sknWin.delete ("winName")		
	Distrugge una Window.	
param	winName	Nome della window
return	1=successo, 0=errore	
call	sknWin.delete("home")	

sknWin.modal ("winName", modal)

Rende modale la Window.

sknWin.modal ("winName", modal)		
	Rende modale la Window.	
param	winName	Nome della window
	modal	true=Window modale, false=standard Window
return	1=successo, 0=errore	
call	sknWin.modal("home", true)	

sknWin.currwin ()

Ritorna il nome della window corrente.

sknWin.currwin ()	
	Ritorna il nome della window corrente.
param	
return	nome window
call	cwin = sknWin.currwin()

sknWin.allowMoveX ()

Controlla l'abilitazione al movimento su asse X di una Window.

sknWin.allowMoveX ("winName", allow)		
Controlla l'abilitazione al movimento su asse X di una Window. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione della window selezionata.		
param	winName	Nome della window
	allow	true=consente il movimento sull'asse x della window, false>window fissa su asse x (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknWin.allowMoveX("home", true) ... print ("\r\nmy allowXmove=" .. sknWin.allowMoveX("home"))	

sknWin.allowMoveY ()

Controlla l'abilitazione al movimento su asse Y di una Window.

sknWin.allowMoveY ("winName", allow)		
	Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione della window selezionata.	
param	winName	Nome della window
	allow	true=consente il movimento sull'asse y della window, false>window fissa su asse y (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknWin.allowMoveY("home", true) ... print ("\r\nmy allowYmove=" .. sknWin.allowMoveY("home"))	

sknWin.allowResize ()

Controlla l'abilitazione alla modifica della dimensione di una Window.

sknWin.allowResize ("winName", allow)		
	Controlla l'abilitazione alla modifica della dimensione di una Window. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di trasparenza della window selezionata.	
param	winName	Nome della window
	allow	true=consente il ridimensionamento della window, false>window non ridimensionabile (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknWin.allowResize("home", true) ... print ("\r\nmy allowResize=" .. sknWin.allowResize("home"))	

sknWin.miniature ()

Configura l'immagine miniatura di una Window.

sknWin.miniature ("winName", "collection", "image")		
	Configura l'immagine miniatura di una Window. La miniatura può essere utilizzata per rappresentare in modo sintetico una finestra.	
param	winName	Nome della window
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknWin.miniature("home", "minia", "winHome")	

sknWin.layout ()

Configura l'immagine di layout di una Window.

sknWin.layout ("winName", "collection", "image")		
	Configura l'immagine di layout di una Window. L'immagine di layout può essere utilizzata per rappresentare in modo sintetico una finestra ed il suo contenuto.	
param	winName	Nome della window
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknWin.layout("home", "minia", "winHomeLay")	

sknWin.bringTop ()

Porta la finestra in primo piano.

sknWin.bringTop ("winName")		
	Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre window attive.	
param	winName	Nome della window
return	1=successo, 0=errore	
call	sknWin.bringTop("home")	

sknWin.bringBot ()

Porta la finestra in sfondo.

sknWin.bringBot ("winName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	winName	Nome della window
return	1=successo, 0=errore	
call	sknWin.bringBot("home")	

sknWin.refresh ()

Forza il ridisegno di una window.

sknWin.refresh ("winName")		
	Forza il ridisegno di una window. Questo metodo viene utilizzato quando il gestore delle finestre non è in grado di determinare un cambiamento di una finestra per eseguirne automaticamente il ridisegno, ad esempio il cambiamento di un flag che durante il ridisegno della finestra coinvolge degli artefatti grafici.	
param	winName	Nome della window
return	1=successo, 0=errore	
call	sknWin.refresh("home")	

sknWin.auxData ()

Controlla l'AuxData di una Window.

sknWin.auxData ("winName", auxData)		
	Controlla l'AuxData di una Window. L'AuxData viene utilizzato per identificare una window con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.	
param	winName	Nome della window
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknWin.auxData("home", 1234) ... print ("\r\nmy auxData=" .. sknWin.auxData("home"))	

sknWin.auxStr ()

Controlla l'AuxString di una Window.

sknWin.auxStr ("winName", auxString)		
Controlla l'AuxString di una Window. L'AuxString viene utilizzato per identificare una window con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	winName	Nome della window
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknWin.auxData("home", "W1") ... print ("\r\nmy auxString=" .. sknWin.auxStr("home"))	

sknWin.group ()

Controlla l'appartenenza di un gruppo di una Window.

sknWin.group ("winName", "grpName")		
Controlla l'appartenenza di un gruppo di una Window. Il grpName viene utilizzato per associare una window con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	winName	Nome della window
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknWin.group("home", "setGrp") ... print ("\r\nmy group=" .. sknWin.group("home"))	

sknWin.subGroup ()

Controlla l'appartenenza di un sotto gruppo di una Window.

sknWin.subGroup ("winName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di una Window. Il subgrpName viene utilizzato per associare una window con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	winName	Nome della window
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknWin.subGroup("home", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknWin.subGroup ("home"))	

sknWin.autoHide ()

Controlla la funzionalità di auto hide di una Window.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknWin.autoHide ("winName", autoHide)		
Controlla la funzionalità di auto hide di una Window. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).		
param	winName	Nome della window
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknWin.autoHide("home", true) ... print ("\r\nmy autohide=" .. sknWin.autoHide ("home"))	

sknWin.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di una Window.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknWin.autoHideTO ("winName", autoHideTimeout)		
Controlla il timeout per la funzionalità di auto hide di una Window. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.		
param	winName	Nome della window
	autoHideTimeout	1-32000 millisecondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	<pre>sknWin.autoHideTO("home", 60) ... print ("\\nmy autohideTO ms =" .. sknWin.autoHideTO ("home"))</pre>	

sknWin.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di una Window.

sknWin.autoHideMode ("winName", "mode")		
Controlla il tipo ti hide per la funzionalità di auto hide di una Window. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.		
param	winName	Nome della window
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	<pre>sknWin.autoHideMode("home", "left") ... print ("\\nmy autohideMode=" .. sknWin.autoHideMode ("home"))</pre>	

sknWin.blinkEn ()

Controlla la funzionalità di blink di una Window.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknWin.blinkEn ("winName", blink)		
Controlla la funzionalità di blink di una Window. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	winName	Nome della window
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknWin.blinkEn("home", true) ... print ("\r\nmy blinkEn=" .. sknWin.blinkEn ("home"))	

sknWin.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di una Window.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknWin.blinkCustEn ("winName", blink)		
Controlla la funzionalità di blink custom o globale di una Window. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	winName	Nome della window
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknWin.blinkCustEn("home", true) ... print ("\r\nmy blinkCustEn=" .. sknWin.blinkCustEn ("home"))	

sknWin.autoHideCBshow ()

Controlla la funzione di callback show di una Window.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknWin.autoHideCBshow ("winName", "callback")

	Controlla la funzione di callback show di una Window. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	winName	Nome della window
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknWin.autoHideCBshow("home", "mycCBshow") ... print ("\r\nmy autohideCBshow=" .. sknWin.autoHideCBshow ("home"))	

sknWin.autoHideCBhide ()

Controlla la funzione di callback hide di una Window.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknWin.autoHideCBhide ("winName", "callback")

	Controlla la funzione di callback hide di una Window. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	winName	Nome della window
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknWin.autoHideCBhide("home", "mycCBshow") ... print ("\r\nmy autohideCBhide=" .. sknWin.autoHideCBhide ("home"))	

sknWin.aniEn ()

Controlla la funzionalità di animation di una Window.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknWin.aniEn ("winName", animation)		
Controlla la funzionalità di animation di una Window. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.		
param	winName	Nome della window
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	sknWin.aniEn("home", true) ... print ("\r\nmy aniEn=" .. sknWin.aniEn ("home"))	

sknWin.aniMode ()

Controlla la modalità di animation di una Window.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknWin.aniMode ("winName", "aniMode")		
Controlla la modalità di animation di una Window. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	winName	Nome della window
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	sknWin.aniMode("home", "shift") ... print ("\r\nmy aniMode=" .. sknWin.aniMode ("home"))	

sknWin.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknWin.autoFocusChild ("winName", autoFocus)

	Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.	
param	winName	Nome della window
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknWin.autoFocusChild("home", true) ... print ("\r\nmy autoFocusChild=" .. sknWin.autoFocusChild ("home"))	

sknWin.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknWin.bringToTopChild ("winName", en)

	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	winName	Nome della window
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknWin.bringToTopChild("home", true) ... print ("\r\nmy bringToTopChild=" .. sknWin.bringToTopChild ("home"))	

sknWin.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknWin.longTouchCB ("winName", "callback")		
	Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	winName	Nome della window
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknWin.longTouchCB("home", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknWin.longTouchCB ("home"))	

sknWin.paintCB ()

Controlla il nome della funzione di callback per il paint della window.

sknWin.paintCB ("winName", "callback")		
	Controlla il nome della funzione di callback per il paint della window. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	winName	Nome della window
	Callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknWin.paintCB("home", "RedrawCb") ... print ("\r\nmy paintCB=" .. sknWin.paintCB ("home"))	

sknWin.keyPressCB ()

Controlla il nome della funzione di callback per il key press della window.

sknWin.keyPressCB ("winName", "callback")		
	Controlla il nome della funzione di callback per il key press della window. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	winName	Nome della window
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknWin.keyPressCB("home", "keyPressWin") ... print ("\r\nmy winKeyPressCB=" .. sknWin.keyPressCB ("home"))	

sknWin.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release della window.

sknWin.keyReleaseCB ("winName", "callback")		
	Controlla il nome della funzione di callback per il key release della window. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	winName	Nome della window
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknWin.keyReleaseCB("home", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknWin.keyReleaseCB ("home"))	

sknWin.gestureCB ()

Controlla il nome della funzione di callback per le gesture della window.

sknWin.gestureCB ("winName", "gesture", "callback")		
	Controlla il nome della funzione di callback per le gesture della window. Se viene omesso il campo cb, viene ritornato il nome della funzione di callback relativa al campo gesture.	
param	winName	Nome della window
	gesture	tipo di gestione {"up", "down", "left", "right"}
	callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknWin.gestureCB("home", "up", "gestureUpCb") ... print ("\r\nmy gestureCB:up=" .. sknWin.gestureCB ("home", "up"))	

L A B E L

La Classe **sknLbl** implementa le funzionalità della Label.

Membro	Descrizione
Control	
sknLbl.create()	Crea una Label
sknLbl.delete()	Distrugge una Label
View	
sknLbl.show()	Visualizza una Label
sknLbl.hide()	Nasconde una Label
Proprietà	
sknLbl.size()	Modifica le dimensioni di una Label
sknLbl.colors()	Modifica i colori di una Label
sknLbl.text()	Modifica il testo di una Label
sknLbl.transp()	Controlla la trasparenza di una Label
sknLbl.font()	Configura il font di una Label
sknLbl.txtalign()	Configura l'allineamento di una Label
sknLbl.wrapmode()	Configura il Wrap mode di una Label
sknLbl.pos()	Modifica la posizione di una Label
sknLbl.image()	Configura l'immagine di una Label
sknLbl.sizetoimage()	Configura l'autosize dell'immagine di una Label
sknLbl.txtOffset()	Configura loffset di partenza del testo di una Label
Permessi	
sknLbl.allowMoveX()	Controlla l'abilitazione al movimento su asse X di una Label
sknLbl.allowMoveY()	Controlla l'abilitazione al movimento su asse Y di una Label
sknLbl.allowResize()	Controlla l'abilitazione alla modifica della dimensione di una Label
Visibilità	
sknLbl.bringTop()	Porta la label in primo piano
sknLbl.bringBot()	Porta la label in sfondo
Aggregazione	
sknLbl.auxData()	Controlla l'AuxData di una Label
sknLbl.auxString()	Controlla l'AuxString di una Label
sknLbl.group()	Controlla l'appartenenza di un gruppo di una Label
sknLbl.subGroup()	Controlla l'appartenenza di un sotto gruppo di una Label

Eventi	
sknLbl.keyPressCB()	Controlla il nome della funzione di callback di key press della Label
sknLbl.keyReleaseCB()	Controlla il nome della funzione di callback di key release della Label
Auto Hide	
sknLbl.autoHide()	Controlla la funzionalità di auto hide di una Label
sknLbl.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di una Label
sknLbl.autoHideMode()	Controlla il tipo ti hide per la funzionalità di auto hide di una Label
Blink	
sknLbl.blinkEn()	Controlla la funzionalità di blink di una Label
sknLbl.blinkCustEn()	Controlla la funzionalità di blink custom o globale di una Label
Animazione	
sknLbl.aniEn()	Controlla la funzionalità di animation di una Label
sknLbl.aniMode()	Controlla la modalità di animation di una Label
Child	
sknLbl.autoFocusChild()	Controlla la funzionalità di autofocus per le child Label
sknLbl.bringToTopChild()	Controlla la funzionalità di brig to top per le child Label
Callbacks	
sknLbl.longTouchCB()	Controlla il nome della funzione di callback per un long touch
sknLbl.paintCB()	Controlla il nome della funzione di callback per il paint dell'oggetto
Sound	
sknLbl.sound()	Configura il suono per l'evento press

Classe [sknLbl]

sknLbl.create ()

Crea una Label.

sknLbl.create ("labelName", "motherWin", xpos, ypos, xsize, ysize, "textAlign", "text")		
	Crea una Label.	
param	labelName	Nome della label
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza della label
	ysize	altezza della label
	textAlign	allineamento {"HorLeft_Vert_Top", "HorLeft_VertBot", "HorLeft_VertCenter", "HorRight_Vert_Top", "HorRight_VertBot", "HorRight_VertCenter", "HorCenter_Vert_Top", "HorCenter_VertBot", "HorCenter_VertCenter"}
	text	Testo della Label
return	1=successo, 0=errore	
call	sknLbl.create("lblHelp", "winHome", 0, 0, 480, 272, "HorLeft_VertCenter", "myLabel")	

sknLbl.show ()

Visualizza una Label.

sknLbl.show ("lblName")		
	Visualizza una Label.	
param	lblName	nome della label
return	1=successo, 0=errore	
call	sknLbl.show("myLbl")	

sknLbl.hide ()

Nasconde una Label.

sknLbl.hide ("lblName")		
	Nasconde una Label.	
param	lblName	Nome della label
return	1=successo, 0=errore	
call	sknLbl.hide("myLbl")	

sknLbl.size ()

Modifica o ritorna le dimensioni di una Label.

sknLbl.size ("lblName", x, y)		
	Modifica o ritorna le dimensioni di una Label. Se si omette x e y la funzione riporta quelli correnti.	
param	lblName	Nome della label
	x	larghezza (opt)
	Y	altezza (opt)
return	Size x, y o 1=successo, 0=errore per Set	
call	sknLbl.size("myLbl", 200, 100) ... x,y = sknLbl.size("myLbl")	

sknLbl.colors ()

Modifica i colori di una Label.

sknLbl.colors ("lblName", color, BGcolor)		
	Modifica i colori di una Label.	
param	lblName	Nome della label
	color	colore del testo (numerico o stringa per colore predefinito)
	BGcolor	colore di sfondo del testo (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknLbl.colors("lblHlp", "blackE", "white")	

sknLbl.text ()

Modifica il testo di una Label.

sknLbl.text ("lblName", "text")		
	Modifica o recupera il testo di una Label.	
param	lblName	Nome della label
	text	testo della label (opt)
return	label text o 1=successo, 0=errore per set	
call	sknLbl.text("lblHlp", "Hello World") ... print (sknLbl.text("lblHlp"))	

sknLbl.font ()

Configura il font di una Label.

sknLbl.font ("lblName", "fontName")		
	Configura il font di una Label.	
param	lblName	Nome della label
	fontName	nome font
return	1=successo, 0=errore	
call	sknLbl.font("lblHlp", "Arial14")	

sknLbl.txtalign ()

Configura l'allineamento di una Label.

sknLbl.txtalign ("lblName", "align")		
	Configura l'allineamento di una Label.	
param	lblName	Nome della label
	Align	allineamento {"HorLeft_Vert_Top", "HorLeft_VertBot", "HorLeft_VertCenter", "HorRight_Vert_Top", "HorRight_VertBot", "HorRight_VertCenter", "HorCenter_Vert_Top", "HorCenter_VertBot", "HorCenter_VertCenter"}
return	1=successo, 0=errore	
call	sknLbl.txtalign("lblHlp", "HorLeft_Vert_Top")	

sknLbl.wrapmode ()

Configura il Wrap mode di una Label.

sknLbl.wrapmode ("lblName", "wrapMode")		
	Configura il Wrap mode di una Label.	
param	lblName	Nome della label
	wrapMode	wrap mode {"none", "word", "char"}
return	1=successo, 0=errore	
call	sknLbl.wrapmode("lblHlp", "word")	

sknLbl.pos ()

Modifica o ritorna la posizione di una Label.

sknLbl.pos ("lblName", x, y)		
Modifica o ritorna la posizione di una Label. Se si omettono x e y la funzione riporta la posizione corrente.		
param	lblName	Nome della label
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknLbl.pos("myLbl", 10, 50) ... print ("r\nmy pos=" .. sknLbl.pos("myLbl"))	

sknLbl.image ()

Configura l'immagine di una Label.

sknLbl.image ("lblName", "collection", "image")		
Configura l'immagine di una Label. Una label può visualizzare una immagine nel background del testo.		
param	lblName	Nome della label
	collection	nome della collezione di immagini
	Image	nome dell'immagine
return	1=successo, 0=errore	
call	sknLabl.image("lablInfo", "icons", "alert")	

sknLbl.sizeToImage ()

Configura l'autosize dell'immagine di una Label.

sknLbl.sizeToImage ("lblName", autosize)		
	Configura l'autosize dell'immagine di una Label. Con questa opzione non è necessario impostare nella label le esatte dimensione dell'immagine che contiene (normalmente poste a 0 di default).	
param	lblName	Nome della label
	Autosize	true=label autosize
return	1=successo, 0=errore	
call	sknLbl.sizeToImage("hmlLbl", true)	

sknLbl.txtOffset ()

Configura l'offset di partenza del testo di una Label.

sknLbl.txtOffset ("lblName", x, y)		
	Configura l'offset di partenza del testo di una Label.	
param	lblName	Nome della label
	x	offset asse x
	y	offset asse y
return	1=successo, 0=errore	
call	sknLbl.txtOffset("hmlLbl", 10, 50)	

sknLbl.delete ()

Distrugge una Label.

sknLbl.delete ("lblName")		
	Distrugge una Label.	
param	lblName	Nome della label
return	1=successo, 0=errore	
call	sknLbl.delete("lblHelp")	

sknLbl.allowMoveX ()

Controlla l'abilitazione al movimento su asse X di una Label.

sknLbl.allowMoveX ("lblName", allow)		
Controlla l'abilitazione al movimento su asse X di una Label. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione della label selezionata.		
param	lblName	Nome della label
	allow	true=consente il movimento sull'asse x della label, false=window fissa su asse x (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknLbl.allowMoveX("myLbl", true) ... print ("\r\nmy allowXmove=" .. sknLbl.allowMoveX("myLbl"))	

sknLbl.allowMoveY ()

Controlla l'abilitazione al movimento su asse Y di una Label.

sknLbl.allowMoveY ("lblName", allow)		
Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione della label selezionata.		
param	lblName	Nome della label
	allow	true=consente il movimento sull'asse y della label, false=window fissa su asse y (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknLbl.allowMoveY("myLbl", true) ... print ("\r\nmy allowYmove=" .. sknLbl.allowMoveY("myLbl"))	

sknLbl.allowResize ()

Controlla l'abilitazione alla modifica della dimensione di una Label.

sknLbl.allowResize ("lblName", allow)		
Controlla l'abilitazione alla modifica della dimensione di una Label. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di trasparenza della label selezionata.		
param	lblName	Nome della label
	allow	true=consente il ridimensionamento della label, false=window non ridimensionabile (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknLbl.allowResize("myLbl", true) ... print ("\\r\\nmy allowResize=" .. sknLbl.allowResize("myLbl"))	

sknLbl.keyPressCB ()

Controlla il nome della funzione di callback per il key press della label.

sknLbl.keyPressCB ("lblName", "callback")		
Controlla il nome della funzione di callback per il key press della label. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	lblName	Nome della label
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknLbl.keyPressCB("myLbl", "keyPressWin") ... print ("\\r\\nmy winKeyPressCB=" .. sknLbl.keyPressCB ("myLbl"))	

sknLbl.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release della label.

sknLbl.keyReleaseCB ("lblName", "callback")		
	Controlla il nome della funzione di callback per il key press della label. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	lblName	Nome della label
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknLbl.keyReleaseCB("myLbl", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknLbl.keyReleaseCB ("myLbl"))	

sknLbl.sound ()

Configura il suono per l'evento press.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknLbl.sound ("lblName", "soundName")		
	Configura il suono per l'evento press.	
param	lblName	Nome della label
	soundName	Nome suono
return	1=successo, 0=errore	
call	sknLbl.sound("closeLbl", "soundEnd")	

sknLbl.trasp()

Controlla la trasparenza di una Label.

sknLbl.trasp ("lblName", transp)		
	Controlla la trasparenza di una Label. Se non viene valorizzato il campo transp, la funzione ritorna il valore corrente di trasparenza della label selezionata.	
param	lblName	Nome della label
	Transp	true=window trasparente, false>window opaca (opt)
return	Tranparency o 1=successo, 0=errore per Set	
call	sknLbl.transp("myLbl", true) ... print ("\r\nmy transp=" .. sknLbl.transp("myLbl"))	

sknLbl.auxData()

Controlla l'AuxData di una Label.

sknLbl.auxData ("lblName", auxData)		
	Controlla l'AuxData di una Label. L'AuxData viene utilizzato per identificare una Label con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.	
param	lblName	Nome della label
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknLbl.auxData("myLbl", 1234) ... print ("\r\nmy auxData=" .. sknLbl.auxData("myLbl"))	

sknLbl.auxStr ()

Controlla l'AuxString di una Label.

sknLbl.auxStr ("lblName", auxString)		
Controlla l'AuxString di una Label. L'AuxString viene utilizzato per identificare una Label con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	lblName	Nome della label
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknLbl.auxData("myLbl", "W1") ... print ("\r\nmy auxString=" .. sknLbl.auxStr("myLbl"))	

sknLbl.bringTop ()

Porta la finestra in primo piano.

sknLbl.bringTop ("lblName")		
Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre window attive.		
param	lblName	Nome della label
return	1=successo, 0=errore	
call	sknLbl.bringTop("myLbl")	

sknLbl.bringBot ()

Porta la finestra in sfondo.

sknLbl.bringBot ("lblName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	lblName	Nome della label
return	1=successo, 0=errore	
call	sknLbl.bringBot("myLbl")	

sknLbl.autoHide ()

Controlla la funzionalità di auto hide di una Label.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknLbl.autoHide ("lblName", autoHide)		
	Controlla la funzionalità di auto hide di una Label. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	lblName	Nome della label
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknLbl.autoHide("myLbl", true) ... print ("\r\nmy autohide=" .. sknLbl.autoHide ("myLbl"))	

sknLbl.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di una Label.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknLbl.autoHideTO ("lblName", autoHideTimeout)

	Controlla il timeout per la funzionalità di auto hide di una Label. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	lblName	Nome della label
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	<pre>sknLbl.autoHideTO("myLbl", 60) ... print ("\\r\\nmy autohideTO=" .. sknLbl.autoHideTO ("myLbl"))</pre>	

sknLbl.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di una Label.

sknLbl.autoHideMode ("lblName", "mode")

	Controlla il tipo ti hide per la funzionalità di auto hide di una Label. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.	
param	lblName	Nome della label
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	<pre>sknLbl.autoHideMode("myLbl", "left") ... print ("\\r\\nmy autohideMode=" .. sknLbl.autoHideMode ("myLbl"))</pre>	

sknLbl.autoHideCBshow ()

Controlla la funzione di callback show di una Label.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknLbl.autoHideCBshow ("lblName", "callback")

	Controlla la funzione di callback show di una Label. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	lblName	Nome della label
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknLbl.autoHideCBshow("myLbl", "myCBshow") ... print ("\\r\\nmy autohideCBshow=" .. sknLbl.autoHideCBshow ("myLbl"))</pre>	

sknLbl.autoHideCBhide ()

Controlla la funzione di callback hide di una Label.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknLbl.autoHideCBhide ("lblName", "callback")

	Controlla la funzione di callback hide di una Label. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	lblName	Nome della label
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknLbl.autoHideCBhide("myLbl", "myCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknLbl.autoHideCBhide ("myLbl"))</pre>	

sknLbl.blinkEn ()

Controlla la funzionalità di blink di una Label.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknLbl.blinkEn ("lblName", blink)		
Controlla la funzionalità di blink di una Label. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	lblName	Nome della label
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknLbl.blinkEn("myLbl", true) ... print ("\r\nmy blinkEn=" .. sknLbl.blinkEn ("myLbl"))	

sknLbl.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di una Label.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknLbl.blinkCustEn ("lblName", blink)		
Controlla la funzionalità di blink custom o globale di una Label. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	lblName	Nome della label
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknLbl.blinkCustEn("myLbl", true) ... print ("\r\nmy blinkCustEn=" .. sknLbl.blinkCustEn ("myLbl"))	

sknLbl.aniEn ()

Controlla la funzionalità di animation di una Label.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknLbl.aniEn ("lblName", animation)		
Controlla la funzionalità di animation di una Label. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.		
param	lblName	Nome della label
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknLbl.aniEn("myLbl", true) ... print ("\r\nmy aniEn=" .. sknLbl.aniEn ("myLbl"))</pre>	

sknLbl.aniMode ()

Controlla la modalità di animation di una Label.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknLbl.aniMode ("lblName", "aniMode")		
Controlla la modalità di animation di una Label. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	lblName	Nome della label
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	<pre>sknLbl.aniMode("myLbl", "shift") ... print ("\r\nmy aniMode=" .. sknLbl.aniMode ("myLbl"))</pre>	

sknLbl.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknLbl.autoFocusChild ("lblName", autoFocus)		
	Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.	
param	lblName	Nome della label
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknLbl.autoFocusChild("myLbl", true) ... print ("\r\nmy autoFocusChild=" .. sknLbl.autoFocusChild ("myLbl"))	

sknLbl.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknLbl.bringToTopChild ("lblName", en)		
	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	lblName	Nome della label
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknLbl.bringToTopChild("myLbl", true) ... print ("\r\nmy bringToTopChild=" .. sknLbl.bringToTopChild ("myLbl"))	

sknLbl.group ()

Controlla l'appartenenza di un gruppo di una Label.

sknLbl.group ("lblName", "grpName")		
Controlla l'appartenenza di un gruppo di una Label. Il grpName viene utilizzato per associare una Label con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	lblName	Nome della label
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknLbl.group("myLbl", "setGrp") ... print ("\r\nmy group=" .. sknLbl.group("myLbl"))	

sknLbl.subGroup ()

Controlla l'appartenenza di un sotto gruppo di una Label.

sknLbl.subGroup ("lblName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di una Label. Il subgrpName viene utilizzato per associare una Label con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	lblName	Nome della label
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknLbl.subGroup("myLbl", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknLbl.subGroup ("myLbl"))	

sknLbl.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknLbl.longTouchCB ("lblName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	lblName	Nome della label
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknLbl.longTouchCB("myLbl", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknLbl.longTouchCB ("myLbl"))	

sknLbl.paintCB ()

Controlla il nome della funzione di callback per il paint della label.

sknLbl.paintCB ("lblName", "callback")		
Controlla il nome della funzione di callback per il paint della label. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	lblName	Nome della label
	Callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknLbl.paintCB("home", "RedrawCb") ... print ("\r\nmy paintCB=" .. sknLbl.paintCB ("home"))	

BUTTON

La Classe **sknBtn** gestisce i Button standard.

