DVD-lab

DVD authoring for creative people

MEDIA CHANCE

● download ● buy now ● search

DVDlab
- Home
- Screenshots
- Encoder
- Banners

DVDlab PRO
- Home
- Screenshots
- History

DVD-9 DL
- Home

Resources
- Tips & Tricks
- Help
- Help
- Help
- Help
- Help
- FAQ
- History

DVD Tools
- Timecode calc
- Re-Aspect

Articles
- H. Theater
- DOF Machine
- 3D Video

Photo-Brush
- Start here

Real-Draw
- Start here

CompactDraw
- Start here

PhotoSEAM
- Start here

Multimedia Builder
- Start here

Other tools
- UltraSnap
- Camera Tools

Keypad Lock part 3
Create a cool keypad-like access to your secret pages. This is part 3 with more runtime script.

This article was written and published in December 2003, the second version of keypad script has been published in 4th of January 2004

Note: this text refer to a PRO version which is currently in beta.
A features presented here may not yet be available in the public beta version

A scripting support is added into **1.4 beta 2**

In the Previous Part 1 we learned how to create the secret keypad lock and then we learned how to export it to a Component.
In the Part 2 we addded a little bit of script to component so we can each time enter our own code to create the keypad lock.

Since this is such a nice example, we will demonstrate more advenced scripting technique in this Part 3.

**WARNING**, this below is a hardcore programming! You have the last chance to hit Back button now!

The Part 2 worked well, but we could have only kepadlock with 3 digits. In order to make a keypad lock each time with different number of digits we need to use the script much, much more.

The trick is to have just one menu and then create with script the other menus and links as we need them. So basically all is done in script, we just use one menu as our template that we can copy and modify in script. Cool.
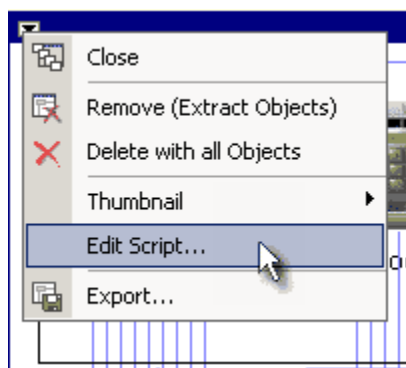
The menu is same as we build in Part 1, background image and 9 "Invisible normal" squares on top of the "buttons" (remember the buttons are in fact drawn on background).

This time we will have just one in our component. All other will be derived from this in script:
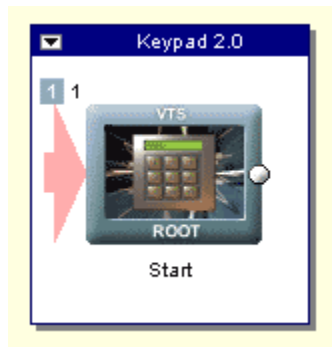
Without much talk, let's jump directly to script. Open the Edit Script from the Component menu.

Let's dig in.

First pahse - we need to get the secret code so we know how many menus we need to create. That's easy using input.

```
nCode=123
input ""Secret Access Code",nCode
// terminate script if pressed cancel
```

```
if bCancelInput then
end
endif
```

Note that **I** boldly added way to terminate script if user press Cancel on the input dialog.

We got our number but then we need to somehow break it into each digits. We will store these digits in array codes[ ]

```
nNum = 1
nTempCode = nCode

// note the label 10 - this is start of our "loop"
10 nNewCode = INT(nTempCode/10)
codes[nNum] = nTempCode-nNewCode*10
trace codes[nNum]

nNum=nNum+1
nTempCode = nNewCode

// still something left - loop to label 10
if (nNewCode>0) then
    goto 10
endif
```