Membro	Descrizione
Control	
sknBtn.create()	Crea un Button
sknBtn.delete()	Distrugge un Button
sknBtn.enable()	Abilita un Button
View	
sknBtn.show()	Visualizza un Button
sknBtn.hide()	Nasconde un Button
Proprietà	
sknBtn.size()	Modifica le dimensioni di un Button
sknBtn.txtcolors()	Modifica i colori del testo di un Button
sknBtn.txtBGcolors()	Modifica i colori del sfondo del testo di un Button
sknBtn.text()	Modifica il testo di un Button
sknBtn.transp()	Controlla la trasparenza di un Button
sknBtn.font()	Configura il font di un Button
sknBtn.txtalign()	Configura l'allineamento di un Button
sknBtn.pos()	Modifica la posizione di un Button
sknBtn.txtOffset()	Configura loffset di partenza del testo di un Button
sknBtn.imgunpress()	Configura l'immagine di un Button non premuto
sknBtn.imgpress()	Configura l'immagine di un Button premuto
sknBtn.imgPos()	Configura la posizione dell'immagine di un Button
sknBtn.focussable()	Configura la funzionalità focussable di un Button
Skin	
sknBtn.skineffect()	Configura gli effetti grafici aggiuntivi di un Button
sknBtn.skintype()	Configura il tipo di skin di un Button
sknBtn.skinradius()	Configura il raggio di curvatura dello skin di un Button
sknBtn.skinbordersize()	Configura il bordo dello skin di un Button
sknBtn.skinbordercolor()	Configura il colore del bordo dello skin di un Button
Permessi	
sknBtn.allowMoveX()	Controlla l'abilitazione al movimento su asse X di un Button
sknBtn.allowMoveY()	Controlla l'abilitazione al movimento su asse Y di un Button
sknBtn.allowResize()	Controlla l'abilitazione alla modifica della dimensione di un Button

Visibilità	
sknBtn.knBtn.bringTop()	Porta il Button in primo piano
sknBtn.bringBot()	Porta il Button in sfondo
Aggregazione	
sknBtn.auxData()	Controlla l'AuxData di un Button
sknBtn.auxStr()	Controlla l'AuxString di un Button
sknBtn.group()	Controlla l'appartenenza di un gruppo di un Button
sknBtn.subGroup()	Controlla l'appartenenza di un sotto gruppo di un Button
Eventi	
sknBtn.keyPressCB()	Controlla il nome della funzione di callback di key press di un Button
sknBtn.keyReleaseCB()	Controlla il nome della funzione di callback di key release di un Button
Auto Hide	
sknBtn.autoHide()	Controlla la funzionalità di auto hide di un Button
sknBtn.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di un Button
sknBtn.autoHideMode()	Controlla il tipo ti hide per la funzionalità di auto hide di un Button
Blink	
sknBtn.blinkEn()	Controlla la funzionalità di blink di un Button
sknBtn.blinkCustEn()	Controlla la funzionalità di blink custom o globale di un Button
Auto Hide callbacks	
sknBtn.autoHideCBshow()	Controlla la funzione di callback hide di un Button
sknBtn.autoHideCBhide()	Controlla la funzione di callback show di un Button
Animazione	
sknBtn.aniEn()	Controlla la funzionalità di animation di un Button
sknBtn.aniMode()	Controlla la modalità di animation di un Button
Child	
sknBtn.autoFocusChild()	Controlla la funzionalità di autofocus per i child Button
sknBtn.bringToTopChild()	Controlla la funzionalità di brig to top per i child Button
Callbacks	
sknBtn.longTouchCB()	Controlla il nome della funzione di callback per un long touch
sknBtn.paintCB()	Controlla il nome della funzione di callback per il paint dell'oggetto
Sound	
sknBtn.sound()	Configura il suono per l'evento press

Classe [sknBtn]

sknBtn.create()

Crea un Button.

sknBtn.create("btnName", "motherWin", xpos, ypos, xsize, ysize)		
	Crea un Button.	
param	btnName	Nome del Button
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza del button
	ysize	altezza del button
return	1=successo, 0=errore	
call	sknBtn.create("btnInfo", "winHome", 30, 50, 42, 42)	

sknBtn.show()

Visualizza un Button.

sknBtn.show ("btnName")		
	Visualizza un Button.	
param	btnName	nome del Button
return	1=successo, 0=errore	
call	sknBtn.show("myBtn")	

sknBtn.hide()

Nasconde un Button.

sknBtn.hide ("btnName")		
	Nasconde un Button.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknBtn.hide("myBtn")	

sknBtn.size ()

Modifica o ritorna le dimensioni di un Button.

sknBtn.size ("btnName", x, y)		
	Modifica o ritorna le dimensioni di un Button. Se si omette x e y la funzione riporta quelli correnti.	
param	btnName	Nome del Button
	x	larghezza (opt)
	Y	altezza (opt)
return	Size x, y o 1=successo, 0=errore per Set	
call	sknBtn.size("myBtn", 200, 100) ... x,y = sknBtn.size("myBtn")	

sknBtn.pos ()

Modifica o ritorna la posizione di un Button.

sknBtn.pos ("btnName", x, y)		
	Modifica o ritorna la posizione di un Button. Se si omettono x e y la funzione riporta la posizione corrente.	
param	btnName	Nome del Button
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknBtn.pos("myBtn", 10, 50) ... print ("\r\nmy pos=" .. sknBtn.pos("myBtn"))	

sknBtn.txtcolors()

Modifica i colori del testo di un Button.

sknBtn.txtcolors("btnName", colorUnpress, colorPress)		
	Modifica i colori del testo di un Button.	
param	btnName	Nome del Button
	colorUnpress	colore del testo del button non premuto (numerico o stringa per colore predefinito)
	colorPress	colore del testo del button premuto (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknBtn.txtcolors("btnHlp", "black", "white")	

sknBtn.txtBGcolors()

Modifica i colori del di sfondo del testo di un Button.

sknBtn.txtBGcolors("btnName", colorUnpress, colorPress)		
	Modifica i colori del di sfondo del testo di un Button.	
param	btnName	Nome del Button
	colorUnpress	colore di sfondo del testo del button non premuto (numerico o stringa per colore predefinito)
	colorPress	colore di sfondo del testo del button premuto (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknBtn.txtBGcolors("btnHlp", "blue", "magenta")	

sknBtn.txtalign()

Configura l'allineamento di un Button.

sknBtn.txtalign("btnName", "align")		
	Configura l'allineamento di un Button.	
param	btnName	Nome del Button
	align	allineamento {"Hor_Vert", "Hor_Top", "Hor_Bottom", "Hor_Baseline", "Hor_Vcenter", "Left_Vert", "Left_Top", "Left_Bottom", "Left_Baseline", "Left_Vcenter", "Right_Vert", "Right_Top", "Right_Bottom", "Right_Baseline", "Right_Vcenter", "Center_Vert", "Center_Top", "Center_Bottom", "Center_Baseline", "Center_Vcenter", "Hcenter_Vert", "Hcenter_Top", "Hcenter_Bottom", "Hcenter_Baseline", "Hcenter_Vcenter"}
return	1=successo, 0=errore	
call	sknBtn.txtalign("btnHlp", "HorLeft_Vert_Top")	

sknBtn.txtOffset()

Configura l'offset di partenza del testo di un Button.

sknBtn.txtOffset("objName", x, y)		
	Configura l'offset di partenza del testo di un Button.	
param	btnName	Nome del Button
	x	offset asse x
	y	offset asse y
return	1=successo, 0=errore	
call	sknBtn.txtOffset("hmlBtn", 10, 50)	

sknBtn.text()

Modifica il testo di un Button.

sknBtn.text("btnName", "text")		
	Modifica il testo di un Button.	
param	btnName	Nome del Button
	text	testo del button
return	1=successo, 0=errore	
call	sknBtn.text("btnHlp", "HELP")	

sknBtn.font()

Configura il font di un Button.

sknBtn.font("btnName", "fontName")		
	Configura il font di un Button.	
param	btnName	Nome del Button
	fontName	nome font
return	1=successo, 0=errore	
call	sknBtn.font("btnHlp", "Arial14")	

sknBtn.trasp()

Controlla la trasparenza di un Button.

sknBtn.trasp ("btnName", transp)		
	Controlla la trasparenza di un Button. Se non viene valorizzato il campo transp, la funzione ritorna il valore corrente di trasparenza del Button selezionata.	
param	btnName	Nome del Button
	Transp	true=Button trasparente, false=Button opaca (opt)
return	Transparency o 1=successo, 0=errore per Set	
call	sknBtn.transp("myBtn", true) ... print ("\r\nmy transp=" .. sknBtn.transp("myBtn"))	

sknBtn.focussable()

Configura la funzionalità focussable di un Button.

sknBtn.focussable("btnName", focussable)

	Configura la funzionalità focussable di un Button. Questa funzionalità è utile quando si vuole scorrere automaticamente fra vari oggetti senza un touch esplicito.	
param	btnName	Nome del Button
	focussable	true=focussable, false=non focussable
return	1=successo, 0=errore	
call	sknBtn.focussable("btnInfo", false)	

sknBtn.imgunpress()

Configura l'immagine di un Button non premuto.

sknBtn.imgunpress("btnName", imageIndex, "collection", "image")

	Configura l'immagine di un Button non premuto. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.	
param	btnName	Nome del Button
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknBtn.imgunpress("btnInfo", 0, "icons", "alert")	

sknBtn.imgpress()

Configura l'immagine di un Button premuto.

sknBtn.imgpress("btnName", "imageIndex", "collection", "image")

	Configura l'immagine di un Button premuto. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button	
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)	
	collection	nome della collezione di immagini	
	image	nome dell'immagine	
return	1=successo, 0=errore		
call	sknBtn.imgpress("btnInfo", 0, "icons", "alert")		

sknBtn.imgdisabled()

Configura l'immagine di un Button disabilitato.

sknBtn.imgdisabled("btnName", "imageIndex", "collection", "image")

	Configura l'immagine di un Button disabilitato. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button	
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)	
	collection	nome della collezione di immagini	
	image	nome dell'immagine	
return	1=successo, 0=errore		
call	sknBtn.imgdisabled("btnInfo", 0, "icons", "alert")		

sknBtn.imgPos()

Configura la posizione dell'immagine di un Button.

sknBtn.imgPos("btnName", imgIndex, xpos, ypos)

	Configura la posizione dell'immagine di un Button. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button	
	imgIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)	
	xpos	posizione x all'interno dell'area del button	
	ypos	posizione y all'interno dell'area del button	
return	1=successo, 0=errore		
call	sknBtn.imgPos("btnInfo", 0, 10, 34)		

sknBtn.enable()

Controlla l'abilitazione di un Button.

sknBtn.enable("btnName", enable)

	Controlla l'abilitazione di un Button. Se non viene valorizzato il campo enable, la funzione ritorna il valore corrente di abilitazione del Button selezionata.		
param	btnName	Nome del Button	
	enable	true=Button abilitata, false=Button disabilitata (opt)	
return	Enable o 1=successo, 0=errore per Set		
call	sknBtn.enable("myBtn", true) ... print ("\\r\\nmy enabled=" .. sknBtn.enable("myBtn"))		

sknBtn.skineffect()

Configura gli effetti grafici aggiuntivi di un Button.

sknBtn.skineffect("btnName", "skineffect")		
	Configura gli effetti grafici aggiuntivi di un Button. Questa funzionalità permette di dare al button un aspetto completamente differente ad un tradizionale button window.	
param	btnName	Nome del Button
	skineffect	effetto Skinner {"none", "flat", "shadow", "Bott_shadow", "vawe_shadow", "glittering", "sparkling"}
return	1=successo, 0=errore	
call	sknBtn.skineffect("btnInfo", "flat")	

sknBtn.skintype()

Configura il tipo di skin di un Button.

sknBtn.skintype("btnName", "skintype")		
	Configura il tipo di skin di un Button. Questa funzionalità permette di dare al button un aspetto completamente differente ad un tradizionale button window.	
param	btnName	Nome del Button
	skintype	effetto Skinner {"none", "round", "half_up", "half_bottom", "half_right", "half_left", "quart_up_left", "quart_up_right", "quart_btm_right", "quart_btm_left", "elliptic", "rect", "image_trasp"}
return	1=successo, 0=errore	
call	sknBtn.skintype("btnInfo", "round")	

sknBtn.skinradius()

Configura il raggio di curvatura dello skin di un Button.

sknBtn.skinradius("btnName", skinradius)		
Configura il raggio di curvatura dello skin di un Button. Questa funzionalità è utilizzata nei button che prevedono il disegno dei bordi arrotondati.		
param	btnName	Nome del Button
	skinradius	numero di pixels di raggio
return	1=successo, 0=errore	
call	sknBtn.skinradius("btnInfo", 5)	

sknBtn.skinbordersize()

Configura il bordo dello skin di un Button.

sknBtn.skinbordersize("btnName", bordersize)		
Configura il bordo dello skin di un Button. Questa funzionalità è utilizzata nei button che prevedono il disegno dei bordi.		
param	btnName	Nome del Button
	bordersize	numero di pixels del bordo
return	1=successo, 0=errore	
call	sknBtn.skinbordersize("btnInfo", 5)	

sknBtn.skinbordercolor()

Configura il colore del bordo dello skin di un Button.

sknBtn.skinbordercolor("btnName", color)		
Configura il colore del bordo dello skin di un Button. Questa funzionalità è utilizzata nei button che prevedono il disegno dei bordi.		
param	btnName	Nome del Button
	color	colore del bordo (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknBtn.skinbordercolor("btnInfo", "blue")	

sknBtn.keyPressCB ()

Controlla il nome della funzione di callback per il key press del Button.

sknBtn.keyPressCB ("btnName", "callback")		
	Controlla il nome della funzione di callback per il key press del Button. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknBtn.keyPressCB("myBtn", "keyPressWin") ... print ("\r\nmy winKeyPressCB=" .. sknBtn.keyPressCB ("myBtn"))	

sknBtn.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release del Button.

sknBtn.keyReleaseCB ("btnName", "callback")		
	Controlla il nome della funzione di callback per il key release del Button. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknBtn.keyReleaseCB("myBtn", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknBtn.keyReleaseCB ("myBtn"))	

sknBtn.sound ()

Configura il suono per l'evento press.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknBtn.sound ("lblName", "soundName")		
	Configura il suono per l'evento press.	
param	lblName	Nome del Button
	soundName	Nome suono
return	1=successo, 0=errore	
call	sknBtn.sound("closeBtn", "soundEnd")	

sknBtn.allowMoveX ()

Controlla l'abilitazione al movimento su asse X di un Button.

sknBtn.allowMoveX ("btnName", allow)		
	Controlla l'abilitazione al movimento su asse X di un Button. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione del Button selezionata.	
param	btnName	Nome del Button
	allow	true=consente il movimento sull'asse x del Button, false=Button fissa su asse x (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknBtn.allowMoveX("myBtn", true) ... print ("\r\nmy allowXmove=" .. sknBtn.allowMoveX("myBtn"))	

sknBtn.allowMoveY ()

Controlla l'abilitazione al movimento su asse Y di un Button.

sknBtn.allowMoveY ("btnName", allow)		
Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione del Button selezionata.		
param	btnName	Nome del Button
	allow	true=consente il movimento sull'asse y del Button, false=Button fissa su asse y (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknBtn.allowMoveY("myBtn", true) ... print ("\r\nmy allowYmove=" .. sknBtn.allowMoveY("myBtn"))	

sknBtn.allowResize ()

Controlla l'abilitazione alla modifica della dimensione di un Button.

sknBtn.allowResize ("btnName", allow)		
Controlla l'abilitazione alla modifica della dimensione di un Button. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di trasparenza del Button selezionata.		
param	btnName	Nome del Button
	allow	true=consente il ridimensionamento del Button, false=Button non ridimensionabile (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknBtn.allowResize("myBtn", true) ... print ("\r\nmy allowResize=" .. sknBtn.allowResize("myBtn"))	

sknBtn.auxData ()

Controlla l'AuxData di un Button.

sknBtn.auxData ("btnName", auxData)		
Controlla l'AuxData di un Button. L'AuxData viene utilizzato per identificare un Button con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	<pre>sknBtn.auxData("myBtn", 1234) ... print ("\\r\\nmy auxData=" .. sknBtn.auxData("myBtn"))</pre>	

sknBtn.auxStr ()

Controlla l'AuxString di un Button.

sknBtn.auxStr ("btnName", auxString)		
Controlla l'AuxString di un Button. L'AuxString viene utilizzato per identificare un Button con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	<pre>sknBtn.auxData("myBtn", "W1") ... print ("\\r\\nmy auxString=" .. sknBtn.auxStr("myBtn"))</pre>	

sknBtn.bringTop ()

Porta la finestra in primo piano.

sknBtn.bringTop ("btnName")

	Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Button attive.
param	btnName Nome del Button
return	1=successo, 0=errore
call	sknBtn.bringTop("myBtn")

sknBtn.bringBot ()

Porta la finestra in sfondo.

sknBtn.bringBot ("btnName")

	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.
param	btnName Nome del Button
return	1=successo, 0=errore
call	sknBtn.bringBot("myBtn")

sknBtn.autoHide ()

Controlla la funzionalità di auto hide di un Button.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknBtn.autoHide ("btnName", autoHide)		
	Controlla la funzionalità di auto hide di un Button. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	btnName	Nome del Button
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknBtn.autoHide("myBtn", true) ... print ("\r\nmy autohide=" .. sknBtn.autoHide ("myBtn"))	

sknBtn.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di un Button.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknBtn.autoHideTO ("btnName", autoHideTimeout)		
	Controlla il timeout per la funzionalità di auto hide di un Button. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	btnName	Nome del Button
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknBtn.autoHideTO("myBtn", 60) ... print ("\r\nmy autohideTO=" .. sknBtn.autoHideTO ("myBtn"))	

sknBtn.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di un Button.

sknBtn.autoHideMode ("btnName", "mode")		
Controlla il tipo ti hide per la funzionalità di auto hide di un Button. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.		
param	btnName	Nome del Button
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknBtn.autoHideMode("myBtn", "left") ... print ("\r\nmy autohideMode=" .. sknBtn.autoHideMode ("myBtn"))	

sknBtn.blinkEn ()

Controlla la funzionalità di blink di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknBtn.blinkEn ("btnName", blink)		
Controlla la funzionalità di blink di un Button. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknBtn.blinkEn("myBtn", true) ... print ("\r\nmy blinkEn=" .. sknBtn.blinkEn ("myBtn"))	

sknBtn.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknBtn.blinkCustEn ("btnName", blink)		
Controlla la funzionalità di blink custom o globale di un Button. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknBtn.blinkCustEn("myBtn", true) ... print ("\r\nmy blinkCustEn=" .. sknBtn.blinkCustEn ("myBtn"))	

sknBtn.autoHideCBshow ()

Controlla la funzione di callback show di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknBtn.autoHideCBshow ("btnName", "callback")		
Controlla la funzione di callback show di un Button. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknBtn.autoHideCBshow("myBtn", "myCBshow") ... print ("\r\nmy autohideCBshow=" .. sknBtn.autoHideCBshow ("myBtn"))	

sknBtn.autoHideCBhide ()

Controlla la funzione di callback hide di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknBtn.autoHideCBhide ("btnName", "callback")

	Controlla la funzione di callback hide di un Button. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknBtn.autoHideCBhide("myBtn", "myCBshow") ... print ("\\nmy autohideCBhide=" .. sknBtn.autoHideCBhide ("myBtn"))</pre>	

sknBtn.aniEn ()

Controlla la funzionalità di animation di un Button.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknBtn.aniEn ("btnName", animation)

	Controlla la funzionalità di animation di un Button. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknBtn.aniEn("myBtn", true) ... print ("\\nmy aniEn=" .. sknBtn.aniEn ("myBtn"))</pre>	

sknBtn.anMode ()

Controlla la modalità di animation di un Button.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknBtn.anMode ("btnName", "aniMode")		
Controlla la modalità di animation di un Button. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	sknBtn.anMode("myBtn", "shift") ... print ("\r\nmy aniMode=" .. sknBtn.anMode ("myBtn"))	

sknBtn.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknBtn.autoFocusChild ("btnName", autoFocus)		
Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknBtn.autoFocusChild("myBtn", true) ... print ("\r\nmy autoFocusChild=" .. sknBtn.autoFocusChild ("myBtn"))	

sknBtn.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknBtn.bringToTopChild ("btnName", en)		
Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknBtn.bringToTopChild("myBtn", true) ... print ("\r\nmy bringToTopChild=" .. sknBtn.bringToTopChild ("myBtn"))	

sknBtn.group ()

Controlla l'appartenenza di un gruppo di un Button.

sknBtn.group ("btnName", "grpName")		
Controlla l'appartenenza di un gruppo di un Button. Il grpName viene utilizzato per associare un Button con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknBtn.group("myBtn", "setGrp") ... print ("\r\nmy group=" .. sknBtn.group("myBtn"))	

sknBtn.subGroup ()

Controlla l'appartenenza di un sotto gruppo di un Button.

sknBtn.subGroup ("btnName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di un Button. Il subgrpName viene utilizzato per associare un Button con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknBtn.subGroup("myBtn", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknBtn.subGroup ("myBtn"))	

sknBtn.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknBtn.longTouchCB ("btnName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknBtn.longTouchCB("myBtn", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknBtn.longTouchCB ("myBtn"))	

sknBtn.delete ()

Distrugge un Button.

Questo metodo non rimuove gli oggetti figli della finestra.

sknBtn.delete ("btnName")		
	Distrugge un Button.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknBtn.delete("myBtn")	

sknBtn.paintCB ()

Controlla il nome della funzione di callback per il paint del button.

sknBtn.paintCB ("btnName", "callback")		
	Controlla il nome della funzione di callback per il paint del button. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del button
	Callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknBtn.paintCB("home", "RedrawCb") ... print ("\r\nmy paintCB=" .. sknBtn.paintCB ("home"))	

I BUTTON

La Classe **sknIbtn** implementa i Button con immagine, una versione semplificata dei sknBtn.

Membro	Descrizione
Control	
sknIbtn.create()	Crea un iButton
sknIbtn.enable()	Abilita un iButton
sknIbtn.delete()	Distrugge un iButton
View	
sknIbtn.show()	Visualizza un iButton
sknIbtn.hide()	Nasconde un iButton
Proprietà	
sknIbtn.size()	Modifica le dimensioni di un iButton
sknIbtn.txtcolors()	Modifica i colori del testo di un iButton
sknIbtn.txtBGcolors()	Modifica i colori del sfondo del testo di un iButton
sknIbtn.text()	Modifica il testo di un iButton
sknIbtn.transp()	Controlla la trasparenza di un iButton
sknIbtn.font()	Configura il font di un iButton
sknIbtn.txtalign()	Configura l'allineamento di un iButton
sknIbtn.pos()	Modifica la posizione di un iButton
sknIbtn.txtOffset()	Configura loffset di partenza del testo di un iButton
sknIbtn.imgunpress()	Configura l'immagine di un iButton non premuto
sknIbtn.imgpress()	Configura l'immagine di un iButton premuto
sknIbtn.imgdisabled()	Configura l'immagine di un iButton disabilitato
sknIbtn.imgPos()	Configura la posizione dell'immagine di un iButton
sknIbtn.focussable()	Configura la funzionalità focussable di un iButton
Permessi	
sknIbtn.allowMoveX()	Controlla l'abilitazione al movimento su asse X di un iButton
sknIbtn.allowMoveY()	Controlla l'abilitazione al movimento su asse Y di un iButton
sknIbtn.allowResize()	Controlla l'abilitazione alla modifica della dimensione di un iButton
Visibilità	
sknIbtn.bringTop()	Porta il iButton in primo piano
sknIbtn.bringBot()	Porta il iButton in sfondo
Aggregazione	

sknIbtn.auxData()	Controlla l'AuxData di un iButton
sknIbtn.auxStr()	Controlla l'AuxString di un iButton
sknIbtn.group()	Controlla l'appartenenza di un gruppo di un iButton
sknIbtn.subGroup()	Controlla l'appartenenza di un sotto gruppo di un iButton
Eventi	
sknIbtn.keyPressCB()	Controlla il nome della funzione di callback di key press di un iButton
sknIbtn.keyReleaseCB()	Controlla il nome della funzione di callback di key release di un iButton
Auto Hide	
sknIbtn.autoHide()	Controlla la funzionalità di auto hide di un iButton
sknIbtn.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di un iButton
sknIbtn.autoHideMode()	Controlla il tipo ti hide per la funzionalità di auto hide di un iButton
Blink	
sknIbtn.blinkEn()	Controlla la funzionalità di blink di un iButton
sknIbtn.blinkCustEn()	Controlla la funzionalità di blink custom o globale di un iButton
Auto Hide callbacks	
sknIbtn.autoHideCBshow()	Controlla la funzione di callback hide di un iButton
sknIbtn.autoHideCBhide()	Controlla la funzione di callback show di un iButton
Animazione	
sknIbtn.aniEn()	Controlla la funzionalità di animation di un iButton
sknIbtn.aniMode()	Controlla la modalità di animation di un iButton
Child	
sknIbtn.autoFocusChild()	Controlla la funzionalità di autofocus per i child iButton
sknIbtn.bringToTopChild()	Controlla la funzionalità di brig to top per i child iButton
Callbacks	
sknIbtn.longTouchCB()	Controlla il nome della funzione di callback per un long touch
sknIbtn.paintCB()	Controlla il nome della funzione di callback per il paint dell'oggetto
Sound	
sknIbtn.sound()	Configura il suono per l'evento press

Classe [sknIbtn]

sknIbtn.create()

Crea un iButton.

sknIbtn.create("btnName", "motherWin", xpos, ypos, xsize, ysize)		
	Crea un iButton.	
param	btnName	Nome del Button
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza del button
	ysize	altezza del button
return	1=successo, 0=errore	
call	sknlbtn.create("btnInfo", "winHome", 30, 50, 42, 42)	

sknIbtn.show ()

Visualizza un iButton.

sknIbtn.show ("btnName")		
	Visualizza un iButton.	
param	btnName	nome del Button
return	1=successo, 0=errore	
call	sknlbtn.show("myBtn")	

sknIbtn.hide ()

Nasconde un iButton.

sknIbtn.hide ("btnName")		
	Nasconde un iButton.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknlbtn.hide("myBtn")	

sknIbtn.size ()

Modifica o ritorna le dimensioni di un iButton.

sknIbtn.size ("btnName", x, y)		
Modifica o ritorna le dimensioni di un iButton. Se si omette x e y la funzione riporta quelli correnti.		
param	btnName	Nome del Button
	x	larghezza (opt)
	Y	altezza (opt)
return	Size x, y o 1=successo, 0=errore per Set	
call	sknlbtn.size("myBtn", 200, 100) ... x,y = sknlbtn.size("myBtn")	

sknIbtn.pos ()

Modifica o ritorna la posizione di un iButton.

sknIbtn.pos ("btnName", x, y)		
Modifica o ritorna la posizione di un iButton. Se si omettono x e y la funzione riporta la posizione corrente.		
param	btnName	Nome del Button
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknlbtn.pos("myBtn", 10, 50) ... print ("\r\nmy pos=" .. sknlbtn.pos("myBtn"))	

sknIbtn.enable ()

Controlla l'abilitazione di un iButton.

sknIbtn.enable ("btnName", enable)		
	Controlla l'abilitazione di un iButton. Se non viene valorizzato il campo enable, la funzione ritorna il valore corrente di abilitazione del Button selezionata.	
param	btnName	Nome del Button
	enable	true=Button abilitata, false=Button disabilitata (opt)
return	Enable o 1=successo, 0=errore per Set	
call	sknlbtn.enable("myBtn", true) ... print ("\r\nmy enabled=" .. sknlbtn.enable("myBtn"))	

sknIbtn.focussable()

Configura la funzionalità focussable di un iButton.

sknIbtn.focussable("btnName", focussable)		
	Configura la funzionalità focussable di un iButton. Questa funzionalità è utile quando si vuole scorrere automaticamente fra vari oggetti senza un touch esplicito.	
param	btnName	Nome del Button
	focussable	true=focussable, false=non focussable
return	1=successo, 0=errore	
call	sknlbtn.focussable("btnInfo", false)	

sknIbtn.imgunpress()

Configura l'immagine di un iButton non premuto.

sknIbtn.imgunpress("btnName", "imageIndex", "collection", "image")

	Configura l'immagine di un iButton non premuto. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button	
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)	
	collection	nome della collezione di immagini	
	image	nome dell'immagine	
return	1=successo, 0=errore		
call	sknlbtn.imgunpress("btnInfo", 0, "icons", "alert")		

sknIbtn.imgpress()

Configura l'immagine di un iButton premuto.

sknIbtn.imgpress("btnName", "imageIndex", "collection", "image")

	Configura l'immagine di un iButton premuto. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button	
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)	
	collection	nome della collezione di immagini	
	image	nome dell'immagine	
return	1=successo, 0=errore		
call	sknlbtn.imgpress("btnInfo", 0, "icons", "alert")		

sknIbtn.imgdisabled()

Configura l'immagine di un iButton disabilitato.

sknIbtn.imgdisabled("btnName", imageIndex, "collection", "image")

	Configura l'immagine di un iButton disabilitato. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button	
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)	
	collection	nome della collezione di immagini	
	image	nome dell'immagine	
return	1=successo, 0=errore		
call	sknlbtn.imgdisabled("btnInfo", 0, "icons", "alert")		

sknIbtn.imgPos()

Configura la posizione dell'immagine di un iButton.

sknIbtn.imgPos("btnName", imgIndex, xpos, ypos)

	Configura la posizione dell'immagine di un iButton. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button	
	imgIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)	
	xpos	posizione x all'interno dell'area del button	
	ypos	posizione y all'interno dell'area del button	
return	1=successo, 0=errore		
call	sknlbtn.imgPos("btnInfo", 0, 10, 34)		

sknIbtn.txtcolors()

Modifica i colori del testo di un iButton.

sknIbtn.txtcolors("btnName", colorUnpress, colorPress)		
	Modifica i colori del testo di un iButton.	
param	btnName	Nome del Button
	colorUnpress	colore del testo del button non premuto (numerico o stringa per colore predefinito)
	colorPress	colore del testo del button premuto (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknlbtn.txtcolors("btnHlp", "black", "white")	

sknIbtn.txtBGcolors()

Modifica i colori del di sfondo del testo di un iButton.

sknIbtn.txtBGcolors("btnName", colorUnpress, colorPress)		
	Modifica i colori del di sfondo del testo di un iButton.	
param	btnName	Nome del Button
	colorUnpress	colore di sfondo del testo del button non premuto (numerico o stringa per colore predefinito)
	colorPress	colore di sfondo del testo del button premuto (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknlbtn.txtBGcolors("btnHlp", "blue", "magenta")	

sknIbtn.txtalign()

Configura l'allineamento di un iButton.

sknIbtn.txtalign("btnName", "align")		
	Configura l'allineamento di un iButton.	
param	btnName	Nome del Button
	align	allineamento {"Hor_Vert", "Hor_Top", "Hor_Bottom", "Hor_Baseline", "Hor_Vcenter", "Left_Vert", "Left_Top", "Left_Bottom", "Left_Baseline", "Left_Vcenter", "Right_Vert", "Right_Top", "Right_Bottom", "Right_Baseline", "Right_Vcenter", "Center_Vert", "Center_Top", "Center_Bottom", "Center_Baseline", "Center_Vcenter", "Hcenter_Vert", "Hcenter_Top", "Hcenter_Bottom", "Hcenter_Baseline", "Hcenter_Vcenter"}
return	1=successo, 0=errore	
call	sknlbtn.txtalign("btnHlp", "HorLeft_Vert_Top")	

sknIbtn.txtOffset()

Configura l'offset di partenza del testo di un iButton.

sknIbtn.txtOffset("objName", x, y)		
	Configura l'offset di partenza del testo di un iButton.	
param	btnName	Nome del Button
	x	offset asse x
	y	offset asse y
return	1=successo, 0=errore	
call	sknlbtn.txtOffset("hmlBtn", 10, 50)	

sknIbtn.text()

Modifica il testo di un iButton.

sknIbtn.text("btnName", "text")		
		Modifica il testo di un iButton.
param	btnName	Nome del Button
	text	testo del button
return	1=successo, 0=errore	
call	sknlbtn.text("btnHIp", "HELP")	

sknIbtn.font()

Configura il font di un iButton.

sknIbtn.font("btnName", "fontName")		
		Configura il font di un iButton.
param	btnName	Nome del Button
	fontName	nome font
return	1=successo, 0=errore	
call	sknlbtn.font("btnHIp", "Arial14")	

sknIbtn.keyPressCB()

Controlla il nome della funzione di callback per il key press del Button.

sknIbtn.keyPressCB ("btnName", "callback")		
		Controlla il nome della funzione di callback per il key press del Button. Se non valorizzato il campo callback, viene ritornato il valore corrente.
param	btnName	Nome del Button
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknlbtn.keyPressCB("myBtn", "keyPressWin") ... print ("\r\nmy winKeyPressCB=" .. sknlbtn.keyPressCB ("myBtn"))	

sknIbtn.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release del Button.

sknIbtn.keyReleaseCB ("btnName" , "callback")		
	Controlla il nome della funzione di callback per il key press del Button. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknlbtn.keyReleaseCB("myBtn", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknlbtn.keyReleaseCB ("myBtn"))	

sknIbtn.trasp ()

Controlla la trasparenza di un iButton.

sknIbtn.trasp ("btnName" , transp)		
	Controlla la trasparenza di un iButton. Se non viene valorizzato il campo transp, la funzione ritorna il valore corrente di trasparenza del Button selezionata.	
param	btnName	Nome del Button
	Transp	true=Button trasparente, false=Button opaca (opt)
return	Transparency o 1=successo, 0=errore per Set	
call	sknlbtn.transp("myBtn", true) ... print ("\r\nmy transp=" .. sknlbtn.transp("myBtn"))	

sknIbtn.sound ()

Configura il suono per l'evento press.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknIbtn.sound ("lblName", "soundName")		
	Configura il suono per l'evento press.	
param	lblName	Nome del Button
	soundName	Nome suono
return	1=successo, 0=errore	
call	sknIbtn.sound("closeBtn", "soundEnd")	

sknIbtn.delete ()

Distrugge un iButton.

Questo metodo non rimuove gli oggetti figli della finestra.

sknIbtn.delete ("btnName")		
	Distrugge un iButton.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknIbtn.delete("myBtn")	

sknIbtn.allowMoveX ()

Controlla l'abilitazione al movimento su asse X di un iButton.

sknIbtn.allowMoveX ("btnName", allow)		
	Controlla l'abilitazione al movimento su asse X di un iButton. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione del Button selezionata.	
param	btnName	Nome del Button
	allow	true=consente il movimento sull'asse x del Button, false=Button fissa su asse x (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknlbtn.allowMoveX("myBtn", true) ... print ("\r\nmy allowXmove=" .. sknlbtn.allowMoveX("myBtn"))	

sknIbtn.allowMoveY ()

Controlla l'abilitazione al movimento su asse Y di un iButton.

sknIbtn.allowMoveY ("btnName", allow)		
	Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione del Button selezionata.	
param	btnName	Nome del Button
	allow	true=consente il movimento sull'asse y del Button, false=Button fissa su asse y (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknlbtn.allowMoveY("myBtn", true) ... print ("\r\nmy allowYmove=" .. sknlbtn.allowMoveY("myBtn"))	

sknIbtn.allowResize ()

Controlla l'abilitazione alla modifica della dimensione di un iButton.

sknIbtn.allowResize ("btnName", allow)		
	Controlla l'abilitazione alla modifica della dimensione di un iButton. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di trasparenza del Button selezionata.	
param	btnName	Nome del Button
	allow	true=consente il ridimensionamento del Button, false=Button non ridimensionabile (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknlbtn.allowResize("myBtn", true) ... print ("\r\nmy allowResize=" .. sknlbtn.allowResize("myBtn"))	

sknIbtn.auxData ()

Controlla l'AuxData di un iButton.

sknIbtn.auxData ("btnName", auxData)		
	Controlla l'AuxData di un iButton. L'AuxData viene utilizzato per identificare un iButton con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.	
param	btnName	Nome del Button
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknlbtn.auxData("myBtn", 1234) ... print ("\r\nmy auxData=" .. sknlbtn.auxData("myBtn"))	

sknIbtn.auxStr ()

Controlla l'AuxString di un iButton.

sknIbtn.auxStr ("btnName", auxString)		
Controlla l'AuxString di un iButton. L'AuxString viene utilizzato per identificare un iButton con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknlbtn.auxData("myBtn", "W1") ... print ("\r\nmy auxString=" .. sknlbtn.auxStr("myBtn"))	

sknIbtn.bringTop ()

Porta la finestra in primo piano.

sknIbtn.bringTop ("btnName")		
Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Button attive.		
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknlbtn.bringTop("myBtn")	

sknIbtn.bringBot ()

Porta la finestra in sfondo.

sknIbtn.bringBot ("btnName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknlbtn.bringBot("myBtn")	

sknIbtn.autoHide ()

Controlla la funzionalità di auto hide di un iButton.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknIbtn.autoHide ("btnName", autoHide)		
	Controlla la funzionalità di auto hide di un iButton. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	btnName	Nome del Button
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknlbtn.autoHide("myBtn", true) ... print ("\r\nmy autohide=" .. sknlbtn.autoHide ("myBtn"))	

sknIbtn.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di un iButton.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknIbtn.autoHideTO ("btnName", autoHideTimeout)

	Controlla il timeout per la funzionalità di auto hide di un iButton. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	btnName	Nome del Button
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknlbtn.autoHideTO("myBtn", 60) ... print ("\r\nmy autohideTO=" .. sknlbtn.autoHideTO ("myBtn"))	

sknIbtn.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di un iButton.

sknIbtn.autoHideMode ("btnName", "mode")

	Controlla il tipo ti hide per la funzionalità di auto hide di un iButton. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.	
param	btnName	Nome del Button
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknlbtn.autoHideMode("myBtn", "left") ... print ("\r\nmy autohideMode=" .. sknlbtn.autoHideMode ("myBtn"))	

sknIbtn.blinkEn ()

Controlla la funzionalità di blink di un iButton.

Funzionalità condizionata dalla profilazione `SKN_BLINK_OBJ_EN == 1`

sknIbtn.blinkEn ("btnName", blink)		
Controlla la funzionalità di blink di un iButton. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknlbtn.blinkEn("myBtn", true) ... print ("\r\nmy blinkEn=" .. sknlbtn.blinkEn ("myBtn"))	

sknIbtn.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di un iButton.

Funzionalità condizionata dalla profilazione `SKN_BLINK_OBJ_EN == 1`

sknIbtn.blinkCustEn ("btnName", blink)		
Controlla la funzionalità di blink custom o globale di un iButton. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknlbtn.blinkCustEn("myBtn", true) ... print ("\r\nmy blinkCustEn=" .. sknlbtn.blinkCustEn ("myBtn"))	

sknIbtn.autoHideCBshow ()

Controlla la funzione di callback show di un iButton.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknIbtn.autoHideCBshow ("btnName", "callback")

Controlla la funzione di callback show di un iButton. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknlbtn.autoHideCBshow("myBtn", "myCBshow") ... print ("\\r\\nmy autohideCBshow=" .. sknlbtn.autoHideCBshow ("myBtn"))</pre>	

sknIbtn.autoHideCBhide ()

Controlla la funzione di callback hide di un iButton.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknIbtn.autoHideCBhide ("btnName", "callback")

Controlla la funzione di callback hide di un iButton. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknlbtn.autoHideCBhide("myBtn", "myCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknlbtn.autoHideCBhide ("myBtn"))</pre>	

sknIbtn.aniEn ()

Controlla la funzionalità di animation di un iButton.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknIbtn.aniEn ("btnName", animation)		
Controlla la funzionalità di animation di un iButton. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknlbtn.aniEn("myBtn", true) ... print ("\r\nmy aniEn=" .. sknlbtn.aniEn ("myBtn"))</pre>	

sknIbtn.animode ()

Controlla la modalità di animation di un iButton.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknIbtn.animode ("btnName", "aniMode")		
Controlla la modalità di animation di un iButton. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	<pre>sknlbtn.animode("myBtn", "shift") ... print ("\r\nmy aniMode=" .. sknlbtn.animode ("myBtn"))</pre>	

sknIbtn.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknIbtn.autoFocusChild ("btnName", autoFocus)		
	Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknlbtn.autoFocusChild("myBtn", true) ... print ("\r\nmy autoFocusChild=" .. sknlbtn.autoFocusChild ("myBtn"))	

sknIbtn.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknIbtn.bringToTopChild ("btnName", en)		
	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknlbtn.bringToTopChild("myBtn", true) ... print ("\r\nmy bringToTopChild=" .. sknlbtn.bringToTopChild ("myBtn"))	

sknIbtn.group ()

Controlla l'appartenenza di un gruppo di un iButton.

sknIbtn.group ("btnName", "grpName")		
Controlla l'appartenenza di un gruppo di un iButton. Il grpName viene utilizzato per associare un iButton con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknlbtn.group("myBtn", "setGrp") ... print ("\r\nmy group=" .. sknlbtn.group("myBtn"))	

sknIbtn.subGroup ()

Controlla l'appartenenza di un sotto gruppo di un iButton.

sknIbtn.subGroup ("btnName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di un iButton. Il subgrpName viene utilizzato per associare un iButton con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknlbtn.subGroup("myBtn", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknlbtn.subGroup ("myBtn"))	

sknIbtn.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknIbtn.longTouchCB ("btnName", "callback")		
	Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknlbtn.longTouchCB("myBtn", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknlbtn.longTouchCB ("myBtn"))	

sknIbtn.paintCB ()

Controlla il nome della funzione di callback per il paint del button.

sknIbtn.paintCB ("btnName", "callback")		
	Controlla il nome della funzione di callback per il paint della label. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del button
	Callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknlbtn.paintCB("home", "RedrawCb") ... print ("\r\nmy paintCB=" .. sknlbtn.paintCB ("home"))	

ITBUTTON

La Classe **sknITbtn** implementa Text and Image Buttons..

Membro	Descrizione
Control	
sknITbtn.create()	Crea un ITbutton
sknITbtn.enable()	Abilita un ITbutton
sknITbtn.delete()	Distrugge un ITbutton
View	
sknITbtn.show()	Visualizza un ITbutton
sknITbtn.hide()	Nasconde un ITbutton
Proprietà	
sknITbtn.size()	Modifica le dimensioni di un ITbutton
sknITbtn.txtcolors()	Modifica i colori del testo di un ITbutton
sknITbtn.text()	Modifica il testo di un ITbutton
sknITbtn.transp()	Controlla la trasparenza di un ITbutton
sknITbtn.font()	Configura il font di un ITbutton
sknITbtn.txtalign()	Configura l'allineamento di un ITbutton
sknITbtn.pos()	Modifica la posizione di un ITbutton
sknITbtn.imgunpress()	Configura l'immagine di un ITbutton non premuto
sknITbtn.imgpress()	Configura l'immagine di un ITbutton premuto
sknITbtn.imgdisabled()	Configura l'immagine di un ITbutton disabilitato
sknITbtn.imgPos()	Configura la posizione dell'immagine di un ITbutton
sknITbtn.focussable()	Configura la funzionalità focussable di un ITbutton
Permessi	
sknITbtn.allowMoveX()	Controlla l'abilitazione al movimento su asse X di un ITbutton
sknITbtn.allowMoveY()	Controlla l'abilitazione al movimento su asse Y di un ITbutton
sknITbtn.allowResize()	Controlla l'abilitazione alla modifica della dimensione di un ITbutton
Visibilità	
sknITbtn.bringTop()	Porta il ITbutton in primo piano
sknITbtn.bringBot()	Porta il ITbutton in sfondo
Aggregazione	
sknITbtn.auxData()	Controlla l'AuxData di un ITbutton
sknITbtn.auxStr()	Controlla l'AuxString di un ITbutton
sknITbtn.group()	Controlla l'appartenenza di un gruppo di un ITbutton

<code>sknITbtn.subGroup()</code>	Controlla l'appartenenza di un sotto gruppo di un ITbutton
Eventi	
<code>sknITbtn.keyPressCB()</code>	Controlla il nome della funzione di callback di key press di un ITbutton
<code>sknITbtn.keyReleaseCB()</code>	Controlla il nome della funzione di callback di key release di un ITbutton
Auto Hide	
<code>sknITbtn.autoHide()</code>	Controlla la funzionalità di auto hide di un ITbutton
<code>sknITbtn.autoHideTO()</code>	Controlla il timeout per la funzionalità di auto hide di un ITbutton
<code>sknITbtn.autoHideMode()</code>	Controlla il tipo ti hide per la funzionalità di auto hide di un ITbutton
Blink	
<code>sknITbtn.blinkEn()</code>	Controlla la funzionalità di blink di un ITbutton
<code>sknITbtn.blinkCustEn()</code>	Controlla la funzionalità di blink custom o globale di un ITbutton
Auto Hide callbacks	
<code>sknITbtn.autoHideCBshow()</code>	Controlla la funzione di callback hide di un ITbutton
<code>sknITbtn.autoHideCBhide()</code>	Controlla la funzione di callback show di un ITbutton
Animazione	
<code>sknITbtn.aniEn()</code>	Controlla la funzionalità di animation di un ITbutton
<code>sknITbtn.aniMode()</code>	Controlla la modalità di animation di un ITbutton
Child	
<code>sknITbtn.autoFocusChild()</code>	Controlla la funzionalità di autofocus per i child ITbutton
<code>sknITbtn.bringToTopChild()</code>	Controlla la funzionalità di brig to top per i child ITbutton
Callbacks	
<code>sknITbtn.longTouchCB()</code>	Controlla il nome della funzione di callback per un long touch
<code>sknITtn.paintCB()</code>	Controlla il nome della funzione di callback per il paint dell'oggetto
Sound	
<code>sknITbtn.sound()</code>	Configura il suono per l'evento press

Classe [sknITbtn]

sknITbtn.create()

Crea un Button.

sknITbtn.create("btnName", "motherWin", xpos, ypos, xsize, ysize)		
	Crea un Button.	
param	btnName	Nome del Button
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza del button
	ysize	altezza del button
return	1=successo, 0=errore	
call	sknITbtn.create("btnInfo", "winHome", 30, 50, 42, 42)	

sknITbtn.show ()

Visualizza un Button.

sknITbtn.show ("btnName")		
	Visualizza un Button.	
param	btnName	nome del Button
return	1=successo, 0=errore	
call	sknITbtn.show("myBtn")	

sknITbtn.hide ()

Nasconde un Button.

sknITbtn.hide ("btnName")		
	Nasconde un Button.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknITbtn.hide("myBtn")	

sknITbtn.size ()

Modifica o ritorna le dimensioni di un Button.

sknITbtn.size ("btnName", x, y)		
Modifica o ritorna le dimensioni di un Button. Se si omette x e y la funzione riporta quelli correnti.		
param	btnName	Nome del Button
	x	larghezza (opt)
	Y	altezza (opt)
return	Size x, y o 1=successo, 0=errore per Set	
call	sknITbtn.size("myBtn", 200, 100) ... x,y = sknITbtn.size("myBtn")	

sknITbtn.pos ()

Modifica o ritorna la posizione di un Button.

sknITbtn.pos ("btnName", x, y)		
Modifica o ritorna la posizione di un Button. Se si omettono x e y la funzione riporta la posizione corrente.		
param	btnName	Nome del Button
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknITbtn.pos("myBtn", 10, 50) ... print ("\r\nmy pos=" .. sknITbtn.pos("myBtn"))	

sknITbtn.enable()

Controlla l'abilitazione di un Button.

sknITbtn.enable ("btnName", enable)		
	Controlla l'abilitazione di un Button. Se non viene valorizzato il campo enable, la funzione ritorna il valore corrente di abilitazione del Button selezionata.	
param	btnName	Nome del Button
	enable	true=Button abilitata, false=Button disabilitata (opt)
return	Enable o 1=successo, 0=errore per Set	
call	sknITbtn.enable("myBtn", true) ... print ("\r\nmy enabled=" .. sknITbtn.enable("myBtn"))	

sknITbtn.focussable()

Configura la funzionalità focussable di un Button.

sknITbtn.focussable("btnName", focussable)		
	Configura la funzionalità focussable di un Button. Questa funzionalità è utile quando si vuole scorrere automaticamente fra vari oggetti senza un touch esplicito.	
param	btnName	Nome del Button
	focussable	true=focussable, false=non focussable
return	1=successo, 0=errore	
call	sknITbtn.focussable("btnInfo", false)	

sknITbtn.imgunpress()

Configura l'immagine di un Button non premuto.

sknITbtn.imgunpress("btnName", "imageIndex", "collection", "image")		
Configura l'immagine di un Button non premuto. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknITbtn.imgunpress("btnInfo", 0, "icons", "alert")	

sknITbtn.imgpress()

Configura l'immagine di un Button premuto.

sknITbtn.imgpress("btnName", "imageIndex", "collection", "image")		
Configura l'immagine di un Button premuto. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.		
param	btnName	Nome del Button
	imageIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknITbtn.imgpress("btnInfo", 0, "icons", "alert")	

sknITbtn.imgdisabled()

Configura l'immagine di un Button disabilitato.

sknITbtn.imgdisabled("btnName", imgIndex, "collection", "image")

	Configura l'immagine di un Button disabilitato. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.	
param	btnName	Nome del Button
	imgIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknITbtn.imgdisabled("btnInfo", 0, "icons", "alert")	

sknITbtn.imgPos()

Configura la posizione dell'immagine di un Button.

sknITbtn.imgPos("btnName", imgIndex, xpos, ypos)

	Configura la posizione dell'immagine di un Button. Si possono visualizzare differenti immagini all'interno dello stesso Button, tramite l'indice si seleziona l'immagine relativa. L'indice 0 rappresenta l'immagine in sfondo e l'indice massimo identifica quella in primo piano. Al fine di poter comporre le immagini più agevolmente è possibile configurare le coordinate di partenza delle singole immagini (il default è nel corner in alto a sinistra del bottone). Le coordinate si riferiscono all'interno del bottone, l'angolo in alto a sinistra corrisponde a 0,0.	
param	btnName	Nome del Button
	imgIndex	indice dell'immagine (compreso fra 0 e il massimo supportato dal Bsp: GUI_MAX_BUTTON_IMAGES-1)
	xpos	posizione x all'interno dell'area del button
	ypos	posizione y all'interno dell'area del button
return	1=successo, 0=errore	
call	sknITbtn.imgPos("btnInfo", 0, 10, 34)	

sknITbtn.keyPressCB ()

Controlla il nome della funzione di callback per il key press del Button.

sknITbtn.keyPressCB ("btnName" , "callback")		
	Controlla il nome della funzione di callback per il key press del Button. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknITbtn.keyPressCB("myBtn", "keyPressWin") ... print ("\r\nmy winKeyPressCB=" .. sknITbtn.keyPressCB ("myBtn"))	

sknITbtn.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release del Button.

sknITbtn.keyReleaseCB ("btnName" , "callback")		
	Controlla il nome della funzione di callback per il key release del Button. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknITbtn.keyReleaseCB("myBtn", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknITbtn.keyReleaseCB ("myBtn"))	

sknITbtn.trasp ()

Controlla la trasparenza di un Button.

sknITbtn.trasp ("btnName", transp)		
	Controlla la trasparenza di un Button. Se non viene valorizzato il campo transp, la funzione ritorna il valore corrente di trasparenza del Button selezionata.	
param	btnName	Nome del Button
	Transp	true=Button trasparente, false=Button opaca (opt)
return	Tranparency o 1=successo, 0=errore per Set	
call	sknITbtn.transp("myBtn", true) ... print ("\r\nmy transp=" .. sknITbtn.transp("myBtn"))	

sknITbtn.sound ()

Configura il suono per l'evento press.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknITbtn.sound ("lblName", "soundName")		
	Configura il suono per l'evento press.	
param	lblName	Nome del Button
	soundName	Nome suono
return	1=successo, 0=errore	
call	sknITbtn.sound("closeBtn", "soundEnd")	

sknITbtn.txtcolors()

Modifica i colori del testo di un Button.

sknITbtn.txtcolors("btnName", colorUnpress, colorPress)		
	Modifica i colori del testo di un Button.	
param	btnName	Nome del Button
	colorUnpress	colore del testo del button non premuto (numerico o stringa per colore predefinito)
	colorPress	colore del testo del button premuto (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknITbtn.txtcolors("btnHlp", "black", "white")	

sknITbtn.text()

Modifica il testo di un Button.

sknITbtn.text("btnName", "text")		
	Modifica il testo di un Button.	
param	btnName	Nome del Button
	text	testo del button
return	1=successo, 0=errore	
call	sknITbtn.text("btnHlp", "HELP")	

sknITbtn.font()

Configura il font di un Button.

sknITbtn.font("btnName", "fontName")		
	Configura il font di un Button.	
param	btnName	Nome del Button
	fontName	nome font
return	1=successo, 0=errore	
call	sknITbtn.font("btnHlp", "Arial14")	

sknITbtn.txtalign()

Configura l'allineamento di un Button.

sknITbtn.txtalign("btnName", "align")		
	Configura l'allineamento di un Button.	
param	btnName	Nome del Button
	align	allineamento {"Hor_Vert", "Hor_Top", "Hor_Bottom", "Hor_Baseline", "Hor_Vcenter", "Left_Vert", "Left_Top", "Left_Bottom", "Left_Baseline", "Left_Vcenter", "Right_Vert", "Right_Top", "Right_Bottom", "Right_Baseline", "Right_Vcenter", "Center_Vert", "Center_Top", "Center_Bottom", "Center_Baseline", "Center_Vcenter", "Hcenter_Vert", "Hcenter_Top", "Hcenter_Bottom", "Hcenter_Baseline", "Hcenter_Vcenter"}
return	1=successo, 0=errore	
call	sknITbtn.txtalign("btnName", "align")	

sknITbtn.delete()

Distrugge un Button.

Questo metodo non rimuove gli oggetti figli della finestra.

sknITbtn.delete ("btnName")		
	Distrugge un Button.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknITbtn.delete("btnName")	

sknITbtn.allowMoveX ()

Controlla l'abilitazione al movimento su asse X di un Button.

sknITbtn.allowMoveX ("btnName", allow)		
	Controlla l'abilitazione al movimento su asse X di un Button. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione del Button selezionata.	
param	btnName	Nome del Button
	allow	true=consente il movimento sull'asse x del Button, false=Button fissa su asse x (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknITbtn.allowMoveX("myBtn", true) ... print ("\r\nmy allowXmove=" .. sknITbtn.allowMoveX("myBtn"))	

sknITbtn.allowMoveY ()

Controlla l'abilitazione al movimento su asse Y di un Button.

sknITbtn.allowMoveY ("btnName", allow)		
	Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione del Button selezionata.	
param	btnName	Nome del Button
	allow	true=consente il movimento sull'asse y del Button, false=Button fissa su asse y (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknITbtn.allowMoveY("myBtn", true) ... print ("\r\nmy allowYmove=" .. sknITbtn.allowMoveY("myBtn"))	

sknITbtn.allowResize ()

Controlla l'abilitazione alla modifica della dimensione di un Button.

sknITbtn.allowResize ("btnName", allow)		
	Controlla l'abilitazione alla modifica della dimensione di un Button. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di trasparenza del Button selezionata.	
param	btnName	Nome del Button
	allow	true=consente il ridimensionamento del Button, false=Button non ridimensionabile (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknITbtn.allowResize("myBtn", true) ... print ("\r\nmy allowResize=" .. sknITbtn.allowResize("myBtn"))	

sknITbtn.auxData ()

Controlla l'AuxData di un Button.

sknITbtn.auxData ("btnName", auxData)		
	Controlla l'AuxData di un Button. L'AuxData viene utilizzato per identificare un Button con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.	
param	btnName	Nome del Button
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknITbtn.auxData("myBtn", 1234) ... print ("\r\nmy auxData=" .. sknITbtn.auxData("myBtn"))	

sknITbtn.auxStr ()

Controlla l'AuxString di un Button.

sknITbtn.auxStr ("btnName", auxString)		
Controlla l'AuxString di un Button. L'AuxString viene utilizzato per identificare un Button con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknITbtn.auxData("myBtn", "W1") ... print ("\r\nmy auxString=" .. sknITbtn.auxStr("myBtn"))	

sknITbtn.bringTop ()

Porta la finestra in primo piano.

sknITbtn.bringTop ("btnName")		
Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Button attive.		
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknITbtn.bringTop("myBtn")	

sknITbtn.bringBot ()

Porta la finestra in sfondo.

sknITbtn.bringBot ("btnName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	btnName	Nome del Button
return	1=successo, 0=errore	
call	sknITbtn.bringBot("myBtn")	

sknITbtn.autoHide ()

Controlla la funzionalità di auto hide di un Button.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknITbtn.autoHide ("btnName", autoHide)		
	Controlla la funzionalità di auto hide di un Button. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	btnName	Nome del Button
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknITbtn.autoHide("myBtn", true) ... print ("\r\nmy autohide=" .. sknITbtn.autoHide ("myBtn"))	

sknITbtn.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di un Button.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknITbtn.autoHideTO ("btnName", autoHideTimeout)		
Controlla il timeout per la funzionalità di auto hide di un Button. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.		
param	btnName	Nome del Button
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknITbtn.autoHideTO("myBtn", 60) ... print ("\r\nmy autohideTO=" .. sknITbtn.autoHideTO ("myBtn"))	

sknITbtn.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di un Button.

sknITbtn.autoHideMode ("btnName", "mode")		
Controlla il tipo ti hide per la funzionalità di auto hide di un Button. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.		
param	btnName	Nome del Button
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknITbtn.autoHideMode("myBtn", "left") ... print ("\r\nmy autohideMode=" .. sknITbtn.autoHideMode ("myBtn"))	

sknITbtn.blinkEn ()

Controlla la funzionalità di blink di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknITbtn.blinkEn ("btnName", blink)		
Controlla la funzionalità di blink di un Button. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknITbtn.blinkEn("myBtn", true) ... print ("\r\nmy blinkEn=" .. sknITbtn.blinkEn ("myBtn"))	

sknITbtn.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknITbtn.blinkCustEn ("btnName", blink)		
Controlla la funzionalità di blink custom o globale di un Button. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknITbtn.blinkCustEn("myBtn", true) ... print ("\r\nmy blinkCustEn=" .. sknITbtn.blinkCustEn ("myBtn"))	

sknITbtn.autoHideCBshow ()

Controlla la funzione di callback show di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknITbtn.autoHideCBshow ("btnName", "callback")

	Controlla la funzione di callback show di un Button. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknITbtn.autoHideCBshow("myBtn", "myCBshow") ... print ("\\r\\nmy autohideCBshow=" .. sknITbtn.autoHideCBshow ("myBtn"))</pre>	

sknITbtn.autoHideCBhide ()

Controlla la funzione di callback hide di un Button.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknITbtn.autoHideCBhide ("btnName", "callback")

	Controlla la funzione di callback hide di un Button. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknITbtn.autoHideCBhide("myBtn", "myCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknITbtn.autoHideCBhide ("myBtn"))</pre>	

sknITbtn.aniEn ()

Controlla la funzionalità di animation di un Button.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknITbtn.aniEn ("btnName", animation)		
Controlla la funzionalità di animation di un Button. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	sknITbtn.aniEn("myBtn", true) ... print ("\r\nmy aniEn=" .. sknITbtn.aniEn ("myBtn"))	

sknITbtn.aniMode ()

Controlla la modalità di animation di un Button.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknITbtn.aniMode ("btnName", "aniMode")		
Controlla la modalità di animation di un Button. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	btnName	Nome del Button
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	sknITbtn.aniMode("myBtn", "shift") ... print ("\r\nmy aniMode=" .. sknITbtn.aniMode ("myBtn"))	

sknITbtn.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknITbtn.autoFocusChild ("btnName", autoFocus)		
	Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknITbtn.autoFocusChild("myBtn", true) ... print ("\r\nmy autoFocusChild=" .. sknITbtn.autoFocusChild ("myBtn"))	

sknITbtn.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknITbtn.bringToTopChild ("btnName", en)		
	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknITbtn.bringToTopChild("myBtn", true) ... print ("\r\nmy bringToTopChild=" .. sknITbtn.bringToTopChild ("myBtn"))	

sknITbtn.group ()

Controlla l'appartenenza di un gruppo di un Button.

sknITbtn.group ("btnName", "grpName")		
Controlla l'appartenenza di un gruppo di un Button. Il grpName viene utilizzato per associare un Button con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	<pre>sknITbtn.group("myBtn", "setGrp") ... print ("\\r\\nmy group=" .. sknITbtn.group("myBtn"))</pre>	

sknITbtn.subGroup ()

Controlla l'appartenenza di un sotto gruppo di un Button.

sknITbtn.subGroup ("btnName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di un Button. Il subgrpName viene utilizzato per associare un Button con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	btnName	Nome del Button
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	<pre>sknITbtn.subGroup("myBtn", "mySubGrp") ... print ("\\r\\nmy subGrp=" .. sknITbtn.subGroup ("myBtn"))</pre>	

sknITbtn.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknITbtn.longTouchCB ("btnName", "callback")		
	Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del Button
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknITbtn.longTouchCB("myBtn", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknITbtn.longTouchCB ("myBtn"))	

sknITbtn.paintCB ()

Controlla il nome della funzione di callback per il paint del button.

sknITbtn.paintCB ("btnName", "callback")		
	Controlla il nome della funzione di callback per il paint della label. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	btnName	Nome del button
	Callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknITbtn.paintCB("home", "RedrawCb") ... print ("\r\nmy paintCB=" .. sknITbtn.paintCB ("home"))	

IMAGE

La Classe **sknImg** implementa la presentazione delle immagini.