That above is a bit scary but it makes sense. In the line
10 nNewCode = INT(nTempCode/10)
we have 10 at the beginning of line. This is called a **label**. later we can use **goto** to this label such as "goto 10".
This line and the line below simply feed the codes [ ] array with the digits, one by one in the loop, but backwards from right to left. The trick used is that we divide the number by ten, then make it integer so we will loose everything after decimal point and then subtract the original with 10 fold of this. Messy, but it is a simple math and here is real example:

nCode = **123**

123/10 = 12.3, INT of it is **12** (nNewCode)
123 - **12***10 = 123 - 120 = **3** <- first digit from right

12/10 = 1.2, INT is **1** (nNewCode)
12 - **1***10 = 12 - 10 = **2** <-second digit from right

1/10 = 0.1 , INT is 0 (nNewCode)
1 - 0*10 = 1 - 0 = **1** <- third digit from right

After all these steps we don't need to loop (goto 10) anymore since nNewCode is now zero. Our array **codes** looks like this 3,2,1. Good, but reversed - we just need to remember that.
If you can't digest the lines above please note the program below is going

more complicated than that so maybe it is time for cofee break?

The number of digits after this goto loop is obviously :

```
nNumberOfDigits = nNum-1
```

We need to create the first BAD menu and copy it from our first existing menu (template). As we learned in part 2 to address menus in component we use menusInBlackBox[menu in component]. Therefore menusInBlackBox [1] returns the first menu in the component. If we just simply use number 1 then this would mean a first menu in whole project which may not be same as first menu in component.

```
// create a BAD 1
menu //*****************************************
menu = MenuAdd(FALSE,"BAD 1",FALSE)
// copy from the first in black box to the new created menu
MenuCopy(menusInBlackBox[1], menu)
// now link all objects from menusInBlackBox[1] to menu
```

Now link all object from Start menu to the BAD 1 menu - we just setting all buttons to bad, later we will fix this

```
nOb = ObjectGetCount(menusInBlackBox[1]) // how many objects in the menu

// link all from start to bad 1
for x=1 to nOb
    ObjectLinkToMenu(menusInBlackBox[1],x,menu)
next x
```

Now let add a star text in the "display" of BAD 1 menu so user know he pressed first button, for more info see comments in green

```
// lets define some size of font and position of the new text

fontSize = 70
fontFace = "Arial"
fontX = 317
fontY = 100
stars = "*"

// Adding a new object in the BAD menu
object = ObjectAdd(menu,3,RGB(0,0,0),stars)
ObjectSetPos(menu,object,fontX,fontY)
ObjectSetFont(menu,object,fontFace,fontSize)
```

```
ObjectSetShadow(menu,object,0,0)

//Move the menu box in Connection window close to our start menu and
group it inside the Component
MenuPlaceNear(menu, menusInBlackBox[1], 0, 4)
MenuGroupWith(menu, menusInBlackBox[1],0)
```