Membro	Descrizione
Control	
sknImg.create()	Crea una Image
sknImg.delete()	Distrugge una Image
View	
sknImg.show()	Visualizza una Image
sknImg.hide()	Nasconde una Image
Proprietà	
sknImg.size()	Modifica le dimensioni di una Image
sknImg.pos()	Modifica la posizione di una Image
sknImg.image()	Configura l'immagine di una Image
Permessi	
sknImg.allowMoveX()	Controlla l'abilitazione al movimento su asse X di una Image
sknImg.allowMoveY()	Controlla l'abilitazione al movimento su asse Y di una Image
sknImg.allowResize()	Controlla l'abilitazione alla modifica della dimensione di una Image
Visibilità	
sknImg.bringTop()	Porta l'Image in primo piano
sknImg.bringBot()	Porta l'Image in sfondo
Aggregazione	
sknImg.auxData()	Controlla l'AuxData di una Image
sknImg.auxStr()	Controlla l'AuxString di una Image
sknImg.group()	Controlla l'appartenenza di un gruppo di una Image
sknImg.subGroup()	Controlla l'appartenenza di un sotto gruppo di una Image
Eventi	
sknImg.keyPressCB()	Controlla il nome della funzione di callback di key press dell'Image
sknImg.keyReleaseCB()	Controlla il nome della funzione di callback di key release dell'Image
Auto Hide	
sknImg.autoHide()	Controlla la funzionalità di auto hide di una Image
sknImg.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di una Image
sknImg.autoHideMode()	Controlla il tipo ti hide per la funzionalità di auto hide di una Image

Blink	
sknImg.blinkEn()	Controlla la funzionalità di blink di una Image
sknImg.blinkCustEn()	Controlla la funzionalità di blink custom o globale di una Image
Auto Hide callbacks	
sknImg.autoHideCBshow()	Controlla la funzione di callback hide di una Image
sknImg.autoHideCBhide()	Controlla la funzione di callback show di una Image
Animazione	
sknImg.aniEn()	Controlla la funzionalità di animation di una Image
sknImg.aniMode()	Controlla la modalità di animation di una Image
Child	
sknImg.autoFocusChild()	Controlla la funzionalità di autofocus per i child Image
sknImg.bringToTopChild()	Controlla la funzionalità di brig to top per i child Image
Callbacks	
sknImg.longTouchCB()	Controlla il nome della funzione di callback per un long touch
sknImg.paintCB()	Controlla il nome della funzione di callback per il paint dell'oggetto

Classe [sknImg]

sknImg.create()

Crea una Immagine.

sknImg.create("imgName", "motherWin", xpos, ypos, xsize, ysize)		
	Crea una Immagine.	
param	imgName	Nome dell'Immagine
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza dell'Immagine
	ysize	altezza dell'Immagine
return	1=successo, 0=errore	
call	sknImg.create("myImg", "winHome", 30, 50, 42, 42)	

sknImg.show()

Visualizza una Immagine.

sknImg.show ("imgName")		
	Visualizza una Immagine.	
param	imgName	nome dell'Immagine
return	1=successo, 0=errore	
call	sknImg.show("myBtn")	

sknImg.hide()

Nasconde una Immagine.

sknImg.hide ("imgName")		
	Nasconde una Immagine.	
param	imgName	Nome dell'Immagine
return	1=successo, 0=errore	
call	sknImg.hide("myBtn")	

sknImg.size ()

Modifica o ritorna le dimensioni di una Immagine.

sknImg.size ("imgName", x, y)		
	Modifica o ritorna le dimensioni di una Immagine. Se si omette x e y la funzione riporta quelli correnti.	
param	imgName	Nome dell'Immagine
	x	larghezza (opt)
	Y	altezza (opt)
return	Size x, y o 1=successo, 0=errore per Set	
call	sknImg.size("myBtn", 200, 100) ... x,y = sknImg.size("myBtn")	

sknImg.pos ()

Modifica o ritorna la posizione di una Immagine.

sknImg.pos ("imgName", x, y)		
	Modifica o ritorna la posizione di una Immagine. Se si omettono x e y la funzione riporta la posizione corrente.	
param	imgName	Nome dell'Immagine
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknImg.pos("myBtn", 10, 50) ... print ("\r\nmy pos=" .. sknImg.pos("myBtn"))	

sknImg.image()

Configura l'immagine di una Image.

sknImg.image ("imgName", "collection", "image")		
	Configura l'immagine di una Image.	
param	imgName	Nome dell'Immagine
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknImg.image("ImgInfo", "icons", "alert")	

sknImg.delete ()

Distrugge una Immagine.

Questo metodo non rimuove gli oggetti figli della finestra.

sknImg.delete ("imgName")		
	Distrugge una Immagine.	
param	imgName	Nome dell'Immagine
return	1=successo, 0=errore	
call	sknImg.delete("myBtn")	

sknImg.allowMoveX ()

Controlla l'abilitazione al movimento su asse X di una Immagine.

sknImg.allowMoveX ("imgName", allow)		
	Controlla l'abilitazione al movimento su asse X di una Immagine. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione dell'Immagine selezionata.	
param	imgName	Nome dell'Immagine
	allow	true=consente il movimento sull'asse x dell'Immagine, false=Immagine fissa su asse x (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknImg.allowMoveX("myBtn", true) ... print ("\r\nmy allowXmove=" .. sknImg.allowMoveX("myBtn"))	

sknImg.allowMoveY ()

Controlla l'abilitazione al movimento su asse Y di una Immagine.

sknImg.allowMoveY ("imgName", allow)		
	Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione dell'Immagine selezionata.	
param	imgName	Nome dell'Immagine
	allow	true=consente il movimento sull'asse y dell'Immagine, false=Immagine fissa su asse y (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknImg.allowMoveY("myBtn", true) ... print ("\r\nmy allowYmove=" .. sknImg.allowMoveY("myBtn"))	

sknImg.allowResize ()

Controlla l'abilitazione alla modifica della dimensione di una Immagine.

sknImg.allowResize ("imgName", allow)		
	Controlla l'abilitazione alla modifica della dimensione di una Immagine. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di trasparenza dell'Immagine selezionata.	
param	imgName	Nome dell'Immagine
	allow	true=consente il ridimensionamento dell'Immagine, false=Immagine non ridimensionabile (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknImg.allowResize("myBtn", true) ... print ("\r\nmy allowResize=" .. sknImg.allowResize("myBtn"))	

sknImg.auxData ()

Controlla l'AuxData di una Immagine.

sknImg.auxData ("imgName", auxData)		
	Controlla l'AuxData di una Immagine. L'AuxData viene utilizzato per identificare una Immagine con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.	
param	imgName	Nome dell'Immagine
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknImg.auxData("myBtn", 1234) ... print ("\r\nmy auxData=" .. sknImg.auxData("myBtn"))	

sknImg.auxStr ()

Controlla l'AuxString di una Immagine.

sknImg.auxStr ("imgName", auxString)		
Controlla l'AuxString di una Immagine. L'AuxString viene utilizzato per identificare una Immagine con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	imgName	Nome dell'Immagine
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknImg.auxData("myBtn", "W1") ... print ("\r\nmy auxString=" .. sknImg.auxStr("myBtn"))	

sknImg.bringTop ()

Porta la finestra in primo piano.

sknImg.bringTop ("imgName")		
Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Immagine attive.		
param	imgName	Nome dell'Immagine
return	1=successo, 0=errore	
call	sknImg.bringTop("myBtn")	

sknImg.bringBot()

Porta la finestra in sfondo.

sknImg.bringBot ("imgName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	imgName	Nome dell'Immagine
return	1=successo, 0=errore	
call	sknImg.bringBot("myBtn")	

sknImg.autoHide()

Controlla la funzionalità di auto hide di una Immagine.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknImg.autoHide ("imgName", autoHide)		
	Controlla la funzionalità di auto hide di una Immagine. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	imgName	Nome dell'Immagine
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknImg.autoHide("myBtn", true) ... print ("\r\nmy autohide=" .. sknImg.autoHide ("myBtn"))	

sknImg.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di una Immagine.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknImg.autoHideTO ("imgName", autoHideTimeout)

	Controlla il timeout per la funzionalità di auto hide di una Immagine. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	imgName	Nome dell'Immagine
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknImg.autoHideTO("myBtn", 60) ... print ("\r\nmy autohideTO=" .. sknImg.autoHideTO ("myBtn"))	

sknImg.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di una Immagine.

sknImg.autoHideMode ("imgName", "mode")

	Controlla il tipo ti hide per la funzionalità di auto hide di una Immagine. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.	
param	imgName	Nome dell'Immagine
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknImg.autoHideMode("myBtn", "left") ... print ("\r\nmy autohideMode=" .. sknImg.autoHideMode ("myBtn"))	

sknImg.blinkEn ()

Controlla la funzionalità di blink di una Immagine.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknImg.blinkEn ("imgName", blink)		
Controlla la funzionalità di blink di una Immagine. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	imgName	Nome dell'Immagine
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknImg.blinkEn("myBtn", true) ... print ("\r\nmy blinkEn=" .. sknImg.blinkEn ("myBtn"))	

sknImg.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di una Immagine.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknImg.blinkCustEn ("imgName", blink)		
Controlla la funzionalità di blink custom o globale di una Immagine. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	imgName	Nome dell'Immagine
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknImg.blinkCustEn("myBtn", true) ... print ("\r\nmy blinkCustEn=" .. sknImg.blinkCustEn ("myBtn"))	

sknImg.autoHideCBshow ()

Controlla la funzione di callback show di una Immagine.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknImg.autoHideCBshow ("imgName", "callback")

	Controlla la funzione di callback show di una Immagine. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	imgName	Nome dell'Immagine
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknImg.autoHideCBshow("myBtn", "myCBshow") ... print ("\\r\\nmy autohideCBshow=" .. sknImg.autoHideCBshow ("myBtn"))</pre>	

sknImg.autoHideCBhide ()

Controlla la funzione di callback hide di una Immagine.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknImg.autoHideCBhide ("imgName", "callback")

	Controlla la funzione di callback hide di una Immagine. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	imgName	Nome dell'Immagine
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknImg.autoHideCBhide("myBtn", "myCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknImg.autoHideCBhide ("myBtn"))</pre>	

sknImg.aniEn ()

Controlla la funzionalità di animation di una Immagine.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknImg.aniEn ("imgName", animation)

	Controlla la funzionalità di animation di una Immagine. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.	
param	imgName	Nome dell'Immagine
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknImg.aniEn("myBtn", true) ... print ("\r\nmy aniEn=" .. sknImg.aniEn ("myBtn"))</pre>	

sknImg.anMode ()

Controlla la modalità di animation di una Immagine.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknImg.anMode ("imgName", "aniMode")

	Controlla la modalità di animation di una Immagine. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.	
param	imgName	Nome dell'Immagine
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	<pre>sknImg.anMode("myBtn", "shift") ... print ("\r\nmy aniMode=" .. sknImg.anMode ("myBtn"))</pre>	

sknImg.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknImg.autoFocusChild ("imgName", autoFocus)

	Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.	
param	imgName	Nome dell'Immagine
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknImg.autoFocusChild("myBtn", true) ... print ("\r\nmy autoFocusChild=" .. sknImg.autoFocusChild ("myBtn"))	

sknImg.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknImg.bringToTopChild ("imgName", en)

	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	imgName	Nome dell'Immagine
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknImg.bringToTopChild("myBtn", true) ... print ("\r\nmy bringToTopChild=" .. sknImg.bringToTopChild ("myBtn"))	

sknImg.group ()

Controlla l'appartenenza di un gruppo di una Immagine.

sknImg.group ("imgName", "grpName")		
Controlla l'appartenenza di un gruppo di una Immagine. Il grpName viene utilizzato per associare una Immagine con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	imgName	Nome dell'Immagine
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknImg.group("myBtn", "setGrp") ... print ("\r\nmy group=" .. sknImg.group("myBtn"))	

sknImg.subGroup ()

Controlla l'appartenenza di un sotto gruppo di una Immagine.

sknImg.subGroup ("imgName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di una Immagine. Il subgrpName viene utilizzato per associare una Immagine con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	imgName	Nome dell'Immagine
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknImg.subGroup("myBtn", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknImg.subGroup ("myBtn"))	

sknImg.keyPressCB ()

Controlla il nome della funzione di callback per il key press dell'Image.

sknImg.keyPressCB ("imgName", "callback")		
	Controlla il nome della funzione di callback per il key press dell'Image. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	imgName	Nome della image
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknImg.keyPressCB("myImg", "keyPressEvent") ... print ("\r\nmy winKeyPressCB=" .. sknImg.keyPressCB ("myImg"))	

sknImg.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release dell'Image.

sknImg.keyReleaseCB ("imgName", "callback")		
	Controlla il nome della funzione di callback per il key release dell'Image. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	imgName	Nome dell'image
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknImg.keyReleaseCB("myImg", "keyReleaseEvent") ... print ("\r\nmy winKeyReleaseCB=" .. sknImg.keyReleaseCB ("myImg"))	

sknImg.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknImg.longTouchCB ("imgName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	imgName	Nome dell'Immagine
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknImg.longTouchCB("myBtn", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknImg.longTouchCB ("myBtn"))	

sknImg.paintCB ()

Controlla il nome della funzione di callback per il paint dell'immagine.

sknImg.paintCB ("imgName", "callback")		
Controlla il nome della funzione di callback per il paint della label. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	imgName	Nome dell'immagine
	Callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknImg.paintCB("home", "RedrawCb") ... print ("\r\nmy paintCB=" .. sknImg.paintCB ("home"))	

ANIMATED IMAGE

La Classe **sknAnimage** implementa le immagini animate.

Membro	Descrizione
Control	
sknAnimage.create()	Crea una Animage
sknAnimage.delete()	Distrugge una Animage
View	
sknAnimage.show()	Visualizza una Animage
sknAnimage.hide()	Nascondi una Animage
Proprietà	
sknAnimage.pos()	Modifica la posizione di una Animage
sknAnimage.multimage()	Configura la collezione di immagini di una Animage
Permessi	
sknAnimage.allowMoveX()	Controlla l'abilitazione al movimento su asse X di una Animage
sknAnimage.allowMoveY()	Controlla l'abilitazione al movimento su asse Y di una Animage
sknAnimage.allowResize()	Controlla l'abilitazione alla modifica della dimensione di una Animage
Visibilità	
sknAnimage.bringTop()	Porta l'Animage in primo piano
sknAnimage.bringBot()	Porta l'Animage in sfondo
Aggregazione	
sknAnimage.auxData()	Controlla l'AuxData di una Animage
sknAnimage.auxStr()	Controlla l'AuxString di una Animage
sknAnimage.group()	Controlla l'appartenenza di un gruppo di una Animage
sknAnimage.subGroup()	Controlla l'appartenenza di un sotto gruppo di una Animage
Auto Hide	
sknAnimage.autoHide()	Controlla la funzionalità di auto hide di una Animage
sknAnimage.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di una Animage
sknAnimage.autoHideMode()	Controlla il tipo ti hide per la funzionalità di auto hide di una Animage
Blink	
sknAnimage.blinkEn()	Controlla la funzionalità di blink di una Animage
sknAnimage.blinkCustEn()	Controlla la funzionalità di blink custom o globale di una Animage
Auto Hide callback	

<code>sknImage.autoHideCBshow()</code>	Controlla la funzione di callback hide di una Animage
<code>sknImage.autoHideCBhide()</code>	Controlla la funzione di callback show di una Animage
Animazione	
<code>sknImage.aniEn()</code>	Controlla la funzionalità di animation di una Animage
<code>sknImage.aniMode()</code>	Controlla la modalità di animation di una Animage
Child	
<code>sknImage.autoFocusChild()</code>	Controlla la funzionalità di autofocus per i child Animage
<code>sknImage.bringToTopChild()</code>	Controlla la funzionalità di brig to top per i child Animage

Classe [sknImage]

sknImage.create()

Crea una Immagine animata.

sknImage.create("aniName", "motherWin", xpos, ypos, xsize, ysize)		
	Crea una Immagine animata.	
param	aniName	Nome dell'Immagine animata
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza dell'Immagine animata
	ysize	altezza dell'Immagine animata
return	1=successo, 0=errore	
call	sknImage.create("imgAni", "winHome", 30, 50, 42, 42)	

sknImage.show()

Visualizza una Immagine animata.

sknImage.show ("aniName")		
	Visualizza una Immagine animata.	
param	aniName	nome dell'Immagine animata
return	1=successo, 0=errore	
call	sknImage.show("myImgAni")	

sknImage.hide()

Nasconde una Immagine animata.

sknImage.hide ("aniName")		
	Nasconde una Immagine animata.	
param	aniName	Nome dell'Immagine animata
return	1=successo, 0=errore	
call	sknImage.hide("myImgAni")	

sknImage.pos ()

Modifica o ritorna la posizione di una Immagine animata.

sknImage.pos ("aniName", x, y)		
Modifica o ritorna la posizione di una Immagine animata. Se si omettono x e y la funzione riporta la posizione corrente.		
param	aniName	Nome dell'Immagine animata
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknImage.pos("myImgAni", 10, 50) ... print ("\r\nmy pos=" .. sknImage.pos("myImgAni"))	

sknImage.multiimage ()

Configura l'immagine di una Animage.

sknImage.multiimage ("aniName", "imageList")		
Configura l'immagine di una Animage.		
param	aniName	Nome dell'Immagine animata
	imageList	nome della collezione di immagini
return	1=successo, 0=errore	
call	sknImage.multiimage("milInfo", "waitInfo")	

sknImage.delete ()

Distrugge una Immagine animata.

Questo metodo non rimuove gli oggetti figli della finestra.

sknImage.delete ("aniName")		
Distrugge una Immagine animata.		
param	aniName	Nome dell'Immagine animata
return	1=successo, 0=errore	
call	sknImage.delete("myImgAni")	

sknImage.allowMoveX ()

Controlla l'abilitazione al movimento su asse X di una Immagine animata.

sknImage.allowMoveX ("aniName", allow)		
	Controlla l'abilitazione al movimento su asse X di una Immagine animata. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione dell'Immagine animata selezionata.	
param	aniName	Nome dell'Immagine animata
	allow	true=consente il movimento sull'asse x dell'Immagine animata, false=Immagine animata fissa su asse x (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknImage.allowMoveX("myImgAni", true) ... print ("\r\nmy allowXmove=" .. sknImage.allowMoveX("myImgAni"))	

sknImage.allowMoveY ()

Controlla l'abilitazione al movimento su asse Y di una Immagine animata.

sknImage.allowMoveY ("aniName", allow)		
	Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di abilitazione dell'Immagine animata selezionata.	
param	aniName	Nome dell'Immagine animata
	allow	true=consente il movimento sull'asse y dell'Immagine animata, false=Immagine animata fissa su asse y (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknImage.allowMoveY("myImgAni", true) ... print ("\r\nmy allowYmove=" .. sknImage.allowMoveY("myImgAni"))	

sknImage.allowResize ()

Controlla l'abilitazione alla modifica della dimensione di una Immagine animata.

sknImage.allowResize ("aniName", allow)		
	Controlla l'abilitazione alla modifica della dimensione di una Immagine animata. Se non viene valorizzato il campo allow, la funzione ritorna il valore corrente di trasparenza dell'Immagine animata selezionata.	
param	aniName	Nome dell'Immagine animata
	allow	true=consente il ridimensionamento dell'Immagine animata, false=Immagine animata non ridimensionabile (opt)
return	allow o 1=successo, 0=errore per Set	
call	sknImage.allowResize("myImgAni", true) ... print ("\r\nmy allowResize=" .. sknImage.allowResize("myImgAni"))	

sknImage.auxData ()

Controlla l'AuxData di una Immagine animata.

sknImage.auxData ("aniName", auxData)		
	Controlla l'AuxData di una Immagine animata. L'AuxData viene utilizzato per identificare una Immagine animata con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.	
param	aniName	Nome dell'Immagine animata
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknImage.auxData("myImgAni", 1234) ... print ("\r\nmy auxData=" .. sknImage.auxData("myImgAni"))	

sknImage.auxStr ()

Controlla l'AuxString di una Immagine animata.

sknImage.auxStr ("aniName", auxString)

	Controlla l'AuxString di una Immagine animata. L'AuxString viene utilizzato per identificare una Immagine animata con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.	
param	aniName	Nome dell'Immagine animata
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknImage.auxData("myImgAni", "W1") ... print ("\r\nmy auxString=" .. sknImage.auxStr("myImgAni"))	

sknImage.bringTop ()

Porta la finestra in primo piano.

sknImage.bringTop ("aniName")

	Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Immagine animata attive.	
param	aniName	Nome dell'Immagine animata
return	1=successo, 0=errore	
call	sknImage.bringTop("myImgAni")	

sknImage.bringBot ()

Porta la finestra in sfondo.

sknImage.bringBot ("aniName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	aniName	Nome dell'Immagine animata
return	1=successo, 0=errore	
call	sknImage.bringBot("myImgAni")	

sknImage.autoHide ()

Controlla la funzionalità di auto hide di una Immagine animata.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknImage.autoHide ("aniName", autoHide)		
	Controlla la funzionalità di auto hide di una Immagine animata. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	aniName	Nome dell'Immagine animata
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknImage.autoHide("myImgAni", true) ... print ("\r\nmy autohide=" .. sknImage.autoHide ("myImgAni"))	

sknAnimage.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di una Immagine animata.
Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknAnimage.autoHideTO ("aniName", autoHideTimeout)

	Controlla il timeout per la funzionalità di auto hide di una Immagine animata. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	aniName	Nome dell'Immagine animata
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknAnimage.autoHideTO("myImgAni", 60) ... print ("\r\nmy autohideTO=" .. sknAnimage.autoHideTO ("myImgAni"))	

sknAnimage.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di una Immagine animata.

sknAnimage.autoHideMode ("aniName", "mode")

	Controlla il tipo ti hide per la funzionalità di auto hide di una Immagine animata. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.	
param	aniName	Nome dell'Immagine animata
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknAnimage.autoHideMode("myImgAni", "left") ... print ("\r\nmy autohideMode=" .. sknAnimage.autoHideMode ("myImgAni"))	

sknImage.blinkEn ()

Controlla la funzionalità di blink di una Immagine animata.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknImage.blinkEn ("aniName", blink)		
Controlla la funzionalità di blink di una Immagine animata. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	aniName	Nome dell'Immagine animata
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknImage.blinkEn("myImgAni", true) ... print ("\r\nmy blinkEn=" .. sknImage.blinkEn ("myImgAni"))	

sknImage.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di una Immagine animata.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknImage.blinkCustEn ("aniName", blink)		
Controlla la funzionalità di blink custom o globale di una Immagine animata. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	aniName	Nome dell'Immagine animata
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknImage.blinkCustEn("myImgAni", true) ... print ("\r\nmy blinkCustEn=" .. sknImage.blinkCustEn ("myImgAni"))	

sknAnimage.autoHideCBshow ()

Controlla la funzione di callback show di una Immagine animata.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknAnimage.autoHideCBshow ("aniName", "callback")

	Controlla la funzione di callback show di una Immagine animata. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	aniName	Nome dell'Immagine animata
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknAnimage.autoHideCBshow("myImgAni", "myCBshow") ... print ("\r\nmy autohideCBshow=" .. sknAnimage.autoHideCBshow ("myImgAni"))	

sknAnimage.autoHideCBhide ()

Controlla la funzione di callback hide di una Immagine animata.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknAnimage.autoHideCBhide ("aniName", "callback")

	Controlla la funzione di callback hide di una Immagine animata. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	aniName	Nome dell'Immagine animata
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknAnimage.autoHideCBhide("myImgAni", "myCBshow") ... print ("\r\nmy autohideCBhide=" .. sknAnimage.autoHideCBhide ("myImgAni"))	

sknAnimage.aniEn ()

Controlla la funzionalità di animation di una Immagine animata.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknAnimage.aniEn ("aniName", animation)

	Controlla la funzionalità di animation di una Immagine animata. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.	
param	aniName	Nome dell'Immagine animata
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknAnimage.aniEn("myImgAni", true) ... print ("\\r\\nmy aniEn=" .. sknAnimage.aniEn ("myImgAni"))</pre>	

sknAnimage.aniMode ()

Controlla la modalità di animation di una Immagine animata.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknAnimage.aniMode ("aniName", "aniMode")

	Controlla la modalità di animation di una Immagine animata. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.	
param	aniName	Nome dell'Immagine animata
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	<pre>sknAnimage.aniMode("myImgAni", "shift") ... print ("\\r\\nmy aniMode=" .. sknAnimage.aniMode ("myImgAni"))</pre>	

sknAnimage.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknAnimage.autoFocusChild ("aniName", autoFocus)		
	Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.	
param	aniName	Nome dell'Immagine animata
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknAnimage.autoFocusChild("myImgAni", true) ... print ("\r\nmy autoFocusChild=" .. sknAnimage.autoFocusChild ("myImgAni"))	

sknAnimage.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknAnimage.bringToTopChild ("aniName", en)		
	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	aniName	Nome dell'Immagine animata
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknAnimage.bringToTopChild("myImgAni", true) ... print ("\r\nmy bringToTopChild=" .. sknAnimage.bringToTopChild ("myImgAni"))	

sknImage.group ()

Controlla l'appartenenza di un gruppo di una Immagine animata.

sknImage.group ("aniName", "grpName")		
Controlla l'appartenenza di un gruppo di una Immagine animata. Il grpName viene utilizzato per associare una Immagine animata con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	aniName	Nome dell'Immagine animata
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknImage.group("myImgAni", "setGrp") ... print ("\\r\\nmy group=" .. sknImage.group("myImgAni"))	

sknImage.subGroup ()

Controlla l'appartenenza di un sotto gruppo di una Immagine animata.

sknImage.subGroup ("aniName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di una Immagine animata. Il subgrpName viene utilizzato per associare una Immagine animata con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	aniName	Nome dell'Immagine animata
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknImage.subGroup("myImgAni", "mySubGrp") ... print ("\\r\\nmy subGrp=" .. sknImage.subGroup ("myImgAni"))	

CHECKBOX

La Classe **sknChkbox** implementa le CheckBox.