It gets even messier, so far we created just one bad menu, we need to create more good and bad menus in a loop! It is basically the same idea as previously, but now in loop and we will also add the links. Please see the comments in code for more info - it is basically repeating what we learned before.

```
// remeber the last good and bad menu, the prevbad menu is the one we
created just few lines above
prevbad = menu
prevgood = menusInBlackBox[1]
// counter for codes array
codecounter = nNumberOfDigits

// now we need to create nNumberOfDigits-1 menu pairs
//***************************************************
for y=1 to nNumberOfDigits-1

  name = "GOOD "+CHR(48+y)
  menuGood = MenuAdd(FALSE,name,FALSE)
  MenuCopy(menusInBlackBox[1], menuGood)
  // Place the box right of previous good and add it to the component
  MenuPlaceNear(menuGood, prevgood, 0, 2)
  MenuGroupWith(menuGood, prevgood,0)

  name = "BAD "+CHR(48+y+1)
  menuBad = MenuAdd(FALSE,name,FALSE)
  MenuCopy(menusInBlackBox[1], menuBad)
  // Place the box right of previous bad and add it to the component
  MenuPlaceNear(menuBad, prevbad, 0, 2)
  MenuGroupWith(menuBad, prevbad,0)

  // now link this good menu and previous bad menu to the bad menu
  for x=1 to nOb
    ObjectLinkToMenu(menuGood,x,menuBad)
    ObjectLinkToMenu(prevbad,x,menuBad)
  next x

  //link the object corresponding to the secret digit from last good menu to
this good menu
  ObjectLinkToMenu(prevgood,codes[codecounter],menuGood)

  // add the star so users see feedback how many keys he enterred
```

```
    // we defined the stars before loop
    trace stars
    // add text object
    object = ObjectAdd(menuGood,3,RGB(0,0,0),stars)
    ObjectSetPos(menuGood,object,fontX,fontY)
    ObjectSetFont(menuGood,object,fontFace,fontSize)
    ObjectSetShadow(menuGood,object,0,0)

    // the bad has one star more
    stars=stars+"*"
    object = ObjectAdd(menuBad,3,RGB(0,0,0),stars)
    ObjectSetPos(menuBad,object,fontX,fontY)
    ObjectSetFont(menuBad,object,fontFace,fontSize)
    ObjectSetShadow(menuBad,object,0,0)


    // because the array is in opposite order
    codecounter = codecounter-1

    // now remember the previous bad and good
    prevbad = menuBad
    prevgood = menuGood

next y

// One more BAD menu -> BAD end
// ********************************
// the last bad menu doesn't link to anything - it is the dead end ACCESS
DENIED
for x=1 to nOb
  // since the ObjectDelete change the object order
  // we need to delete just the first object few times
  ObjectDelete(prevbad,1)
next x

// delete also the stars text, we will put access denied
ObjectDelete(prevbad,1)
object = ObjectAdd(prevbad,3,RGB(0,0,0),"WRONG CODE")
ObjectSetPos(prevbad,object,fontX,fontY+5)
ObjectSetFont(prevbad,object,fontFace,fontSize/3)
ObjectSetShadow(prevbad,object,0,0)

// link it to the start
MenuEndLink(prevbad, menusInBlackBox[1])
// set timer to 1 sec before we go to the start
MenuSetPBC(prevbad,1,0,0)

// GOOD END
```

```
*************************************************
// this is the last menu
menuGood = MenuAdd(FALSE,"GOOD END",FALSE)
MenuCopy(menusInBlackBox[1], menuGood)
// right of last good menu
MenuPlaceNear(menuGood, prevgood, 0, 2)
// put it inside the Component
MenuGroupWith(menuGood, prevgood,0)

// delete all objects
for x=1 to nOb
   // since the ObjectDelete change the object order
   // we need to delete just the first object few times
   ObjectDelete(menuGood,1)
next x

ObjectLinkToMenu(prevgood,codes[codecounter],menuGood)

// it should be 1, since this is the last digit = first in the array)
trace "Last Digit ",codecounter

// put a ACCESS OK text
object = ObjectAdd(menuGood,3,RGB(0,0,0),"ACCESS OK")
ObjectSetPos(menuGood,object,fontX,fontY+5)
ObjectSetFont(menuGood,object,fontFace,fontSize/3)
ObjectSetShadow(menuGood,object,0,0)

// set 2 sec timeout
MenuSetPBC(menuGood,2,0,0)

// set this menu to be an output object in connection
MenuSetComponent(menuGood, FALSE, TRUE)

// now you need to connect the GOOD END with any of your menu
```

Auch!

Now there are few possibilities what just happened after you read all this:

1. **You have no slightest idea what was it about. Nothing. Zero.**
   Don't worry, that's fine. You don't have to use LAB-talk at all, it is not necessary for DVD creation. How is the weather?
2. **You have some idea how it works in general, but you would need more time to look at each command**
   That's great, you are ready candidate to learn and use LAB-talk script!
3. **You know exactly what I was talking about and you see the whole structure clearly in front of you**
   You are most likely smoking illegal substance

The whole listing is <u>here</u> or the component is <u>here</u> (unzip it, and add the two files to DVDlabPRO folder \Extras\Components\)