Membro	Descrizione
Control	
sknChkbox.create()	Crea un CheckBox
sknChkbox.delete()	Distrugge un CheckBox
View	
sknChkbox.show()	Visualizza un CheckBox
sknChkbox.hide()	Nascondi un CheckBox
Proprietà	
sknChkbox.size()	Modifica le dimensioni di un CheckBox
sknChkbox.enable()	Controlla l'enable del CheckBox
sknChkbox.txtcolors()	Modifica i colori del testo di un CheckBox
sknChkbox.boxbkcolors()	Modifica i colori di sfondo del box di un CheckBox
sknChkbox.focuscolor()	Modifica i colori di focus di un CheckBox
sknChkbox.text()	Modifica il testo di un CheckBox
sknChkbox.font()	Configura il font di un CheckBox
sknChkbox.numstates()	Configura il numero di states di un CheckBox
sknChkbox.state()	Configura lo state di un CheckBox
sknChkbox.align()	Configura l'allineamento del test di un CheckBox
sknChkbox.check()	Configura a checked un CheckBox
sknChkbox.uncheck()	Configura a unchecked un CheckBox
sknChkbox.image()	Configura una immagine di un CheckBox
sknChkbox.spacing()	Configura lo spazio fra il checkbox ed il testo un CheckBox
sknChkbox.pos()	Modifica la posizione di un CheckBox
sknChkbox.focussable()	Configura la funzionalità focussable di un CheckBox
Visibilità	
sknChkbox.bringTop()	Porta il CheckBox in primo piano
sknChkbox.bringBot()	Porta il CheckBox in sfondo
Aggregazione	
sknChkbox.auxData()	Controlla l'AuxData di un CheckBox
sknChkbox.auxString()	Controlla l'AuxString di un CheckBox
sknChkbox.group()	Controlla l'appartenenza di un gruppo di un CheckBox

sknChkbox.subGroup()	Controlla l'appartenenza di un sotto gruppo di un CheckBox
Auto Hide	
sknChkbox.autoHide()	Controlla la funzionalità di auto hide di un CheckBox
sknChkbox.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di un CheckBox
sknChkbox.autoHideMode()	Controlla il tipo di hide per la funzionalità di auto hide di un CheckBox
Blink	
sknChkbox.blinkEn()	Controlla la funzionalità di blink di un CheckBox
sknChkbox.blinkCustEn()	Controlla la funzionalità di blink custom o globale di un CheckBox
Auto Hide callbacks	
sknChkbox.autoHideCBshow()	Controlla la funzione di callback hide di un CheckBox
sknChkbox.autoHideCBhide()	Controlla la funzione di callback show di un CheckBox
Animazione	
sknChkbox.aniEn()	Controlla la funzionalità di animation di un CheckBox
sknChkbox.aniMode()	Controlla la modalità di animation di un CheckBox
Child	
sknChkbox.autoFocusChild()	Controlla la funzionalità di autofocus per i child CheckBox
sknChkbox.bringToTopChild()	Controlla la funzionalità di bring to top per i child CheckBox
Callbacks	
sknChkbox.longTouchCB()	Controlla il nome della funzione di callback per un long touch
Sound	
sknChkbox.sound()	Configura il suono per l'evento press

Classe [sknChkbox]

sknChkbox.create()

Crea una Checkbox.

sknChkbox.create("chkName", "motherWin", xpos, ypos, xsize, ysize)		
	Crea una Checkbox.	
param	chkName	Nome della Checkbox
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza della Checkbox
	ysize	altezza della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.create("btnInfo", "winHome", 30, 50, 42, 42)	

sknChkbox.show ()

Visualizza una Checkbox.

sknChkbox.show ("chkName")		
	Visualizza una Checkbox.	
param	chkName	nome della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.show("myBtn")	

sknChkbox.hide ()

Nasconde una Checkbox.

sknChkbox.hide ("chkName")		
	Nasconde una Checkbox.	
param	chkName	Nome della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.hide("myBtn")	

sknChkbox.size ()

Modifica o ritorna le dimensioni di una Checkbox.

sknChkbox.size ("chkName", x, y)		
	Modifica o ritorna le dimensioni di una Checkbox. Se si omette x e y la funzione riporta quelli correnti.	
param	chkName	Nome della Checkbox
	x	larghezza (opt)
	Y	altezza (opt)
return	Size x, y o 1=successo, 0=errore per Set	
call	sknChkbox.size("myBtn", 200, 100) ... x,y = sknChkbox.size("myBtn")	

sknChkbox.enable ("chkName", enable)

Controlla l'enable di un CheckBox.

sknChkbox.enable ("chkName", enable)		
	Controlla l'enable di un CheckBox. Se non viene valorizzato il campo enable, la funzione ritorna il valore corrente di enable della checkbox selezionata.	
param	chkName	Nome della Checkbox
	enable	true=checkbox abilitato, false=checkbox disabilitato
return	enable o 1=successo, 0=errore per Set	
call	sknChkbox.enable("infoChk", true) ... print("\r\nmyChkEnable=", .. sknLbl.enable("infoChk"))	

sknChkbox.txtcolors ()

Modifica i colori del testo di un CheckBox.

sknChkbox.txtcolors ("chkName", color, colorPress)		
	Modifica i colori del testo di un CheckBox.	
param	chkName	Nome della Checkbox
	color	colore del testo del checkbox non premuto (numerico o stringa per colore predefinito)
	colorPress	colore del testo del checkbox premuto (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknChkbox.txtcolors("btnHIp", "black", "white")	

sknChkbox.boxbkcolor ()

Modifica i colori di sfondo del box di un CheckBox.

sknChkbox.boxbkcolor ("chkName", color, state)		
	Modifica i colori di sfondo del box di un CheckBox.	
param	chkName	Nome della Checkbox
	color	colore di sfondo del box del checkbox (numerico o stringa per colore predefinito)
	state	stato del checkbox {0=disabilitato, 1=abilitato}
return	1=successo, 0=errore	
call	sknChkbox.boxbkcolor("btnHIp", "blue", 1)	

sknChkbox.focuscolor ()

Modifica i colori di focus del box di un CheckBox.

sknChkbox.focuscolor ("chkName", color)		
	Modifica i colori di focus del box di un CheckBox.	
param	chkName	Nome della Checkbox
	color	colore di focus del checkbox (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknChkbox.focuscolor("btnHlp", "blue", 1)	

sknChkbox.font()

Configura il font di una Checkbox.

sknChkbox.font ("chkName", "fontName")		
	Configura il font di una Checkbox.	
param	chkName	Nome della Checkbox
	fontName	nome font
return	1=successo, 0=errore	
call	sknChkbox.font("btnHlp", "Arial14")	

sknChkbox.numstates ()

Configura il numero di states di un CheckBox.

sknChkbox.numstates ("chkName", numStates)		
	Configura il numero di states di un CheckBox.	
param	chkName	Nome della Checkbox
	numStates	numero di stati della checkbox {2=checked e unchecked, 3=checked – unchecked - third}
return	1=successo, 0=errore	
call	sknChkbox.numstates("btnHlp", 3)	

sknChkbox.state ()

Configura lo state di un CheckBox.

sknChkbox.state ("chkName", state)		
	Configura lo state di un CheckBox.	
param	chkName	Nome della Checkbox
	state	stato della checkbox {0=unchecked, 1=checked, 3=third}
return	1=successo, 0=errore	
call	sknChkbox.state("btnHlp", 2)	

sknChkbox.text()

Modifica il testo di una Checkbox.

sknChkbox.text ("chkName", "text")		
	Modifica il testo di una Checkbox.	
param	chkName	Nome della Checkbox
	text	testo della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.text("btnHlp", "HELP")	

sknChkbox.align ()

Configura l'allineamento di un CheckBox.

sknChkbox.align ("chkName", "textAlign")		
	Configura l'allineamento di un CheckBox.	
param	chkName	Nome della Checkbox
	textAlign	allineamento {"HorLeft_Vert_Top", "HorLeft_VertBot", "HorLeft_VertCenter", "HorRight_Vert_Top", "HorRight_VertBot", "HorRight_VertCenter", "HorCenter_Vert_Top", "HorCenter_VertBot", "HorCenter_VertCenter"}
return	1=successo, 0=errore	
call	sknChkbox.align("lblHlp", "HorLeft_Vert_Top")	

sknChkbox.check ()

Configura a checked un CheckBox.

sknChkbox.check ("chkName")		
	Configura a checked un CheckBox.	
param	chkName	Nome della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.check("chkMy")	

sknChkbox.uncheck ()

Configura a unchecked un CheckBox.

sknChkbox.uncheck ("chkName")		
	Configura a unchecked un CheckBox.	
param	chkName	Nome della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.uncheck("chkMy")	

sknChkbox.image ()

Configura l'immagine di un CheckBox per uno specifico stato.

sknChkbox.image ("chkName", "collection", "image", state)		
	Configura l'immagine di un CheckBox per uno specifico stato. Per ogni stato del checkbox viene associata una specifica immagine.	
param	chkName	Nome della Checkbox
	collection	nome della collezione di immagini
	image	nome dell'immagine
	state	stato a cui associare l'immagine {0=unchecked, 1=checked, 3=third}
return	1=successo, 0=errore	
call	sknChkbox.image("chkInfo", "icons", "alert", 1)	

sknChkbox.spacing ()

Configura lo spazio fra il checkbox ed il testo un CheckBox.

sknChkbox.spacing ("chkName", spacing)		
	Configura lo spazio fra il checkbox ed il testo un CheckBox.	
param	chkName	Nome della Checkbox
	spacing	spazio in pixels fra l'immagine del checkbox e il testo
return	1=successo, 0=errore	
call	sknChkbox.spacing("chkInfo", 8)	

sknChkbox.pos ()

Modifica o ritorna la posizione di una Checkbox.

sknChkbox.pos ("chkName", x, y)		
	Modifica o ritorna la posizione di una Checkbox. Se si omettono x e y la funzione riporta la posizione corrente.	
param	chkName	Nome della Checkbox
	x	origine asse x (opt)
	Y	origine asse y (opt)
return	x,y o 1=successo, 0=errore per Set	
call	sknChkbox.pos("myBtn", 10, 50) ... print ("\r\nmy pos=" .. sknChkbox.pos("myBtn"))	

sknChkbox.delete ()

Distrugge una Checkbox.

Questo metodo non rimuove gli oggetti figli della finestra.

sknChkbox.delete ("chkName")		
	Distrugge una Checkbox.	
param	chkName	Nome della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.delete("myBtn")	

sknChkbox.auxData ()

Controlla l'AuxData di una Checkbox.

sknChkbox.auxData ("chkName", auxData)		
Controlla l'AuxData di una Checkbox. L'AuxData viene utilizzato per identificare una Checkbox con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.		
param	chkName	Nome della Checkbox
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknChkbox.auxData("myBtn", 1234) ... print ("\r\nmy auxData=" .. sknChkbox.auxData("myBtn"))	

sknChkbox.auxStr ()

Controlla l'AuxString di una Checkbox.

sknChkbox.auxStr ("chkName", auxString)		
Controlla l'AuxString di una Checkbox. L'AuxString viene utilizzato per identificare una Checkbox con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	chkName	Nome della Checkbox
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknChkbox.auxData("myBtn", "W1") ... print ("\r\nmy auxString=" .. sknChkbox.auxStr("myBtn"))	

sknChkbox.focussable()

Configura la funzionalità focussable di una Checkbox.

sknChkbox.focussable("chkName", focussable)		
	Configura la funzionalità focussable di una Checkbox. Questa funzionalità è utile quando si vuole scorrere automaticamente fra vari oggetti senza un touch esplicito.	
param	chkName	Nome della Checkbox
	focussable	true=focussable, false=non focussable
return	1=successo, 0=errore	
call	sknChkbox.focussable("btnInfo", false)	

sknChkbox.bringTop()

Porta la finestra in primo piano.

sknChkbox.bringTop ("chkName")		
	Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Checkbox attive.	
param	chkName	Nome della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.bringTop("myBtn")	

sknChkbox.bringBot()

Porta la finestra in sfondo.

sknChkbox.bringBot ("chkName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	chkName	Nome della Checkbox
return	1=successo, 0=errore	
call	sknChkbox.bringBot("myBtn")	

sknChkbox.autoHide ()

Controlla la funzionalità di auto hide di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknChkbox.autoHide ("chkName", autoHide)		
	Controlla la funzionalità di auto hide di una Checkbox. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	chkName	Nome della Checkbox
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknChkbox.autoHide("myBtn", true) ... print ("\r\nmy autohide=" .. sknChkbox.autoHide ("myBtn"))	

sknChkbox.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknChkbox.autoHideTO ("chkName", autoHideTimeout)		
	Controlla il timeout per la funzionalità di auto hide di una Checkbox. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	chkName	Nome della Checkbox
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknChkbox.autoHideTO("myBtn", 60) ... print ("\r\nmy autohideTO=" .. sknChkbox.autoHideTO ("myBtn"))	

sknChkbox.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di una Checkbox.

sknChkbox.autoHideMode ("chkName", "mode")		
	Controlla il tipo ti hide per la funzionalità di auto hide di una Checkbox. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.	
param	chkName	Nome della Checkbox
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknChkbox.autoHideMode("myBtn", "left") ... print ("\r\nmy autohideMode=" .. sknChkbox.autoHideMode ("myBtn"))	

sknChkbox.blinkEn ()

Controlla la funzionalità di blink di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknChkbox.blinkEn ("chkName", blink)		
	Controlla la funzionalità di blink di una Checkbox. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.	
param	chkName	Nome della Checkbox
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknChkbox.blinkEn("myBtn", true) ... print ("\r\nmy blinkEn=" .. sknChkbox.blinkEn ("myBtn"))	

sknChkbox.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknChkbox.blinkCustEn ("chkName", blink)		
	Controlla la funzionalità di blink custom o globale di una Checkbox. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.	
param	chkName	Nome della Checkbox
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknChkbox.blinkCustEn("myBtn", true) ... print ("\r\nmy blinkCustEn=" .. sknChkbox.blinkCustEn ("myBtn"))	

sknChkbox.autoHideCBshow ()

Controlla la funzione di callback show di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknChkbox.autoHideCBshow ("chkName", "callback")		
	Controlla la funzione di callback show di una Checkbox. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	chkName	Nome della Checkbox
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknChkbox.autoHideCBshow("myBtn", "myCBshow") ... print ("\r\nmy autohideCBshow=" .. sknChkbox.autoHideCBshow ("myBtn"))	

sknChkbox.autoHideCBhide ()

Controlla la funzione di callback hide di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknChkbox.autoHideCBhide ("chkName", "callback")

	Controlla la funzione di callback hide di una Checkbox. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	chkName	Nome della Checkbox
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknChkbox.autoHideCBhide("myBtn", "myCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknChkbox.autoHideCBhide ("myBtn"))</pre>	

sknChkbox.aniEn ()

Controlla la funzionalità di animation di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknChkbox.aniEn ("chkName", animation)

	Controlla la funzionalità di animation di una Checkbox. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.	
param	chkName	Nome della Checkbox
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknChkbox.aniEn("myBtn", true) ... print ("\\r\\nmy aniEn=" .. sknChkbox.aniEn ("myBtn"))</pre>	

sknChkbox.anMode ()

Controlla la modalità di animation di una Checkbox.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknChkbox.anMode ("chkName", "aniMode")

Controlla la modalità di animation di una Checkbox. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	chkName	Nome della Checkbox
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	sknChkbox.anMode("myBtn", "shift") ... print ("\r\nmy aniMode=" .. sknChkbox.anMode ("myBtn"))	

sknChkbox.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknChkbox.autoFocusChild ("chkName", autoFocus)

Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.		
param	chkName	Nome della Checkbox
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknChkbox.autoFocusChild("myBtn", true) ... print ("\r\nmy autoFocusChild=" .. sknChkbox.autoFocusChild ("myBtn"))	

sknChkbox.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknChkbox.bringToTopChild ("chkName", en)		
Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.		
param	chkName	Nome della Checkbox
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknChkbox.bringToTopChild("myBtn", true) ... print ("\r\nmy bringToTopChild=" .. sknChkbox.bringToTopChild ("myBtn"))	

sknChkbox.group ()

Controlla l'appartenenza di un gruppo di una Checkbox.

sknChkbox.group ("chkName", "grpName")		
Controlla l'appartenenza di un gruppo di una Checkbox. Il grpName viene utilizzato per associare una Checkbox con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	chkName	Nome della Checkbox
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknChkbox.group("myBtn", "setGrp") ... print ("\r\nmy group=" .. sknChkbox.group("myBtn"))	

sknChkbox.subGroup ()

Controlla l'appartenenza di un sotto gruppo di una Checkbox.

sknChkbox.subGroup ("chkName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di una Checkbox. Il subgrpName viene utilizzato per associare una Checkbox con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	chkName	Nome della Checkbox
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknChkbox.subGroup("myBtn", "mySubGrp") ... print ("\\r\\nmy subGrp=" .. sknChkbox.subGroup ("myBtn"))	

sknChkbox.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknChkbox.longTouchCB ("chkName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	chkName	Nome della Checkbox
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknChkbox.longTouchCB("myBtn", "LTcb") ... print ("\\r\\nmy longTouchCB=" .. sknChkbox.longTouchCB ("myBtn"))	

sknChkbox.sound ()

Configura il suono per l'evento press.

Funzionalità condizionata dalla profilazione SKN_SOUND_EFFECTS_ON == 1

sknChkbox.sound ("lblName", "soundName")		
	Configura il suono per l'evento press.	
param	lblName	Nome della Checkbox
	soundName	Nome suono
return	1=successo, 0=errore	
call	sknChkbox.sound("closeBtn", "soundEnd")	

PROGRESS BAR

La Classe `sknProgbar` implementa la Progress Bar.

Membro	Descrizione
Control	
<code>sknProgbar.create()</code>	Crea una Progressbar
<code>sknProgbar.delete()</code>	Distrugge una Progressbar
View	
<code>sknProgbar.show()</code>	Visualizza una Progressbar
<code>sknProgbar.hide()</code>	Nasconde una Progressbar
Proprietà	
<code>sknProgbar.setlimits()</code>	Configura i valori min e max di una Progressbar
<code>sknProgbar.setValue()</code>	Configura il valore corrente di una Progressbar
<code>sknProgbar.setText()</code>	Modifica il testo di una Progressbar
<code>sknProgbar.setAspect()</code>	Modifica l'aspetto una Progressbar
Visibilità	
<code>sknProgbar.bringTop()</code>	Porta il progressbar in primo piano
<code>sknProgbar.bringBot()</code>	Porta il progressbar in sfondo
<code>sknProgbar.refresh()</code>	Forza il refresh della progressbar
Aggregazione	
<code>sknProgbar.auxData()</code>	Controlla l'AuxData di una Progressbar
<code>sknProgbar.auxStr()</code>	Controlla l'AuxString di una Progressbar
<code>sknProgbar.group()</code>	Controlla l'appartenenza di un gruppo di una Progressbar
<code>sknProgbar.subGroup()</code>	Controlla l'appartenenza di un sotto gruppo di una Progressbar
Auto Hide	
<code>sknProgbar.autoHide()</code>	Controlla la funzionalità di auto hide di una Progressbar
<code>sknProgbar.autoHideTO()</code>	Controlla il timeout per la funzionalità di auto hide di una Progressbar
<code>sknProgbar.autoHideMode()</code>	Controlla il tipo ti hide per la funzionalità di auto hide di una Progressbar
Blink	
<code>sknProgbar.blinkEn()</code>	Controlla la funzionalità di blink di una Progressbar
<code>sknProgbar.blinkCustEn()</code>	Controlla la funzionalità di blink custom o globale di una Progressbar
Auto Hide callbacks	
<code>sknProgbar.autoHideCBshow()</code>	Controlla la funzione di callback hide di una Progressbar
<code>sknProgbar.autoHideCBhide()</code>	Controlla la funzione di callback show di una Progressbar

Animazione	
sknProgbar.aniEn()	Controlla la funzionalità di animation di una Progressbar
sknProgbar.aniMode()	Controlla la modalità di animation di una Progressbar
Child	
sknProgbar.autoFocusChild()	Controlla la funzionalità di autofocus per i child progressbar
sknProgbar.bringToTopChild()	Controlla la funzionalità di brig to top per i child progressbar
Callbacks	
sknProgbar.longTouchCB()	Controlla il nome della funzione di callback per un long touch

Classe [sknProgbar]

sknProgbar.create()

Crea la Progress Bar.

sknProgbar.create("prgbName", "motherWin", xpos, ypos, xsize, ysize)		
Crea la Progress Bar.		
param	prgbName	Nome della Progress Bar
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza della Progress Bar
	ysize	altezza della Progress Bar
return	1=successo, 0=errore	
call	sknProgbar.create("myProgBar", "winHome", 30, 50, 42, 42)	

sknProgbar.delete ()

Distrugge la Progress Bar.

sknProgbar.delete ("prgbName")		
Distrugge la Progress Bar.		
param	prgbName	Nome della Progress Bar
return	1=successo, 0=errore	
call	sknProgbar.delete("myProgBar")	

sknProgbar.setlimits ()

Configura i valori min e max di una Progressbar.

sknProgbar.setlimits ("progbarName", minValue, maxValue, defValue)		
	Configura i valori min e max di una Progressbar.	
param	progbName	Nome della Checkbox
	minValue	valore minimo {-32000 to +32000}
	maxValue	valore massimo {-32000 to +32000}
	defValue	valore di default {-32000 to +32000}
return	1=successo, 0=errore	
call	sknProgbar.setlimits("myProgb", 0, 100, 0)	

sknProgbar.setvalue ()

Configura il valore corrente di una Progressbar.

sknProgbar.setvalue ("progbarName", value)		
	Configura il valore corrente di una Progressbar.	
param	progbName	Nome della Checkbox
	value	valore {-32000 to +32000}
return	1=successo, 0=errore	
call	sknProgbar.getvalue("myProgb", 15)	

sknProgbar.settext()

Modifica il testo di la Progress Bar.

sknProgbar.settext("prgbName", "text")		
	Modifica il testo di la Progress Bar.	
param	prgbName	Nome della Progress Bar
	text	testo della Progress Bar
return	1=successo, 0=errore	
call	sknProgbar.settext("btnHlp", "LOADING...")	

sknProgbar.setaspect ()

Modifica l'aspetto di una Progressbar.

```
sknProgbar.setaspect ("progbName", leftBarColor, rightBarColor,  
"textFont", "textAlign", textColorLeftBar, textColorRightBar)
```

	Modifica l'aspetto di una Progressbar.	
param	progbName	Nome della Checkbox
	leftBarColor	colore della progressbar a sinistra (numerico o stringa per colore predefinito)
	rightBarColor	colore della progressbar a destra (numerico o stringa per colore predefinito)
	textFont	font del testo
	textAlign	allineamento del testo {"Left", "Center", "Right"}
	textColorLeftBar	colore del testo nella parte di barra da completare (numerico o stringa per colore predefinito)
	textColorRightBar	colore del testo nella parte di barra completata (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknProgbar.setaspect("progbar","green","red", "Arial14", "left","magenta","white")	

sknProgbar.show ()

Visualizza la Progress Bar.

sknProgbar.show ("prgbName")	
	Visualizza la Progress Bar.
param	prgbName nome della Progress Bar
return	1=successo, 0=errore
call	sknProgbar.show("myProgBar")

sknProgbar.hide ()

Nasconde la Progress Bar.

sknProgbar.hide ("prgbName")	
	Nasconde la Progress Bar.
param	prgbName Nome della Progress Bar
return	1=successo, 0=errore
call	sknProgbar.hide("myProgBar")

sknProgbar.bringTop ()

Porta la finestra in primo piano.

sknProgbar.bringTop ("prgbName")	
	Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Progress Bar attive.
param	prgbName Nome della Progress Bar
return	1=successo, 0=errore
call	sknProgbar.bringTop("myProgBar")

sknProgbar.bringBot ()

Porta la finestra in sfondo.

sknProgbar.bringBot ("prgbName")	
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.
param	prgbName Nome della Progress Bar
return	1=successo, 0=errore
call	sknProgbar.bringBot("myProgBar")

sknProgbar.refresh ()

Forza il ridisegno di una Progressbar.

sknProgbar.refresh ("progbName")	
	Forza il ridisegno di una Progressbar. Questo metodo viene utilizzato quando il gestore delle finestre non è in grado di determinare un cambiamento di una finestra per eseguirne automaticamente il ridisegno, ad esempio il cambiamento di un flag che durante il ridisegno della finestra coinvolge degli artefatti grafici.
param	prgbName Nome della Checkbox
return	1=successo, 0=errore
call	sknProgbar.refresh("progb")

sknProgbar.auxData ()

Controlla l'AuxData di la Progress Bar.

sknProgbar.auxData ("prgbName", auxData)		
Controlla l'AuxData di la Progress Bar. L'AuxData viene utilizzato per identificare la Progress Bar con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.		
param	prgbName	Nome della Progress Bar
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknProgbar.auxData("myProgBar", 1234) ... print ("\r\nmy auxData=" .. sknProgbar.auxData("myProgBar"))	

sknProgbar.auxStr ()

Controlla l'AuxString di la Progress Bar.

sknProgbar.auxStr ("prgbName", auxString)		
Controlla l'AuxString di la Progress Bar. L'AuxString viene utilizzato per identificare la Progress Bar con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	prgbName	Nome della Progress Bar
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknProgbar.auxData("myProgBar", "W1") ... print ("\r\nmy auxString=" .. sknProgbar.auxStr("myProgBar"))	

sknProgbar.autoHide ()

Controlla la funzionalità di auto hide di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknProgbar.autoHide ("prgbName", autoHide)		
	Controlla la funzionalità di auto hide di la Progress Bar. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	prgbName	Nome della Progress Bar
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknProgbar.autoHide("myProgBar", true) ... print ("\r\nmy autohide=" .. sknProgbar.autoHide ("myProgBar"))	

sknProgbar.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknProgbar.autoHideTO ("prgbName", autoHideTimeout)		
	Controlla il timeout per la funzionalità di auto hide di la Progress Bar. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	prgbName	Nome della Progress Bar
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknProgbar.autoHideTO("myProgBar", 60) ... print ("\r\nmy autohideTO=" .. sknProgbar.autoHideTO ("myProgBar"))	

sknProgbar.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di la Progress Bar.

sknProgbar.autoHideMode ("prgbName", "mode")		
Controlla il tipo ti hide per la funzionalità di auto hide di la Progress Bar. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.		
param	prgbName	Nome della Progress Bar
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknProgbar.autoHideMode("myProgBar", "left") ... print ("\r\nmy autohideMode=" .. sknProgbar.autoHideMode ("myProgBar"))	

sknProgbar.blinkEn ()

Controlla la funzionalità di blink di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknProgbar.blinkEn ("prgbName", blink)		
Controlla la funzionalità di blink di la Progress Bar. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	prgbName	Nome della Progress Bar
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknProgbar.blinkEn("myProgBar", true) ... print ("\r\nmy blinkEn=" .. sknProgbar.blinkEn ("myProgBar"))	

sknProgbar.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknProgbar.blinkCustEn ("prgbName", blink)		
Controlla la funzionalità di blink custom o globale di la Progress Bar. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	prgbName	Nome della Progress Bar
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknProgbar.blinkCustEn("myProgBar", true) ... print ("\r\nmy blinkCustEn=" .. sknProgbar.blinkCustEn ("myProgBar"))	

sknProgbar.autoHideCBshow ()

Controlla la funzione di callback show di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknProgbar.autoHideCBshow ("prgbName", "callback")		
Controlla la funzione di callback show di la Progress Bar. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	prgbName	Nome della Progress Bar
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknProgbar.autoHideCBshow("myProgBar", "myCBshow") ... print ("\r\nmy autohideCBshow=" .. sknProgbar.autoHideCBshow ("myProgBar"))	

sknProgbar.autoHideCBhide ()

Controlla la funzione di callback hide di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknProgbar.autoHideCBhide ("prgbName", "callback")

	Controlla la funzione di callback hide di la Progress Bar. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	prgbName	Nome della Progress Bar
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknProgbar.autoHideCBhide("myProgBar", "myCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknProgbar.autoHideCBhide ("myProgBar"))</pre>	

sknProgbar.aniEn ()

Controlla la funzionalità di animation di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknProgbar.aniEn ("prgbName", animation)

	Controlla la funzionalità di animation di la Progress Bar. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.	
param	prgbName	Nome della Progress Bar
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknProgbar.aniEn("myProgBar", true) ... print ("\\r\\nmy aniEn=" .. sknProgbar.aniEn ("myProgBar"))</pre>	

sknProgbar.anMode ()

Controlla la modalità di animation di la Progress Bar.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknProgbar.anMode ("prgbName", "aniMode")

Controlla la modalità di animation di la Progress Bar. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	prgbName	Nome della Progress Bar
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	sknProgbar.anMode("myProgBar", "shift") ... print ("\r\nmy aniMode=" .. sknProgbar.anMode ("myProgBar"))	

sknProgbar.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknProgbar.autoFocusChild ("prgbName", autoFocus)

Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.		
param	prgbName	Nome della Progress Bar
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknProgbar.autoFocusChild("myProgBar", true) ... print ("\r\nmy autoFocusChild=" .. sknProgbar.autoFocusChild ("myProgBar"))	

sknProgbar.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknProgbar.bringToTopChild ("prgbName", en)		
	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	prgbName	Nome della Progress Bar
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknProgbar.bringToTopChild("myProgBar", true) ... print ("\r\nmy bringToTopChild=" .. sknProgbar.bringToTopChild ("myProgBar"))	

sknProgbar.group ()

Controlla l'appartenenza di un gruppo di la Progress Bar.

sknProgbar.group ("prgbName", "grpName")		
	Controlla l'appartenenza di un gruppo di la Progress Bar. Il grpName viene utilizzato per associare la Progress Bar con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.	
param	prgbName	Nome della Progress Bar
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknProgbar.group("myProgBar", "setGrp") ... print ("\r\nmy group=" .. sknProgbar.group("myProgBar"))	

sknProgbar.subGroup ()

Controlla l'appartenenza di un sotto gruppo di la Progress Bar.

sknProgbar.subGroup ("prgbName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di la Progress Bar. Il subgrpName viene utilizzato per associare la Progress Bar con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	prgbName	Nome della Progress Bar
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknProgbar.subGroup("myProgBar", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknProgbar.subGroup ("myProgBar"))	

sknProgbar.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknProgbar.longTouchCB ("prgbName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	prgbName	Nome della Progress Bar
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknProgbar.longTouchCB("myProgBar", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknProgbar.longTouchCB ("myProgBar"))	

MULTI IMAGE

La Classe **sknMimage** implementa le collezioni di immagini da utilizzare anche nelle immagini animate.

Membro	Descrizione
Control	
sknMimage.new()	Crea una Multi Image
sknMimage.delete()	Distrugge una Multi Image
View	
sknMimage.addimage()	Aggiungi una immagine ad una Multi Image
sknMimage.showtime()	Configura il tempo di animazione di una Multi Image

Classe [sknMimage]

sknMimage.new ()

Crea una Multi Image.

sknMimage.new ("imageName")		
	Crea una Multi Image.	
param	imageName	Nome della Multi Image
return	1=successo, 0=errore	
call	sknMimage.new("search")	

sknMimage.delete ()

Distrugge la Multi Image.

sknMimage.delete ("imageName")		
	Distrugge la Multi Image.	
param	imageName	Nome della Multi Image
return	1=successo, 0=errore	
call	sknMimage.delete("myProgBar")	

sknMimage.addimage ()

Aggiungi una immagine ad una Multi Image.

sknMimage.addimage ("imageName", "collection", "image")		
	Aggiungi una immagine ad una Multi Image.	
param	imageName	Nome della Multi Image
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknMimage.addimage("search", "icons", "alert")	

sknMimage.showtime ()

Configura il tempo di animazione di una Multi Image.

sknMimage.showtime ("imageArrayName", showTimeMsec)

	Configura il tempo di animazione di una Multi Image.	
param	imageArrayName	Nome della Multi Image
	showTimeMsec	tempo di visualizzazione di una immagine {1 to 32000} in millisecondi
return	1=successo, 0=errore	
call	sknMimage.showtime("search", 200)	

KEYBOARD

La Classe **sknKbrd** implementa una Keyboard configurabile.

Membro	Descrizione
Control	
sknKbrd.new()	Configura la Keyboard
sknKbrd.addBtnTxt()	Configura il testo di un tasto della Keyboard
sknKbrd.addBtnImg()	Configura le immagini di un tasto della Keyboard
sknKbrd.addBtnEnd()	Finalizza un tasto della Keyboard
sknKbrd.delete()	Distrugge la Keyboard
View	
sknKbrd.show()	Visualizza la Keyboard
sknKbrd.exitFunction()	Funzione di Uscita
Proprietà	
sknKbrd.edittxt()	Configura la zona di editing della Keyboard
sknKbrd.editlen()	Configura la lunghezza del testo editabile della Keyboard
sknKbrd.seteffect()	Configura gli effetti opzionali dei tasti della Keyboard
sknKbrd.caller()	Configura il nome della finestra che ha chiamato la Keyboard
sknKbrd.image()	Configura l'immagine di sfondo della Keyboard
sknKbrd.specmodes()	Configura la modalità di editing numerico della Keyboard
sknKbrd.extratxt()	Configura una label opzionale della Keyboard
sknKbrd.cursor()	Configura il carattere del cursore

Classe [**sknKbrd**]

sknKbrd.new ()

Configura la Keyboard.

sknKbrd.new (xpos, ypos, width, height backColor textColorPress textColorUnPress)

	Configura la Keyboard. Questa operazione crea una keyboard se non esiste, altrimenti viene solo inizializzata con i nuovi parametri. La keyboard riutilizza parte dei componenti creati nella prima new se non viene invocata una delete. Per ogni tasto è necessario invocare nell'ordine: sknKbrd.addBtnTxt(), [opzionalmente sknKbrd.addBtnImg()], infine sknKbrd.addBtnEnd() per finalizzare il tasto.	
param	xpos	origine asse x
	ypos	origine asse y
	width	larghezza della Keyboard
	height	altezza della Keyboard
	backColor	colore di sfondo (numerico o stringa per colore predefinito)
	textColorPress	colore del testo dei pulsanti nello stato di premuto (numerico o stringa per colore predefinito)
	textColorUnPress	colore del testo dei pulsanti nello stato di rilasciato (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknKbrd.new(10, 10, 300, 150,"black","red","green")	

sknKbrd.addBtnTxt ()

Configura il testo di un tasto della Keyboard.

sknKbrd.addBtnTxt (x, y, xsize, ysize, "txtAlphaU", "txtAlphaL", "txtNum")		
	Configura il testo di un tasto della Keyboard. Per ogni tasto è necessario invocare nell'ordine: sknKbrd.addBtnTxt(), [opzionalmente sknKbrd.addBtnImg()], infine sknKbrd.addBtnEnd() per finalizzare il tasto.	
param	x	asse x
	y	asse y
	xsize	larghezza del tasto
	ysize	altezza del tasto
	txtAlphaU	testo del tasto in modalità Maiuscolo
	txtAlphaL	testo del tasto in modalità Minuscolo
	txtNum	testo del tasto in modalità Numerico
return	1=successo, 0=errore	
call	sknKbrd.addBtnTxt(20,40, 50, 50, "A", "a", "0")	

sknKbrd.addBtnImg ()

Configura le immagini di un tasto della Keyboard.

sknKbrd.addBtnImg ("imgColl", "imgpressU", "imgunpressU", "imgpressL", "imgunpressL", "imgpressN", "imgunpressN")		
	Configura le immagini di un tasto della Keyboard. Questo comando viene utilizzato se si utilizzano immagini nei tasti, nella modalità solo testo non è necessario invocarlo. Per ogni tasto è necessario invocare nell'ordine: sknKbrd.addBtnTxt(), [opzionalmente sknKbrd.addBtnImg()], infine sknKbrd.addBtnEnd() per finalizzare il tasto.	
param	imgColl	nome della collezione di immagini
	imgpressU	immagine del tasto premuto in modalità Maiuscolo
	imgunpressU	immagine del tasto rilasciato in modalità Maiuscolo
	imgpressL	immagine del tasto premuto in modalità Minuscolo
	imgunpressL	immagine del tasto rilasciato in modalità Minuscolo
	imgpressN	immagine del tasto premuto in modalità Numerico
	imgunpressN	immagine del tasto rilasciato in modalità Numerico
return	1=successo, 0=errore	
call	sknKbrd.addBtnImg("btns", "Apress", "Aunpr", "mApres", "mAunpr", "Opress", "Ounpr")	

sknKbrd.addbtnEnd()

Finalizza un tasto della Keyboard.

```
sknKbrd.addbtnEnd ("keyType", TextVisible, "fontName"  
"usrFunctPress", "usrFunctRelease", imageMode)
```

	Finalizza un tasto della Keyboard. Per ogni tasto è necessario invocare nell'ordine: sknKbrd.addBtnTxt(), [opzionalmente sknKbrd.addBtnImg()], infine sknKbrd.addBtnEnd() per finalizzare il tasto.	
param	keyType	tipo di tasto (la tastiera è in grado di eseguire autonomamente operazioni tipo Abort, Confirm, Delete,...) {"key"=tasto normale, "exit"=Abbandona l'editing, "ok"=Conferma l'editing e ritorna il nuovo valore, "shift"=Cambia la modalità da maiuscolo a minuscolo e viceversa "cancel"=Cancella l'ultimo carattere digitato "alphanum"=Cambia la modalità da alfanumerico a numerico e viceversa
	TextVisible	true=Rende visibile il testo del tasto
	fontName	nome del font del testo visualizzato nei tasti
	usrFunctPress	nome della funzione da chiamare nell'evento di pressione di un tasto o stringa vuota
	usrFunctRelease	nome della funzione da chiamare nell'evento di rilascio di un tasto o stringa vuota
	imageMode	true=visualizza le immagini nei tasti
return	1=successo, 0=errore	
call	sknKbrd.addBtnEnd("key",true, "Arial14", "mycbPress", "", true)	

sknKbrd.edittxt ()

Configura la zona di editing della Keyboard.

sknKbrd.edittxt (x, y, width, height, "labelAlign", backColor, textColor, borderColor, "dataExangeName", "fontName")

	Configura la zona di editing della Keyboard.	
param	x	origine asse x
	y	origine asse y
	width	larghezza dell'area di editing
	height	altezza dell'area di editing
	labelAlign	tipo di allineamento {"HorLeft_Vert_Top", "HorLeft_VertBot", "HorLeft_VertCenter", "HorRight_Vert_Top", "HorRight_VertBot", "HorRight_VertCenter", "HorCenter_Vert_Top", "HorCenter_VertBot", "HorCenter_VertCenter"}
	backColor	colore di sfondo (numerico o stringa per colore predefinito)
	textColor	colore del testo (numerico o stringa per colore predefinito)
	borderColor	Colore del bordo (numerico o stringa per colore predefinito)
	dataExangeName	nome della variabile globale per lo scambio del testo da editare. La variabile deve esistere al momento della Show
	fontName	nome del font del testo
return	1=successo, 0=errore	
call	sknKbrd.edittxt(4, 25, 100, 16, "HorLeft_Vert_Top", "white", "blu", "myExchangeVar", "Arial14")	

sknKbrd.editlen ()

Configura la lunghezza del testo editabile della Keyboard.

sknKbrd.editlen (len)		
	Configura la lunghezza del testo editabile della Keyboard.	
param	len	lunghezza massima editabile {1..100}
return	1=successo, 0=errore	
call	sknKbrd.editlen(5)	

sknKbrd.seteffect ()

Configura gli effetti opzionali dei tasti della Keyboard.

sknKbrd.seteffect (skinType, skinEffect)		
	Configura gli effetti opzionali dei tasti della Keyboard. Questa funzione viene utilizzata solo nei casi in cui sia necessario ottenere effetti grafici solo tramite le primitive della Gui.	
param	skinType	tipo di tasto della keyboard {"none", "round", "half_up", "half_bottom", "half_right", "half_left", "quart_up_left", "quart_up_right", "quart_btm_right", "quart_btm_left", "elliptic", "rect", "image_trasp"}
	skinEffect	tipo di effetto grafico del tasto {"none", "flat", "shadow", "Bott_shadow", "vawe_shadow", "glittering", "sparkling"}
return	1=successo, 0=errore	
call	sknKbrd.seteffect("hlf_up", "shadow")	

sknKbrd.caller ()

Configura il nome della finestra che ha chiamato la Keyboard.

sknKbrd.caller ("callerName")		
	Configura il nome della finestra che ha chiamato la Keyboard. Questa funzionalità permette di ottenere l'hide automatico del chiamante	
param	callerName	nome della finestra (oggetto) che ha invocato la keyboard
return	1=successo, 0=errore	
call	sknKbrd.caller("cfgWin")	

sknKbrd.exitFunction ()

Configura il nome della funzione di uscita.

sknKbrd.exitFunction ("functName")		
	Configura il nome della funzione di uscita.	
param	functName	nome della funzione di uscita
return	1=successo, 0=errore	
call	sknKbrd.exitFunction("extFunctKey")	

sknKbrd.show ()

Mostra la Keyboard.

sknKbrd.show ()		
	Mostra la Keyboard.	
param		
return	1=successo, 0=errore	
call	sknKbrd.show()	

sknKbrd.image ()

Configura l'immagine di sfondo della Keyboard.

sknKbrd.image ("collection", "image")		
	Configura l'immagine di sfondo della Keyboard. Da utilizzare solo se necessaria un'immagine di sfondo nella keyboard	
param	collection	nome della collezione di immagini
	image	immagine di sfondo della keyboard
return	1=successo, 0=errore	
call	sknKbrd.image("img", "bkKbrd")	

sknKbrd.delete ()

Distrugge la Keyboard.

sknKbrd.delete ()	
	Distrugge la Keyboard.
param	
return	1=successo, 0=errore
call	sknKbrd.delete()

sknKbrd.specmodes ()

Configura la modalità di editing numerico della Keyboard.

sknKbrd.specmodes (allowLeftZero, setZeroIsEmpty)		
Configura la modalità di editing numerico della Keyboard. Consente di avere o meno zeri a sinistra del valore e può inserire automaticamente lo zero quando non c'e' testo nel campo di edit.		
param	allowLeftZero	true=consente di avere zeri a sinistra
	setZeroIsEmpty	true=inserimento dello zero automatico quando non esiste testo
return	1=successo, 0=errore	
call	sknKbrd.specmodes(true, false)	

sknKbrd.extratxt ()

Configura una label opzionale della Keyboard.

sknKbrd.extratxt (xpos, ypos, width, height, backColor, textColor, text, fontName)		
	Configura una label opzionale della Keyboard. Questo comando viene normalmente utilizzato per aggiungere una label all'aread di editing (ad esempio "Kwh" di finaco al valore da digitare)	
param	xpos	origine asse x
	ypos	origine asse y
	width	larghezza dell'area del testo
	height	altezza dell'area del testo
	backColor	colore di sfondo del testo (numerico o stringa per colore predefinito)
	textColor	colore del testo (numerico o stringa per colore predefinito)
	text	Testo fisso o string vuota
	fontName	nome del font del testo
return	1=successo, 0=errore	
call	sknKbrd.extratxt(100, 15, 120, 20,"black","white", "Kwh", "Arial14")	

sknKbrd.cursor ()

Configura il carattere che identifica il cursore della zona di editing.

sknKbrd.cursor (cursChar)		
	Permette di visualizzare un carattere indicante il cursore nell'area di editing. Una string vuota indica che il cursore non deve essere visualizzato	
param	cursChar	string contenente un solo carattere o "" per nessun cursore
return	1=successo, 0=errore	
call	sknKbrd.cursor("_")	

GRAPH

La Classe **sknGraph** implementa i grafici cartesiani.

Membro	Descrizione
Control	
sknGraph.create()	Crea un Graph
sknGraph.delete()	Distrugge un Graph
sknGraph.dataclear()	Elimina tutti i settings di un Graph
View	
sknGraph.show()	Visualizza un Graph
sknGraph.hide()	Nasconde un Graph
Proprietà	
sknGraph.reconfig()	Riconfigura un Graph
sknGraph.confaxis()	Configura gli assi di un Graph
sknGraph.axisopt()	Configura le componenti opzionali degli assi di un Graph
sknGraph.setscale()	Configura il range di un Graph
sknGraph.setimages()	Configura l'immagine di un Graph
sknGraph.aspect()	Modifica l'aspetto di un Graph
sknGraph.setdata()	Configura i dati di un Graph
sknGraph.ghostval()	Configura il valore fantasma di un Graph
Visibilità	
sknGraph.bringTop()	Porta il Graph in primo piano
sknGraph.bringBot()	Porta il Graph in sfondo
sknGraph.refresh()	Forza il refresh del Graph
Aggregazione	
sknGraph.auxData()	Controlla l'AuxData di un Graph
sknGraph.auxStr()	Controlla l'AuxString di un Graph
sknGraph.group()	Controlla l'appartenenza di un gruppo di un Graph
sknGraph.subGroup()	Controlla l'appartenenza di un sotto gruppo di un Graph
Auto Hide	
sknGraph.autoHide()	Controlla la funzionalità di auto hide di un Graph
sknGraph.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di un Graph
sknGraph.autoHideMode()	Controlla il tipo di hide per la funzionalità di auto hide di un Graph
Blink	
sknGraph.blinkEn()	Controlla la funzionalità di blink di un Graph

<code>sknGraph.blinkCustEn()</code>	Controlla la funzionalità di blink custom o globale di un Graph
Eventi	
<code>sknGraph.keyPressCB()</code>	Controlla il nome della funzione di callback di key press di un Graph
<code>sknGraph.keyReleaseCB()</code>	Controlla il nome della funzione di callback di key release di un Graph
Auto Hide callbacks	
<code>sknGraph.autoHideCBshow()</code>	Controlla la funzione di callback hide di un Graph
<code>sknGraph.autoHideCBhide()</code>	Controlla la funzione di callback show di un Graph
Animazione	
<code>sknGraph.aniEn()</code>	Controlla la funzionalità di animation di un Graph
<code>sknGraph.aniMode()</code>	Controlla la modalità di animation di un Graph
Child	
<code>sknGraph.autoFocusChild()</code>	Controlla la funzionalità di autofocus per i child Graph
<code>sknGraph.bringToTopChild()</code>	Controlla la funzionalità di brig to top per i child Graph
Callbacks	
<code>sknGraph.longTouchCB()</code>	Controlla il nome della funzione di callback per un long touch
<code>sknGraph.gestureCB()</code>	Controlla il nome della funzione di callback per una gesture

Classe [sknGraph]

sknGraph.create ()

Crea un Graph.

```
sknGraph.create ("graphName", "motherWin", xpos, ypos, xsize,
ysize, bkColor, sepColor, axisColor, textColor, "textFont",
spaceLeft, spaceBotom)
```

	Crea un Graph.	
param	graphName	nome del Graph
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza del Graph
	ysize	altezza del Graph
	bkColor	colore di sfondo (numerico o stringa per colore predefinito)
	sepColor	colore del separatore (numerico o stringa per colore predefinito)
	axisColor	colore degli assi (numerico o stringa per colore predefinito)
	textColor	colore del testo (numerico o stringa per colore predefinito)
	textFont	font del testo
	spaceLeft	spazio a sinistra del grafico
	spaceBotom	spazio sotto il grafico
return	1=successo, 0=errore	
call	sknGraph.create("gr", "winHome", 10, 10, 300, 150,"black","red","white", "Arial14", 30, 20)	

sknGraph.delete ()

Distrugge un Graph.

Questo metodo non rimuove gli oggetti figli della finestra.

```
sknGraph.delete ("graphName")
```

	Distrugge un Graph.	
param	graphName	Nome del Graph
return	1=successo, 0=errore	
call	sknGraph.delete("myGraph")	

sknGraph.reconfig ()

Riconfigura un Graph.

```
sknGraph.reconfig ("graphName", xpos, ypos, xsize, ysize, bkColor,  
sepColor, axisColor, textColor, "textFont" spaceLeft, spaceBotom)
```

	Riconfigura un Graph.	
param	graphName	nome del Graph
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza del Graph
	ysize	altezza del Graph
	bkColor	colore di sfondo (numerico o stringa per colore predefinito)
	sepColor	colore del separatore (numerico o stringa per colore predefinito)
	axisColor	colore degli assi (numerico o stringa per colore predefinito)
	textColor	colore del testo (numerico o stringa per colore predefinito)
	textFont	font del testo
	spaceLeft	spazio a sinistra del grafico
	spaceBotom	spazio sotto il grafico
return	1=successo, 0=errore	
call	sknGraph.reconfig("gr", 10, 10, 300, 150,"black","red","white", "Arial14", 30, 20)	

sknGraph.confaxis ()

Configura gli assi di un Graph.

```
sknGraph.confaxis ("graphName", XaxisDiv, XstepForDiv, XdivForTxt,  
XdecimalDigit, YaxisDiv, YstepForDiv, YdivForTxt, YdecimalDigit)
```

	Configura gli assi di un Graph.	
param	graphName	nome del Graph
	XaxisDiv	divisore per l'asse X
	XstepForDiv	numero di step per ogni divisione X
	XdivForTxt	numero di divisioni per ogni label dell'asse X
	XdecimalDigit	numero di decimali sull'asse X
	YaxisDiv	divisore per l'asse Y
	YstepForDiv	numero di step per ogni divisione Y
	YdivForTxt	numero di divisioni per ogni label dell'asse Y
	YdecimalDigit	numero di decimali sull'asse Y
return	1=successo, 0=errore	
call	sknGraph.confaxis("gr", 1,1,1,0,10,1,1,2)	

sknGraph.axisopt ()

Configura le componenti opzionali deli gli assi di un Graph.

```
sknGraph.axisopt ("graphName", barSpacing, spacyngColor,  
freeBarColor graduationEnable, xAxisTextEn, yAxisTextEn,  
XgridEnable, YgridEnable, XgridStyle, XgridStyle, YgridStyle,  
YgridStyle)
```

	Configura le componenti opzionali deli gli assi di un Graph.	
param	graphName	nome del Graph
	barSpacing	spaziatura fra le barre
	spacyngColor	colore dello spazio fra le barre (numerico o stringa per colore predefinito)
	freeBarColor	colore della barra free (numerico o stringa per colore predefinito)
	graduationEnable	true=abilita la graduazione
	xAxisTextEn	true=abilita le label di testo nell'asse X
	yAxisTextEn	true=abilita le label di testo nell'asse Y
	XgridEnable	true=abilita la griglia sull'asse X
	YgridEnable	true=abilita la griglia sull'asse Y
	XgridStyle	stile della griglia prima sezione sull'asse X {(-1 = compact style, 1...N = dashed style, pixels for fill and empty line)}
	XgridStyleOff	stile della griglia seconda sezione sull'asse X {(-1 = compact style, 1...N = dashed style, pixels for fill and empty line)}
	YgridStyle	stile della griglia prima sezione sull'asse Y {(-1 = compact style, 1...N = dashed style, pixels for fill and empty line)}
	YgridStyleOff	stile della griglia seconda sezione sull'asse Y {(-1 = compact style, 1...N = dashed style, pixels for fill and empty line)}
return	1=successo, 0=errore	
call	sknGraph.axisopt("gr", 10,"magenta","black", true, true, true, true,true, -1, -1, -1, -1)	

sknGraph.setScale ()

Configura il range di un Graph.

sknGraph.setScale ("graphName", YminValue, Y.MaxValue)		
	Configura il range di un Graph.	
param	graphName	nome del Graph
	Y minValue	valore minimo del grafico {-32000 ... +32000}
	Y maxValue	valore massimo del grafico {-32000 ... +32000}
return	1=successo, 0=errore	
call	sknGraph.setScale("gr", 1, 230)	

sknGraph.setImages ()

Configura l'immagine di un Graph.

sknGraph.setImages ("graphName", chartItem, "collection", "image")		
	Configura l'immagine di un Graph.	
param	graphName	nome del Graph
	chartItem	item (grafico) selezionato {1 ... N}
	collection	nome della collezione di immagini
	image	nome dell'immagine
return	1=successo, 0=errore	
call	sknGraph.image("gr", 1, "icons", "alert")	

sknGraph.aspect ()

Modifica l'aspetto di un Graph.

sknGraph.aspect ("graphName", graphItem, "grapType", graphColor, lineSize)		
---	--	--

	Modifica l'aspetto di un Graph.	
param	graphName	nome del Graph
	graphItem	item (grafico) selezionato {1 ... N}
	graphType	tipo di grafico {"bar"= barre verticali, "barstack" = barre verticali impilate (fra differenti items), "lineborder" = linea, "fullimages" = immagini sui vertici dei campioni}
	graphColor	colore del valore (numerico o stringa per colore predefinito)
	lineSize	numero di pixels della linea (valido per il tipo "lineborder")
return	1=successo, 0=errore	
call	sknGraph.aspect("gr", 1, "bar", "magenta", 1)	

sknGraph.setdata ()

Configura i dati di un Graph.

sknGraph.setdata ("graphName", graphItem, "DataSetPtr", numItems, "TextLabelsPtr")		
---	--	--

	Configura i dati di un Graph.	
param	graphName	nome del Graph
	graphItem	item (grafico) selezionato {1 ... N}
	DataSetPtr	nome della tabella che contiene i campioni (ex: dataT = {1,5,4,23,56,7,88})
	numItems	numero di samples rappresentabili dal grafico
	TextLabelsPtr	nome della tabella che contiene le labels dell'asse X (ex: labT = {"Gen", "Feb", "Mar"})
return	1=successo, 0=errore	
call	sknGraph.setdata("gr", 1, "dataT", 25, "labT")	

sknGraph.ghostval ()

Configura il valore fantasma di un Graph.

sknGraph.ghostval ("graphName", chartItem, enGhost, ghostValue)		
	Configura il valore fantasma di un Graph. Questo valore non viene visualizzato.	
param	graphName	nome del Graph
	chartItem	item (grafico) selezionato {1 ... N}
	enGhost	true=abilita la gestione del valore ghost
	ghostValue	valore da non visualizzare {-32000 ... +32000}
return	1=successo, 0=errore	
call	sknGraph.ghostval("gr", true, 1024)	

sknGraph.bringTop ()

Porta la finestra in primo piano.

sknGraph.bringTop ("graphName")		
	Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Graph attive.	
param	graphName	Nome del Graph
return	1=successo, 0=errore	
call	sknGraph.bringTop("myGraph")	

sknGraph.bringBot ()

Porta la finestra in sfondo.

sknGraph.bringBot ("graphName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	graphName	Nome del Graph
return	1=successo, 0=errore	
call	sknGraph.bringBot("myGraph")	

sknGraph.refresh ()

Forza il ridisegno di un Graph.

sknGraph.refresh ("graphName")	
	Forza il ridisegno di un Grafico. Questo metodo viene utilizzato quando il gestore delle finestre non è in grado di determinare un cambiamento di una finestra per eseguirne automaticamente il ridisegno, ad esempio il cambiamento di un flag che durante il ridisegno della finestra coinvolge degli artefatti grafici.
param	graphName Nome del Graph
return	1=successo, 0=errore
call	sknGraph.refresh("myGraph")

sknGraph.keyPressCB ()

Controlla il nome della funzione di callback per il key press del Graph.

sknGraph.keyPressCB ("graphName", "callback")	
	Controlla il nome della funzione di callback per il key press del Graph. Se non valorizzato il campo callback, viene ritornato il valore corrente.
param	graphName Nome del Graph
	Callback funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set
call	sknGraph.keyPressCB("myGraph", "keyPressWin") ... print ("\r\nmy winKeyPressCB=" .. sknGraph.keyPressCB ("myGraph"))

sknGraph.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release del Graph.

sknGraph.keyReleaseCB ("graphName", "callback")		
	Controlla il nome della funzione di callback per il key press del Graph. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	graphName	Nome del Graph
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknGraph.keyReleaseCB("myGraph", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknGraph.keyReleaseCB ("myGraph"))	

sknGraph.dataclear ()

Elimina tutti i settings di un Graph.

sknGraph.dataclear ("graphName")		
	Elimina tutti i settings di un Graph.	
param	graphName	nome del Graph
return	1=successo, 0=errore	
call	sknGraph.dataclear("gr")	

sknGraph.show ()

Visualizza un Graph.

sknGraph.show ("graphName")		
	Visualizza un Graph.	
param	graphName	nome del Graph
return	1=successo, 0=errore	
call	sknGraph.show("myGraph")	

sknGraph.hide ()

Nasconde un Graph.

sknGraph.hide ("graphName")	
	Nasconde un Graph.
param	graphName
	Nome del Graph
return	1=successo, 0=errore
call	sknGraph.hide("myGraph")

sknGraph.auxData ()

Controlla l'AuxData di un Graph.

sknGraph.auxData ("graphName", auxData)	
	Controlla l'AuxData di un Graph. L'AuxData viene utilizzato per identificare un Graph con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.
param	graphName
	Nome del Graph
	auxData
	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set
call	sknGraph.auxData("myGraph", 1234) ... print ("\r\nmy auxData=" .. sknGraph.auxData("myGraph"))

sknGraph.auxStr ()

Controlla l'AuxString di un Graph.

sknGraph.auxStr ("graphName", auxString)		
Controlla l'AuxString di un Graph. L'AuxString viene utilizzato per identificare un Graph con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	graphName	Nome del Graph
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	<pre>sknGraph.auxData("myGraph", "W1") ... print ("\r\nmy auxString=" .. sknGraph.auxStr("myGraph"))</pre>	

sknGraph.autoHide ()

Controlla la funzionalità di auto hide di un Graph.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknGraph.autoHide ("graphName", autoHide)		
Controlla la funzionalità di auto hide di un Graph. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).		
param	graphName	Nome del Graph
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	<pre>sknGraph.autoHide("myGraph", true) ... print ("\r\nmy autohide=" .. sknGraph.autoHide ("myGraph"))</pre>	

sknGraph.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di un Graph.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknGraph.autoHideTO ("graphName", autoHideTimeout)		
	Controlla il timeout per la funzionalità di auto hide di un Graph. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	graphName	Nome del Graph
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknGraph.autoHideTO("myGraph", 60) ... print ("\r\nmy autohideTO=" .. sknGraph.autoHideTO ("myGraph"))	

sknGraph.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di un Graph.

sknGraph.autoHideMode ("graphName", "mode")		
	Controlla il tipo ti hide per la funzionalità di auto hide di un Graph. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.	
param	graphName	Nome del Graph
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknGraph.autoHideMode("myGraph", "left") ... print ("\r\nmy autohideMode=" .. sknGraph.autoHideMode ("myGraph"))	

sknGraph.blinkEn ()

Controlla la funzionalità di blink di un Graph.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGraph.blinkEn ("graphName", blink)		
Controlla la funzionalità di blink di un Graph. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	graphName	Nome del Graph
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknGraph.blinkEn("myGraph", true) ... print ("\\r\\nmy blinkEn=" .. sknGraph.blinkEn ("myGraph"))	

sknGraph.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di un Graph.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGraph.blinkCustEn ("graphName", blink)		
Controlla la funzionalità di blink custom o globale di un Graph. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	graphName	Nome del Graph
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknGraph.blinkCustEn("myGraph", true) ... print ("\\r\\nmy blinkCustEn=" .. sknGraph.blinkCustEn ("myGraph"))	

sknGraph.autoHideCBshow ()

Controlla la funzione di callback show di un Graph.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGraph.autoHideCBshow ("graphName", "callback")

Controlla la funzione di callback show di un Graph.
E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento.
Se non valorizzato il campo callback, viene ritornato il valore corrente.

param graphName Nome del Graph

callback nome della funzione callback dell'evento (opt)

return callback o 1=successo, 0=errore per Set

call sknGraph.autoHideCBshow("myGraph", "myCBshow")

...
print ("\r\nmy autohideCBshow=" .. sknGraph.autoHideCBshow ("myGraph"))

sknGraph.autoHideCBhide ()

Controlla la funzione di callback hide di un Graph.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGraph.autoHideCBhide ("graphName", "callback")

Controlla la funzione di callback hide di un Graph.
E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento.
Se non valorizzato il campo callback, viene ritornato il valore corrente.

param graphName Nome del Graph

callback nome della funzione callback dell'evento (opt)

return callback o 1=successo, 0=errore per Set

call sknGraph.autoHideCBhide("myGraph", "myCBshow")

...
print ("\r\nmy autohideCBhide=" .. sknGraph.autoHideCBhide ("myGraph"))

sknGraph.aniEn ()

Controlla la funzionalità di animation di un Graph.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknGraph.aniEn ("graphName", animation)		
Controlla la funzionalità di animation di un Graph. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.		
param	graphName	Nome del Graph
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknGraph.aniEn("myGraph", true) ... print ("\\r\\nmy aniEn=" .. sknGraph.aniEn ("myGraph"))</pre>	

sknGraph.aniMode ()

Controlla la modalità di animation di un Graph.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknGraph.aniMode ("graphName", "aniMode")		
Controlla la modalità di animation di un Graph. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	graphName	Nome del Graph
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	<pre>sknGraph.aniMode("myGraph", "shift") ... print ("\\r\\nmy aniMode=" .. sknGraph.aniMode ("myGraph"))</pre>	

sknGraph.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknGraph.autoFocusChild ("graphName", autoFocus)

	Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.	
param	graphName	Nome del Graph
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknGraph.autoFocusChild("myGraph", true) ... print ("\r\nmy autoFocusChild=" .. sknGraph.autoFocusChild ("myGraph"))	

sknGraph.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknGraph.bringToTopChild ("graphName", en)

	Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.	
param	graphName	Nome del Graph
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknGraph.bringToTopChild("myGraph", true) ... print ("\r\nmy bringToTopChild=" .. sknGraph.bringToTopChild ("myGraph"))	

sknGraph.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknGraph.longTouchCB ("graphName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	graphName	Nome del Graph
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknGraph.longTouchCB("myGraph", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknGraph.longTouchCB ("myGraph"))	

sknGraph.group ()

Controlla l'appartenenza di un gruppo di un Graph.

sknGraph.group ("graphName", "grpName")		
Controlla l'appartenenza di un gruppo di un Graph. Il grpName viene utilizzato per associare un Graph con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	graphName	Nome del Graph
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknGraph.group("myGraph", "setGrp") ... print ("\r\nmy group=" .. sknGraph.group("myGraph"))	

sknGraph.subGroup ()

Controlla l'appartenenza di un sotto gruppo di un Graph.

sknGraph.subGroup ("graphName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di un Graph. Il subgrpName viene utilizzato per associare un Graph con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	graphName	Nome del Graph
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknGraph.subGroup("myGraph", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknGraph.subGroup ("myGraph"))	

sknGraph.gestureCB ()

Controlla il nome della funzione di callback per le gesture della window.

sknGraph.gestureCB ("graphName", "gesture", "callback")		
Controlla il nome della funzione di callback per le gesture del Graph. Se viene omesso il campo cb, viene ritornato il nome della funzione di callback relativa al campo gesture.		
param	graphName	Nome del graph
	gesture	tipo di gestione {"up", "down", "left", "right"}
	callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknGraph.gestureCB("myGraph", "up", "gestureUpCb") ... print ("\r\nmy gestureCB:up=" .. sknGraph.gestureCB ("home", "up"))	

GAUGE

La Classe **sknGauge** implementa la Gauge.

Membro	Descrizione
Control	
sknGauge.create()	Crea un Gauge
sknGauge.delete()	Distrugge un Gauge
View	
sknGauge.show()	Visualizza un Gauge
sknGauge.hide()	Nascondi un Gauge
Proprietà	
sknGauge.modegraphonly()	Configura l'aspetto di un Gauge
sknGauge.setValue()	Assegna i dati ad un Gauge
Visibilità	
sknGauge.bringTop()	Porta il Gauge in primo piano
sknGauge.bringBot()	Porta il Gauge in sfondo
sknGauge.refresh()	Forza il refresh del Gauge
Aggregazione	
sknGauge.auxData()	Controlla l'AuxData di un Gauge
sknGauge.auxStr()	Controlla l'AuxString di un Gauge
sknGauge.group()	Controlla l'appartenenza di un gruppo di un Gauge
sknGauge.subGroup()	Controlla l'appartenenza di un sotto gruppo di un Gauge
Auto Hide	
sknGauge.autoHide()	Controlla la funzionalità di auto hide di un Gauge
sknGauge.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di un Gauge
sknGauge.autoHideMode()	Controlla il tipo di hide per la funzionalità di auto hide di un Gauge
Blink	
sknGauge.blinkEn()	Controlla la funzionalità di blink di un Gauge
sknGauge.blinkCustEn()	Controlla la funzionalità di blink custom o globale di un Gauge
Eventi	
sknGauge.keyPressCB()	Controlla il nome della funzione di callback di key press di un Gauge
sknGauge.keyReleaseCB()	Controlla il nome della funzione di callback di key release di un Gauge
Auto Hide callbacks	
sknGauge.autoHideCBshow()	Controlla la funzione di callback hide di un Gauge

sknGauge.autoHideCBhide()	Controlla la funzione di callback show di un Gauge
Animazione	
sknGauge.aniEn()	Controlla la funzionalità di animation di un Gauge
sknGauge.aniMode()	Controlla la modalità di animation di un Gauge
Child	
sknGauge.autoFocusChild()	Controlla la funzionalità di autofocus per i child Gauge
sknGauge.bringToTopChild()	Controlla la funzionalità di brig to top per i child Gauge
Callbacks	
sknGauge.longTouchCB()	Controlla il nome della funzione di callback per un long touch

Classe [sknGauge]

sknGauge.create()

Crea una Gauge.

sknGauge.create("gaugeName", "motherWin", xpos, ypos, xszie, ysize)		
	Crea una Gauge.	
param	gaugeName	Nome della Gauge
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza della Gauge
	ysize	altezza della Gauge
return	1=successo, 0=errore	
call	sknGauge.create("pwrGauge", "winHome", 30, 50, 42, 42)	

sknGauge.delete()

Distrugge una Gauge.

Questo metodo non rimuove gli oggetti figli della finestra.

sknGauge.delete ("gaugeName")		
	Distrugge una Gauge.	
param	gaugeName	Nome della Gauge
return	1=successo, 0=errore	
call	sknGauge.delete("pwrGauge")	

sknGauge.modegraphonly()

Configura l'aspetto di un Gauge.

```
sknGraph.modegraphonly ("gaugeName", "Imagecoll", "backImage",
minValGauge, maxValGauge, bkColor, needleColor, needleCenter_x,
needleCenter_y, needleLenght, "needlePolipointsTbl",
needleNumOfPolPnts, startScaleAngle, endScaleAngle)
```

	Configura l'aspetto di un Gauge.	
param	gaugeName	nome della Gauge
	Imagecoll	Collezione di immagini
	backImage	immagine di sfondo del gauge
	minValGauge	valore minimo ammesso dal gauge
	maxValGauge	valore massimo ammesso dal gauge
	bkColor	colore di sfondo del gauge (numerico o stringa per colore predefinito)
	needleColor	colore dell'indicatore (numerico o stringa per colore predefinito)
	needleCenter_x	centro dell'indicatore asse X
	needleCenter_y	centro dell'indicatore asse Y
	needleLenght	lunghezza dell'indicatore
	needlePolPointsTbl	tabella che contiene il poligono dell'indicatore. formato: {x0, y0, x1, y1, x2, y2, xN, yN}
	needleNumOfPolPnts	numero di punti del poligono che disegna l'indicatore
	startScaleAngle	angolo dell'inizio scala
	endScaleAngle	angolo della fine scala
return	1=successo, 0=errore	
call	sknGauge.modegraphonly("gauge", "simg", "gaugelmg", 0,4000,"black", "red",100, 100, 35, "needleTable", 5, 225, 275)	

sknGauge.setvalue()

Assegna i dati ad un Gauge.

```
sknGraph.setvalue ("gaugeName", gaugeValue)
```

	Assegna i dati ad un Gauge.	
param	gaugeName	nome della Gauge
	gaugeValue	valore del gauge {-32000 to +32000}
return	1=successo, 0=errore	
call	sknGauge.setvalue("gauge", 1250)	

sknGauge.bringTop ()

Porta la finestra in primo piano.

sknGauge.bringTop ("gaugeName")		
	Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Gauge attive.	
param	gaugeName	Nome della Gauge
return	1=successo, 0=errore	
call	sknGauge.bringTop("pwrGauge")	

sknGauge.bringBot ()

Porta la finestra in sfondo.

sknGauge.bringBot ("gaugeName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	gaugeName	Nome della Gauge
return	1=successo, 0=errore	
call	sknGauge.bringBot("pwrGauge")	

sknGauge.refresh ()

Forza il ridisegno di una Gauge.

sknGauge.refresh ("gaugeName")		
	Forza il ridisegno di una gauge. Questo metodo viene utilizzato quando il gestore delle finestre non è in grado di determinare un cambiamento di una finestra per eseguirne automaticamente il ridisegno, ad esempio il cambiamento di un flag che durante il ridisegno della finestra coinvolge degli artefatti grafici.	
param	winName	Nome della Gauge
return	1=successo, 0=errore	
call	sknGauge.refresh("pwrGauge")	

sknGauge.show ()

Visualizza una Gauge.

sknGauge.show ("gaugeName")		
	Visualizza una Gauge.	
param	gaugeName	nome della Gauge
return	1=successo, 0=errore	
call	sknGauge.show("pwrGauge")	

sknGauge.hide ()

Nasconde una Gauge.

sknGauge.hide ("gaugeName")		
	Nasconde una Gauge.	
param	gaugeName	Nome della Gauge
return	1=successo, 0=errore	
call	sknGauge.hide("pwrGauge")	

sknGauge.auxData ()

Controlla l'AuxData di una Gauge.

sknGauge.auxData ("gaugeName", auxData)		
Controlla l'AuxData di una Gauge. L'AuxData viene utilizzato per identificare una Gauge con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.		
param	gaugeName	Nome della Gauge
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknGauge.auxData("pwrGauge", 1234) ... print ("\r\nmy auxData=" .. sknGauge.auxData("pwrGauge"))	

sknGauge.auxStr ()

Controlla l'AuxString di una Gauge.

sknGauge.auxStr ("gaugeName", auxString)		
Controlla l'AuxString di una Gauge. L'AuxString viene utilizzato per identificare una Gauge con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	gaugeName	Nome della Gauge
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknGauge.auxData("pwrGauge", "W1") ... print ("\r\nmy auxString=" .. sknGauge.auxStr("pwrGauge"))	

sknGauge.keyPressCB ()

Controlla il nome della funzione di callback per il key press della Gauge.

sknGauge.keyPressCB ("gaugeName", "callback")		
Controlla il nome della funzione di callback per il key press della Gauge. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknGauge.keyPressCB("pwrGauge", "keyPressWin") ... print ("\r\nmy winKeyPressCB=" .. sknGauge.keyPressCB ("pwrGauge"))	

sknGauge.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release della Gauge.

sknGauge.keyReleaseCB ("gaugeName", "callback")		
Controlla il nome della funzione di callback per il key release della Gauge. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknGauge.keyReleaseCB("pwrGauge", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknGauge.keyReleaseCB ("pwrGauge"))	

sknGauge.autoHide ()

Controlla la funzionalità di auto hide di una Gauge.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknGauge.autoHide ("gaugeName", autoHide)		
	Controlla la funzionalità di auto hide di una Gauge. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	gaugeName	Nome della Gauge
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknGauge.autoHide("pwrGauge", true) ... print ("\r\nmy autohide=" .. sknGauge.autoHide ("pwrGauge"))	

sknGauge.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di una Gauge.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknGauge.autoHideTO ("gaugeName", autoHideTimeout)		
	Controlla il timeout per la funzionalità di auto hide di una Gauge. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	gaugeName	Nome della Gauge
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknGauge.autoHideTO("pwrGauge", 60) ... print ("\r\nmy autohideTO=" .. sknGauge.autoHideTO ("pwrGauge"))	

sknGauge.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di una Gauge.

sknGauge.autoHideMode ("gaugeName", "mode")		
Controlla il tipo ti hide per la funzionalità di auto hide di una Gauge. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.		
param	gaugeName	Nome della Gauge
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknGauge.autoHideMode("pwrGauge", "left") ... print ("\r\nmy autohideMode=" .. sknGauge.autoHideMode ("pwrGauge"))	

sknGauge.blinkEn ()

Controlla la funzionalità di blink di una Gauge.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGauge.blinkEn ("gaugeName", blink)		
Controlla la funzionalità di blink di una Gauge. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknGauge.blinkEn("pwrGauge", true) ... print ("\r\nmy blinkEn=" .. sknGauge.blinkEn ("pwrGauge"))	

sknGauge.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di una Gauge.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGauge.blinkCustEn ("gaugeName", blink)		
Controlla la funzionalità di blink custom o globale di una Gauge. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknGauge.blinkCustEn("pwrGauge", true) ... print ("\r\nmy blinkCustEn=" .. sknGauge.blinkCustEn ("pwrGauge"))	

sknGauge.autoHideCBshow ()

Controlla la funzione di callback show di una Gauge.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGauge.autoHideCBshow ("gaugeName", "callback")		
Controlla la funzione di callback show di una Gauge. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknGauge.autoHideCBshow("pwrGauge", "myCBshow") ... print ("\r\nmy autohideCBshow=" .. sknGauge.autoHideCBshow ("pwrGauge"))	

sknGauge.autoHideCBhide ()

Controlla la funzione di callback hide di una Gauge.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknGauge.autoHideCBhide ("gaugeName", "callback")		
Controlla la funzione di callback hide di una Gauge. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknGauge.autoHideCBhide("pwrGauge", "mycCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknGauge.autoHideCBhide ("pwrGauge"))</pre>	

sknGauge.aniEn ()

Controlla la funzionalità di animation di una Gauge.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknGauge.aniEn ("gaugeName", animation)		
Controlla la funzionalità di animation di una Gauge. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknGauge.aniEn("pwrGauge", true) ... print ("\\r\\nmy aniEn=" .. sknGauge.aniEn ("pwrGauge"))</pre>	

sknGauge.anMode ()

Controlla la modalità di animation di una Gauge.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknGauge.anMode ("gaugeName", "aniMode")

Controlla la modalità di animation di una Gauge. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	sknGauge.anMode("pwrGauge", "shift") ... print ("\r\nmy aniMode=" .. sknGauge.anMode ("pwrGauge"))	

sknGauge.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknGauge.autoFocusChild ("gaugeName", autoFocus)

Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknGauge.autoFocusChild("pwrGauge", true) ... print ("\r\nmy autoFocusChild=" .. sknGauge.autoFocusChild ("pwrGauge"))	

sknGauge.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknGauge.bringToTopChild ("gaugeName", en)		
Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknGauge.bringToTopChild("pwrGauge", true) ... print ("\r\nmy bringToTopChild=" .. sknGauge.bringToTopChild ("pwrGauge"))	

sknGauge.group ()

Controlla l'appartenenza di un gruppo di una Gauge.

sknGauge.group ("gaugeName", "grpName")		
Controlla l'appartenenza di un gruppo di una Gauge. Il grpName viene utilizzato per associare una Gauge con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	gaugeName	Nome della Gauge
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknGauge.group("pwrGauge", "setGrp") ... print ("\r\nmy group=" .. sknGauge.group("pwrGauge"))	

sknGauge.subGroup ()

Controlla l'appartenenza di un sotto gruppo di una Gauge.

sknGauge.subGroup ("gaugeName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di una Gauge. Il subgrpName viene utilizzato per associare una Gauge con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	gaugeName	Nome della Gauge
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknGauge.subGroup("pwrGauge", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknGauge.subGroup ("pwrGauge"))	

sknGauge.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknGauge.longTouchCB ("gaugeName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	gaugeName	Nome della Gauge
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknGauge.longTouchCB("pwrGauge", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknGauge.longTouchCB ("pwrGauge"))	

PIE

La Classe **sknPie** gestisce i grafici a Torta.

Membro	Descrizione
Control	
sknPie.create()	Crea un Pie
sknPie.delete()	Distrugge un Pie
View	
sknPie.show()	Visualizza un Pie
sknPie.hide()	Nascondi un Pie
Proprietà	
sknPie.aspect()	Configura l'aspetto di un Pie
sknPie.setdata()	Assegna i dati ad un Pie
Visibilità	
sknPie.bringTop()	Porta il Pie in primo piano
sknPie.bringBot()	Porta il Pie in sfondo
sknPie.refresh()	Forza il refresh del Pie
Aggregazione	
sknPie.auxData()	Controlla l'AuxData di un Pie
sknPie.auxStr()	Controlla l'AuxString di un Pie
sknPie.group()	Controlla l'appartenenza di un gruppo di un Pie
sknPie.subGroup()	Controlla l'appartenenza di un sotto gruppo di un Pie
Auto Hide	
sknPie.autoHide()	Controlla la funzionalità di auto hide di un Pie
sknPie.autoHideTO()	Controlla il timeout per la funzionalità di auto hide di un Pie
sknPie.autoHideMode()	Controlla il tipo ti hide per la funzionalità di auto hide di un Pie
Blink	
sknPie.blinkEn()	Controlla la funzionalità di blink di un Pie
sknPie.blinkCustEn()	Controlla la funzionalità di blink custom o globale di un Pie
Eventi	
sknPie.keyPressCB()	Controlla il nome della funzione di callback di key press di un Pie
sknPie.keyReleaseCB()	Controlla il nome della funzione di callback di key release di un Pie
Auto Hide callbacks	
sknPie.autoHideCBshow()	Controlla la funzione di callback hide di un Pie
sknPie.autoHideCBhide()	Controlla la funzione di callback show di un Pie

Animazione	
sknPie.aniEn()	Controlla la funzionalità di animation di un Pie
sknPie.aniMode()	Controlla la modalità di animation di un Pie
Child	
sknPie.autoFocusChild()	Controlla la funzionalità di autofocus per i child Pie
sknPie.bringToTopChild()	Controlla la funzionalità di brig to top per i child Pie
Callbaks	
sknPie.longTouchCB()	Controlla il nome della funzione di callback per un long touch
sknPie.gestureCB()	Controlla il nome della funzione di callback per le gesture della Pie

Classe [sknPie]

sknPie.create()

Crea una Pie.

sknPie.create("pieName", "motherWin", xpos, ypos, xsize, ysize)		
	Crea una Pie.	
param	pieName	Nome della Pie
	motherWin	finestra madre
	xpos	origine asse x
	ypos	origine asse y
	xsize	larghezza della Pie
	ysize	altezza della Pie
return	1=successo, 0=errore	
call	sknPie.create("pwrPie", "winHome", 30, 50, 42, 42)	

sknPie.delete ()

Distrugge una Pie.

Questo metodo non rimuove gli oggetti figli della finestra.

sknPie.delete ("pieName")		
	Distrugge una Pie.	
param	pieName	Nome della Pie
return	1=successo, 0=errore	
call	sknPie.delete("pwrPie")	

sknPie.aspect ()

Configura l'aspetto di un Pie.

```
sknPie.aspect ("pieName", xCenter, yCenter, pieRadius, backColor,  
"legendaFont" legendaXstart, legendaYstart, legendaH, legendaWidth  
legendaTextColor)
```

	Configura l'aspetto di un Pie.	
param	pieName	nome della Pie
	xCenter	centro della pie asse X
	yCenter	centro della pie asse y
	pieRadius	raggio del pie
	backColor	colore di sfondo del pie (numerico o stringa per colore predefinito)
	legendaFont	font della legenda
	legendaXstart	start dell'are della legenda X
	legendaYstart	start dell'are della legenda Y
	legendaH	altezza della legenda
	legendaWidth	larghezza della legenda
	legendaTextColor	colore del testo della legenda (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknPie.aspect("pie", 100, 100, 70,"black", "Arial14", 300, 20, 100, 80, "red")	

sknPie.setdata ()

Assegna i dati ad un Pie.

sknPie.setdata ("pieName", maxVal, numSamples, "samplesTable")		
	Assegna i dati ad un Pie.	
param	pieName	nome della Pie
	maxVal	massimo valore del pie (il totale di tutti i campioni)
	numSamples	numero di campioni che compongono il pie
	samplesTable	nome della tabella che contiene i campioni. La tabella deve avere la seguente morfologia: {sample1, color1,"text1", sample2, color2, "text2", ...} dove i sample hanno rande -32000 to +32000
return	1=successo, 0=errore	
call	sknPie.setdata("pie", 100, 3, "pieTable")	

sknPie.auxData ()

Controlla l'AuxData di una Pie.

sknPie.auxData ("pieName", auxData)		
	Controlla l'AuxData di una Pie. L'AuxData viene utilizzato per identificare una Pie con un codice numerico impostato dall'applicazione. Se non viene impostato il valore di AuxData, la funzione ritorna quello corrente.	
param	pieName	Nome della Pie
	auxData	valore a 32 bit (opt)
return	auxData o 1=successo, 0=errore per Set	
call	sknPie.auxData("pwrPie", 1234) ... print ("\r\nmy auxData=" .. sknPie.auxData("pwrPie"))	

sknPie.auxStr ()

Controlla l'AuxString di una Pie.

sknPie.auxStr ("pieName", auxString)		
Controlla l'AuxString di una Pie. L'AuxString viene utilizzato per identificare una Pie con un codice alfanumerico impostato dall'applicazione. Se non viene impostato il valore di AuxString, la funzione ritorna quello corrente.		
param	pieName	Nome della Pie
	auxString	Stringa (opt)
return	auxString o 1=successo, 0=errore per Set	
call	sknPie.auxData("pwrPie", "W1") ... print ("\r\nmy auxString=" .. sknPie.auxStr("pwrPie"))	

sknPie.bringTop ()

Porta la finestra in primo piano.

sknPie.bringTop ("pieName")		
Porta la finestra in primo piano. Portare la finestra in primo piano consente la visualizzazione della stessa senza dover nascondere le altre Pie attive.		
param	pieName	Nome della Pie
return	1=successo, 0=errore	
call	sknPie.bringTop("pwrPie")	

sknPie.bringBot ()

Porta la finestra in sfondo.

sknPie.bringBot ("pieName")		
	Porta la finestra in sfondo. Portare la finestra in sfondo consente la portare la finestra corrente dietro quelle esistenti senza doverla nascondere.	
param	pieName	Nome della Pie
return	1=successo, 0=errore	
call	sknPie.bringBot("pwrPie")	

sknPie.refresh ()

Forza il ridisegno di una Pie.

sknPie.refresh ("pieName")		
	Forza il ridisegno di una Pie. Questo metodo viene utilizzato quando il gestore delle finestre non è in grado di determinare un cambiamento di una finestra per eseguirne automaticamente il ridisegno, ad esempio il cambiamento di un flag che durante il ridisegno della finestra coinvolge degli artefatti grafici.	
param	winName	Nome della Pie
return	1=successo, 0=errore	
call	sknPie.refresh("pwrPie")	

sknPie.keyPressCB ()

Controlla il nome della funzione di callback per il key press della Pie.

sknPie.keyPressCB ("pieName", "callback")		
	Controlla il nome della funzione di callback per il key press della Pie. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	pieName	Nome della Pie
	Callback	funzione di callback per key press (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknPie.keyPressCB("pwrPie", "keyPressWin") ... print ("\r\nmy winKeyPressCB=" .. sknPie.keyPressCB ("pwrPie"))	

sknPie.keyReleaseCB ()

Controlla il nome della funzione di callback per il key release della Pie.

sknPie.keyReleaseCB ("pieName", "callback")		
	Controlla il nome della funzione di callback per il key press della Pie. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	pieName	Nome della Pie
	Callback	funzione di callback per key release (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknPie.keyReleaseCB("pwrPie", "keyReleaseWin") ... print ("\r\nmy winKeyReleaseCB=" .. sknPie.keyReleaseCB ("pwrPie"))	

sknPie.show ()

Visualizza una Pie.

sknPie.show ("pieName")		
	Visualizza una Pie.	
param	pieName	nome della Pie
return	1=successo, 0=errore	
call	sknPie.show("pwrPie")	

sknPie.hide ()

Nasconde una Pie.

sknPie.hide ("pieName")		
	Nasconde una Pie.	
param	pieName	Nome della Pie
return	1=successo, 0=errore	
call	sknPie.hide("pwrPie")	

sknPie.autoHide ()

Controlla la funzionalità di auto hide di una Pie.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknPie.autoHide ("pieName", autoHide)		
	Controlla la funzionalità di auto hide di una Pie. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout).	
param	pieName	Nome della Pie
	autoHide	
return	autoHide o 1=successo, 0=errore per Set	
call	sknPie.autoHide("pwrPie", true) ... print ("\r\nmy autohide=" .. sknPie.autoHide ("pwrPie"))	

sknPie.autoHideTO ()

Controlla il timeout per la funzionalità di auto hide di una Pie.

Funzionalità condizionata dalla profilazione SKN_HIDDEN_OBJ_EN == 1

sknPie.autoHideTO ("pieName", autoHideTimeout)		
	Controlla il timeout per la funzionalità di auto hide di una Pie. L'auto hide consente di nascondere una finestra quando non viene rilevata attività utente (user timeout), questo parametro ne controlla il tempo che intercorre fra l'ultima attività dell'utente e la scomparsa della finestra.	
param	pieName	Nome della Pie
	autoHideTimeout	1-32000 secondi (opt)
return	Timeout o 1=successo, 0=errore per Set	
call	sknPie.autoHideTO("pwrPie", 60) ... print ("\r\nmy autohideTO=" .. sknPie.autoHideTO ("pwrPie"))	

sknPie.autoHideMode ()

Controlla il tipo ti hide per la funzionalità di auto hide di una Pie.

sknPie.autoHideMode ("pieName", "mode")		
Controlla il tipo ti hide per la funzionalità di auto hide di una Pie. Sono possibili differenti modalità di auto hide per ottenere differenti effetti visivi. Se non valorizzato il campo mode, viene ritornata la modalità corrente.		
param	pieName	Nome della Pie
	Mode	auto hide mode {"left", "right", "up", "down", "hide"} (opt)
return	mode o 1=successo, 0=errore per Set	
call	sknPie.autoHideMode("pwrPie", "left") ... print ("\r\nmy autohideMode=" .. sknPie.autoHideMode ("pwrPie"))	

sknPie.blinkEn ()

Controlla la funzionalità di blink di una Pie.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknPie.blinkEn ("pieName", blink)		
Controlla la funzionalità di blink di una Pie. Questa funzionalità permette di ottenere effetti di blink senza dover caricare la parte applicativa del timing e dello stato dei vari oggetti coinvolti. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	pieName	Nome della Pie
	blink	true=blink abilitato, false=blink disabilitato (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknPie.blinkEn("pwrPie", true) ... print ("\r\nmy blinkEn=" .. sknPie.blinkEn ("pwrPie"))	

sknPie.blinkCustEn ()

Controlla la funzionalità di blink custom o globale di una Pie.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknPie.blinkCustEn ("pieName", blink)		
Controlla la funzionalità di blink custom o globale di una Pie. E' possibile associare la funzionalità di blink a una configurazione globale o personalizzata. Se non valorizzato il campo blink, viene ritornato il valore corrente.		
param	pieName	Nome della Pie
	blink	true=blink custom, false=blink globale (opt)
return	blink o 1=successo, 0=errore per Set	
call	sknPie.blinkCustEn("pwrPie", true) ... print ("\r\nmy blinkCustEn=" .. sknPie.blinkCustEn ("pwrPie"))	

sknPie.autoHideCBshow ()

Controlla la funzione di callback show di una Pie.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknPie.autoHideCBshow ("pieName", "callback")		
Controlla la funzione di callback show di una Pie. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	pieName	Nome della Pie
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknPie.autoHideCBshow("pwrPie", "mycCBshow") ... print ("\r\nmy autohideCBshow=" .. sknPie.autoHideCBshow ("pwrPie"))	

sknPie.autoHideCBhide ()

Controlla la funzione di callback hide di una Pie.

Funzionalità condizionata dalla profilazione SKN_BLINK_OBJ_EN == 1

sknPie.autoHideCBhide ("pieName", "callback")

	Controlla la funzione di callback hide di una Pie. E' possibile associare l'evento di auto hide ad una funzione utente per poter aggiungere ulteriori funzionalità all'evento. Se non valorizzato il campo callback, viene ritornato il valore corrente.	
param	pieName	Nome della Pie
	callback	nome della funzione callback dell'evento (opt)
return	callback o 1=successo, 0=errore per Set	
call	<pre>sknPie.autoHideCBhide("pwrPie", "mycCBshow") ... print ("\\r\\nmy autohideCBhide=" .. sknPie.autoHideCBhide ("pwrPie"))</pre>	

sknPie.aniEn ()

Controlla la funzionalità di animation di una Pie.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknPie.aniEn ("pieName", animation)

	Controlla la funzionalità di animation di una Pie. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo animation, viene ritornato il valore corrente.	
param	pieName	Nome della Pie
	animation	true=animazione attiva (opt)
return	animation o 1=successo, 0=errore per Set	
call	<pre>sknPie.aniEn("pwrPie", true) ... print ("\\r\\nmy aniEn=" .. sknPie.aniEn ("pwrPie"))</pre>	

sknPie.anMode ()

Controlla la modalità di animation di una Pie.

Funzionalità condizionata dalla profilazione SKN_ANIMATION_ACTIVE== 1

sknPie.anMode ("Piepp", "aniMode")		
Controlla la modalità di animation di una Pie. Le animazioni permettono di creare particolari effetti visivi durante le fasi di show e hide della finestra. Se non valorizzato il campo aniMode, viene ritornato il valore corrente.		
param	pieName	Nome della Pie
	aniMode	modo di animazione (opt) "shift" = spostamento laterale "squeeze" = collassata "squeezecentral" = collassata centrale "global" = riflette la configurazione globale "none" = nessuna animazione
return	aniMode o 1=successo, 0=errore per Set	
call	sknPie.anMode("pwrPie", "shift") ... print ("\r\nmy aniMode=" .. sknPie.anMode ("pwrPie"))	

sknPie.autoFocusChild ()

Controlla la funzionalità di autofocus per le child window.

sknPie.autoFocusChild ("pieName", autoFocus)		
Controlla la funzionalità di autofocus per le child window. Se non valorizzato il campo autoFocus, viene ritornato il valore corrente.		
param	pieName	Nome della Pie
	autoFocus	true=child auto focus attiva (opt)
return	autoFocus o 1=successo, 0=errore per Set	
call	sknPie.autoFocusChild("pwrPie", true) ... print ("\r\nmy autoFocusChild=" .. sknPie.autoFocusChild ("pwrPie"))	

sknPie.bringToTopChild ()

Controlla la funzionalità di brig to top per le child window.

sknPie.bringToTopChild ("pieName", en)		
Controlla la funzionalità di brig to top per le child window. Se non valorizzato il campo en, viene ritornato il valore corrente.		
param	pieName	Nome della Pie
	En	true=bring to top attiva (opt)
return	en o 1=successo, 0=errore per Set	
call	sknPie.bringToTopChild("pwrPie", true) ... print ("\r\nmy bringToTopChild=" .. sknPie.bringToTopChild ("pwrPie"))	

sknPie.group ()

Controlla l'appartenenza di un gruppo di una Pie.

sknPie.group ("pieName", "grpName")		
Controlla l'appartenenza di un gruppo di una Pie. Il grpName viene utilizzato per associare una Pie con un gruppo [sottoGruppo] al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di grpName, la funzione ritorna quello corrente.		
param	pieName	Nome della Pie
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknPie.group("pwrPie", "setGrp") ... print ("\r\nmy group=" .. sknPie.group("pwrPie"))	

sknPie.subGroup ()

Controlla l'appartenenza di un sotto gruppo di una Pie.

sknPie.subGroup ("pieName", "grpName")		
Controlla l'appartenenza di un sotto gruppo di una Pie. Il subgrpName viene utilizzato per associare una Pie con un sotto gruppo al fine di poter eseguire delle azioni (show, hide, ...) di gruppo. Se non viene impostato il valore di subgrpName, la funzione ritorna quello corrente.		
param	pieName	Nome della Pie
	grpName	nome gruppo esistente (opt)
return	grpName o 1=successo, 0=errore per Set	
call	sknPie.subGroup("pwrPie", "mySubGrp") ... print ("\r\nmy subGrp=" .. sknPie.subGroup ("pwrPie"))	

sknPie.longTouchCB ()

Controlla il nome della funzione di callback per un long touch.

sknPie.longTouchCB ("pieName", "callback")		
Controlla il nome della funzione di callback per un long touch. Se non valorizzato il campo callback, viene ritornato il valore corrente.		
param	pieName	Nome della Pie
	callback	funzione di callback per un long touch (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknPie.longTouchCB("pwrPie", "LTcb") ... print ("\r\nmy longTouchCB=" .. sknPie.longTouchCB ("pwrPie"))	

sknPie.gestureCB ()

Controlla il nome della funzione di callback per le gesture della Pie.

sknPie.gestureCB ("PieName", "gesture", "callback")		
	Controlla il nome della funzione di callback per le gesture della Pie. Se viene omesso il campo cb, viene ritornato il nome della funzione di callback relativa al campo gesture.	
param	pieName	Nome della Pie
	gesture	tipo di gestione {"up", "down", "left", "right"}
	callback	funzione di callback per un ridisegno della finestra (opt)
return	callback o 1=successo, 0=errore per Set	
call	sknPie.gestureCB("myPie", "up", "gestureUpCb") ... print ("\r\nmy gestureCB:up=" .. sknPie.gestureCB ("myPie", "up"))	

DRAW (NON ATTIVO)

La Classe **sknDraw** implementa le funzionalità di base per il drawing, l'Anti Aliasing e i Memory Devices.

Membro	Descrizione
Draw	
sknDraw.text()	Visualizza Testo
sknDraw.image()	Visualizza immagine
sknDraw.line()	Visualizza Linea
sknDraw.rect()	Visualizza Rettangolo
sknDraw.fillRect()	Visualizza Rettangolo pieno
sknDraw.circle()	Visualizza Cerchio
sknDraw.fillCircle()	Visualizza Cerchio pieno
sknDraw.arc()	Visualizza Arco
sknDraw.clear()	Cancella lo screen
sknDraw.fonty()	Determina l'altezza del di un Font
sknDraw.imageFromFile()	Visualizza immagine proveniente da File
Draw Anti Aliasing	
sknDraw.AAline()	Visualizza Linea con Anti Aliasing
sknDraw.AAfillcircle()	Visualizza Cerchio pieno con Anti Aliasing
sknDraw.AAarc()	Visualizza Arco con Anti Aliasing
Memory Devices	
sknDraw.memdev_create()	Crea un Memory Device
sknDraw.memdev_delete()	Distruggi un Memory Device
sknDraw.memdev_clear()	Cancella l'area di un Memory Device
sknDraw.memdev_select()	Seleziona un Memory Device
sknDraw.memdev_cpy2memdev()	Esegui la copia nel memory device
sknDraw.memdev_cpy2lcd()	Esegui la copia del memory device nello screen
sknDraw.memdev_cpy2lcdAA()	Esegui la copia del memory device nello screen con Anti Aliasing

Classe [sknDraw]

sknDraw.text ()

Visualizza Testo.

sknDraw.text (xpos, ypos, xsize, ysize, color, BGcolor, "font", "text", "textAlign", "wrapMode")		
	Visualizza Testo.	
param	xpos	Posizione di start x
	ypos	Posizione di start y
	xsize	Dimensione orizzontale del testo
	ysize	Dimensione verticale del testo
	color	Colore del Testo (numerico o stringa per colore predefinito)
	BGcolor	Colore di sfondo (numerico o stringa per colore predefinito)
	font	nome del font
	text	testo da visualizzare
	textAlign	allineamento {"HorLeft_Vert_Top", "HorLeft_VertBot", "HorLeft_VertCenter", "HorRight_Vert_Top", "HorRight_VertBot", "HorRight_VertCenter", "HorCenter_Vert_Top", "HorCenter_VertBot", "HorCenter_VertCenter"}
	wrapMode	wrap mode {"none", "word", "char"}
return	1=successo, 0=errore	
call	sknDraw.text (15, 50, 120, 45,"white","green", "Arial16", "Draw Text Direct", "textAlign", "word")	

sknDraw.image ()

Visualizza immagine.

sknDraw.image (xpos, ypos, "imgColl", "image")		
	Visualizza immagine.	
param	xpos	Origine x
	ypos	Origine y
	imgColl	Nome della collezione di immagini
	image	nome della immagine
return	1=successo, 0=errore	
call	sknDraw.image (100, 75, "Icons", "Warning")	

sknDraw.line ()

Visualizza Linea.

sknDraw.line (x0, y0, x1, y1, color, pensize)		
	Visualizza Linea.	
param	x0	Start sinistra x
	y0	Start sinistra y
	x1	End destra x
	y1	End destra y
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.line (10, 50, 110, 90,"white", 2)	

sknDraw.rect ()

Visualizza Rettangolo.

sknDraw.rect (x0, y0, x1, y1, color, pensize)		
	Visualizza Rettangolo.	
param	x0	Start sinistra Top x
	y0	Start sinistra Top y
	x1	End destra Bot x
	y1	End destra Bot y
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.rect (10, 50, 110, 90, "white", 1)	

sknDraw.fillRect ()

Visualizza Rettangolo pieno.

sknDraw.fillRect (x0, y0, x1, y1, color)

	Visualizza Rettangolo pieno.	
param	x0	Start sinistra Top x
	y0	Start sinistra Top y
	x1	End destra Bot x
	y1	End destra Bot y
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.fillRect (10, 50, 110, 90,"white", 2)	

sknDraw.circle ()

Visualizza Cerchio.

sknDraw.circle (x0, y0, radius, color, pensize)

	Visualizza Cerchio.	
param	x0	Centro x
	y0	Centro y
	radius	Raggio in pixels
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.circle (10, 50, 34,"white", 4)	

sknDraw.fillcircle ()

Visualizza Cerchio pieno.

sknDraw.fillcircle (x0, y0, radius, color, pensize)		
	Visualizza Cerchio pieno.	
param	x0	Centro x
	y0	Centro y
	radius	Raggio in pixels
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.fillcircle (10, 50, 34,"white", 4)	

sknDraw.arc ()

Visualizza Arco.

sknDraw.arc (x0, y0, Xradius, Yradius, startAngle, endAngle, color, pensize)		
	Visualizza Arco.	
param	x0	Centro x
	y0	Centro y
	Xradius	Raggio x
	Yradius	Raggio y
	startAngle	Angolo di inizio
	endAngle	Angolo di fine
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.arc (100, 100, 70, 50, 34, 75,"red", 2)	

sknDraw.clear()

Cancella lo screen.

sknDraw.clear(color)		
	Cancella lo screen.	
param	color	Colore (numerico o stringa per colore predefinito)
return	1=successo, 0=errore	
call	sknDraw.clear ("black")	

sknDraw.fonty ()

Determina l'altezza del di un Font.

sknDraw.fonty ("fontName")		
	Determina l'altezza del di un Font.	
param	fontName	Nome del font
return	Dimensione y del font selezionato	
call	print ("Font size=", sknDraw.fonty ("Arial18"))	

sknDraw.imageFromFile ()

Visualizza immagine proveniente da File.

sknDraw.imageFromFile ("filename", xpos, ypos, [xCoordMax, yCoordMax])		
	Visualizza immagine proveniente da File. Valorizzando i campi xCoordMax e yCoordMax si può eseguire un ritaglio dell'area visualizzata.	
param	filename	Nome del file (completo di path)
	xpos	Corner sinistro Top x
	ypos	Corner sinistro Top y
	xCoordMax	Massima coordinata x (opt)
	yCoorMax	Massima coordinata y (opt)
return	1=successo, 0=errore	
call	sknDraw.imageFromFile ("mmc:\\Images\\FrogFries.png", 100, 100, 150, 150)	

sknDraw.AAline ()

Visualizza Linea con Anti Aliasing.

sknDraw.AAline (x0, y0, x1, y1, color, pensize)		
	Visualizza Linea con Anti Aliasing.	
param	x0	Start sinistra x
	y0	Start sinistra y
	x1	End destra x
	y1	End destra y
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.AAline (10, 50, 110, 90,"white", 2)	

sknDraw-AAfillcircle ()

Visualizza Cerchio pieno con Anti Aliasing.

sknDraw-AAfillcircle (x0, y0, radius, color, pensize)		
	Visualizza Cerchio pieno con Anti Aliasing.	
param	x0	Centro x
	y0	Centro y
	radius	Raggio in pixels
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw-AAfillcircle (10, 50, 34,"white", 4)	

sknDraw.AAarc ()

Visualizza Arco con Anti Aliasing.

sknDraw.AAarc (x0, y0, Xradius, Yradius, startAngle, endAngle, color, pensize)

	Visualizza Arco con Anti Aliasing.	
param	x0	Centro x
	y0	Centro y
	Xradius	Raggio x
	Yradius	Raggio y
	startAngle	Angolo di inizio
	endAngle	Angolo di fine
	color	Colore (numerico o stringa per colore predefinito)
	pensize	Dimensione del tratto
return	1=successo, 0=errore	
call	sknDraw.AAarc (100, 100, 70, 50, 34, 75,"red", 2)	

sknDraw.memdev_create ()

Crea un Memory Device.

sknDraw.memdev_create (xpos, ypos, xszie, ysize)

	Crea un Memory Device.	
param	xpos	Start sinistra x
	ypos	Start sinistra y
	xszie	Dimensione x
	ysize	Dimensione y
return	memory device handle o 0 se errore	
call	memDevHnd = sknDraw.memdev_create (10, 10, 100, 80) if memDevHnd == 0 then print ("Memory Device error") end	

sknDraw.memdev_delete ()

Distruggi un Memory Device.

sknDraw.memdev_delete (memDevHnd)		
	Distruggi un Memory Device.	
param	memDevHnd	Memory Device Handle
return	1=successo, 0=errore	
call	sknDraw.memdev_delete (memDevHnd)	

sknDraw.memdev_clear ()

Cancella l'area di un Memory Device.

sknDraw.memdev_clear (memDevHnd)		
	Cancella l'area di un Memory Device.	
param	memDevHnd	Memory Device Handle
return	1=successo, 0=errore	
call	sknDraw.memdev_clear (memDevHnd)	

sknDraw.memdev_select ()

Seleziona un Memory Device.

sknDraw.memdev_select (memDevHnd)		
	Seleziona un Memory Device.	
param	memDevHnd	Memory Device Handle
return	1=successo, 0=errore	
call	sknDraw.memdev_select (memDevHnd)	

sknDraw.memdev_cpy2memdev ()

Esegui la copia nel memory device.

sknDraw.memdev_cpy2memdev (memDevHnd)		
	Esegui la copia nel memory device.	
param	memDevHnd	Memory Device Handle
return	1=successo, 0=errore	
call	sknDraw.memdev_cpy2memdev (memDevHnd)	

sknDraw.memdev_cpy2lcd ()

Esegui la copia del memory device nello screen.

sknDraw.memdev_cpy2lcd (memDevHnd)		
	Esegui la copia del memory device nello screen.	
param	memDevHnd	Memory Device Handle
return	1=successo, 0=errore	
call	sknDraw.memdev_cpy2lcd (memDevHnd)	

sknDraw.memdev_cpy2lcdAA ()

Esegui la copia del memory device nello screen con Anti Aliasing.

sknDraw.memdev_cpy2lcdAA (memDevHnd)		
	Esegui la copia del memory device nello screen con Anti Aliasing.	
param	memDevHnd	Memory Device Handle
return	1=successo, 0=errore	
call	sknDraw.memdev_cpy2lcdAA (memDevHnd)	

DEBUG

La Classe **sknDbg** consente di eseguire semplici operazioni di debug e indagine sugli oggetti e le loro proprietà, mentre per alcune analisi di specifici oggetti (ad esempio una collezione di immagini) sono implementate direttamente nella Classe relativa.

Membro	Descrizione
<code>sknDbg.objListAll()</code>	Display del riassunto degli oggetti
<code>sknDbg.winList()</code>	Display della lista delle Window
<code>sknDbg.winChain ("winName")</code>	Display del chain delle Window
<code>sknDbg.objAnalysys ("objName")</code>	Display analisi di un oggetto
<code>sknDbg.cfgOutput (showCommonProp, showLayout, showDetails, showInternals, showCB, showDeepLayout, showGraphTree)</code>	Configura l'output del debug

Classe [sknDbg]

sknDbg.objListAll()

Esegue il display del riassunto degli oggetti

class.objListAll()

Display report utilizzo risorse

param

return 1=successo, 0=errore

call sknDbg.objLstAll()

Esempio di Output del comando da console:

```
>sknDbg.objListAll()
===== objects ===== Total : 500
- window      : 7
- iButton      : 18
- Button       : 0
- Label        : 41
- Image         : 23
- CheckBox     : 0
- Graph         : 0
- Home          : 0
- Kbr Alpha    : 0
- Kbr Num      : 0
- GraphBar     : 1
- Gauge         : 0
- Vmessage      : 0
- Rotary        : 0

- Free          : 405
- Unknown       : 5

===== Fonts ===== Total : 100
- Used          : 3

===== Image collections ===== Total : 10
- Used          : 6

===== Dictionary collections == Total : 10
- Used          : 4

===== Groups name ===== Total : 50
- Used          : 0

===== Multi Image collections ===== Total : 30
- Used          : 0
```

1

sknDbg.winList()

Visualizza l'elenco delle Window

sknDbg.winList()	
	Display elenco delle Window
param	
return	1=successo, 0=errore
call	sknDbg.winList()

Esempio di Output del comando da console:

```
> sknDbg.winList()

Windows list:

- winHome
- winCfg
- winGtemp
- winHL1
- winHL2
- winMeteo
- wintset

1
>
```

sknDbg.winChain()

Visualizza il collegamento fra finestre

sknDbg.winChain(winName)	
	Display Window chain
param	“winName”
	Nome della finestra madre
return	1=successo, 0=errore
call	sknDbg.winChain(“myWin”)

Esempio di Output del comando da console:

```
> sknDbg.winChain("wintset")

Object wintset details:
- [Image]: wintsetBgimage

1
>
```

sknDbg.objAnalysys ()

Visualizza l'analisi di un oggetto e i relativi oggetti in esso contenuti.

Questa funzionalità consente di identificare gli oggetti attivi in uno specifico istante, identificarne il nome, posizione e principali settings.

sknDbg.objAnalysys ("objName")

	Visualizza l'analisi di uno specifico oggetto
param	"objName"
	Nome dell'oggetto da analizzare
return	1=successo, 0=errore
call	sknDbg.objAnalysys ("objName")

Esempio di Output del comando da console:

```
> sknDbg.objAnalysys("winTset")

===== [Window : winTset]
Index      = 71
Pos        = 0, 0
Size       = 480, 272
Visible    = NO

===== [Image : winTsetBgimage]
Index      = 72
Pos        = 0, 0
Size       = 480, 272
Visible    = YES

===== [iButton : tsetZplus]
Index      = 73
Pos        = 106, 0
Size       = 54, 54
Event Key release      : win_tempSet._ZMinus
Visible    = YES
Unpress image = [btn] bgArrow
Press image   = [btnP] bgArrow

===== [iButton : tset_back]
Index      = 97
Pos        = 379, 220
Size       = 102, 54
Event Key release      : win_tempSet._editBack
Visible    = YES
Unpress image = [btn] btnRectA
Press image   = [btnP] btnRectA
1>
```

sknDbg.cfgOutput()

Configura l'output generato dalle funzioni di debug.

sknDbg.cfgOutput (showCommonProp, showLayout, showDetails, showInternals, showCB, showDeepLayout, showGraphTree)

	Configura l'output delle funzioni di debug con contenuti e profondità di dettaglio	
param	showCommonProp	true=visualizza le proprietà comuni
	showLayout	true=visualizza il layout
	showDetails	true=visualizza i dettagli
	showInternals	true=visualizza gli internals
	showCB	true=visualizza le Callbacks
	showDeepLayout	true=visualizza il layout di dettaglio
	showGraphTree	true=visualizza il grafico ad albero
return	1=successo, 0=errore	
call	sknDbg.cfgOutput(true, true, true, true, true, true, true)	

CLASSNAME TEMPLATE

La Classe skn??? dddd.

Membro	Descrizione
block	
xxx ()	Ddddddd

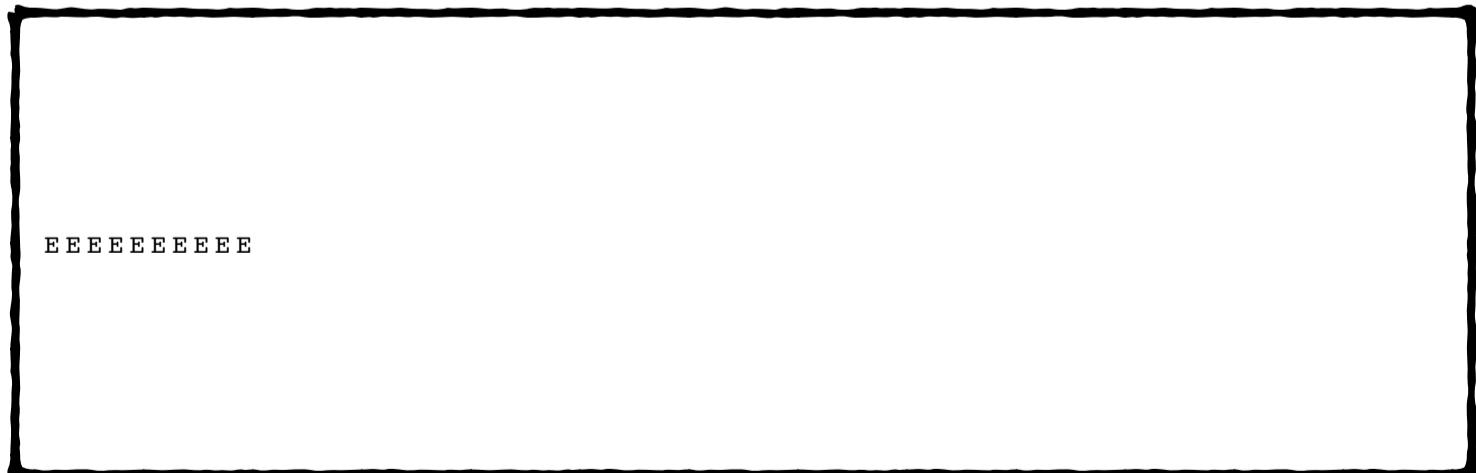
Classe [sknXXXX]

sknSys.

Descrizione del metodo

sknSys.	
	DDDDDD
param	
return	1=successo, 0=errore
call	sknSys.

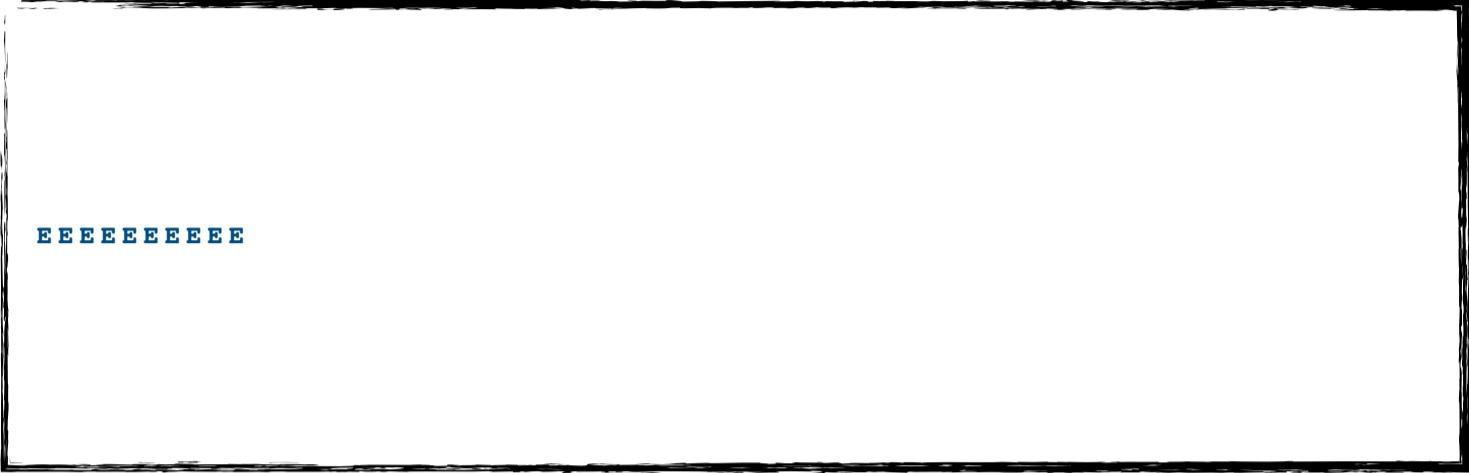
Code example:



NOTE:

nonenotenote

Esempio di Output del comando da console:



EEEEEEEEE