

```
#contentWrapper #fs, #sidebarContent #fs, #contentWrapper div [id * = 'myExtraContent'], #sidebarContent
div [id * = 'myExtraContent'] {display: block;}
```

Kermith's workshop (https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://www.kermith.com)

The other way to see supervision ...

Manage project preferences

We will now memorize the parameters of our application. It is indeed tedious to repeat the entry each time you use the application again. We will use a text file to save our configuration. It will be placed in a folder depending on the type of platform.

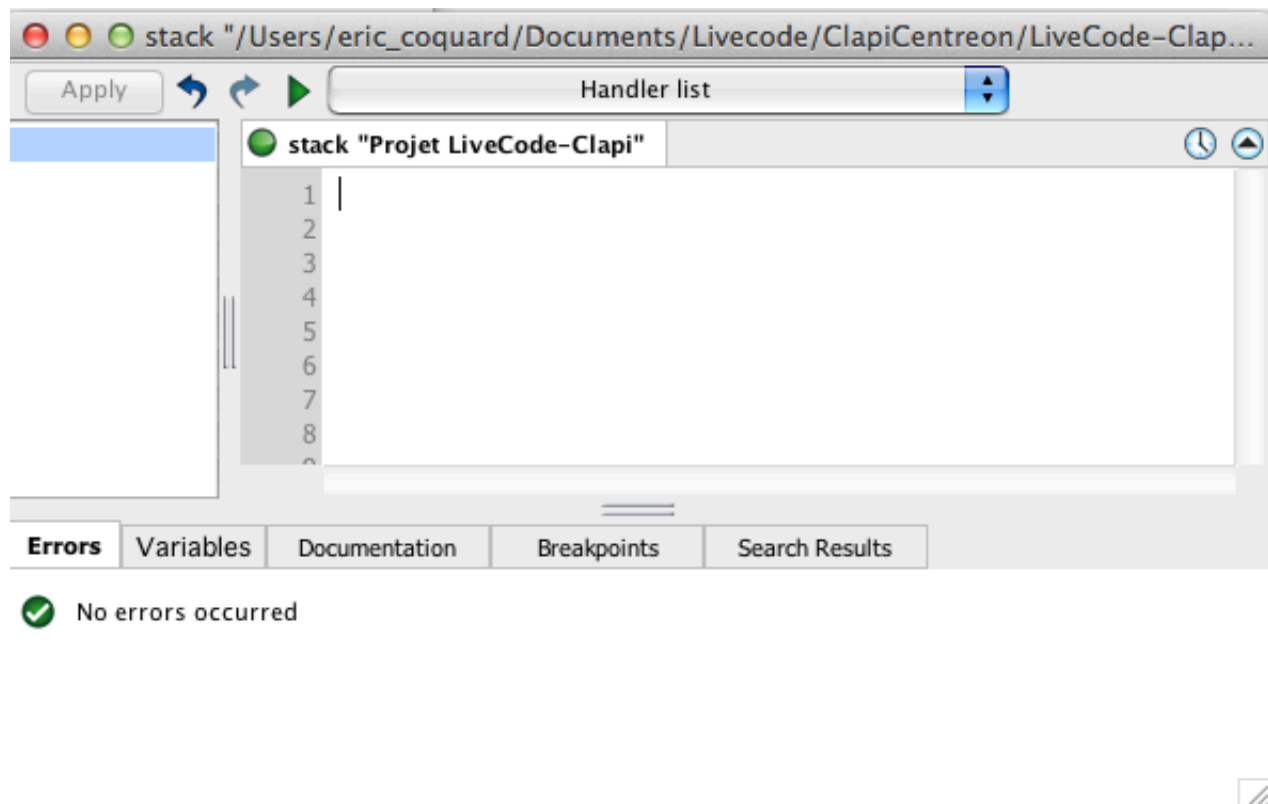
The program will be inspired by an article by Klaus Major from [REV UP n ° 99](#)

([https://translate.googleusercontent.com/translate_c?](https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://www.kermith.com)

[depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://www.kermith.com](https://translate.googleusercontent.com/translate_c?depth=1&hl=en&prev=search&pto=aue&rurl=translate.google.com&sl=fr&sp=nmt4&u=http://www.kermith.com))

Battery functions

We are going to need some functions to find the preferences folder for example, find the configuration file, etc. We will place these routines directly in the stack. Use the Object / Stack Script menu, the code window will display the empty "Handler List".



Code window

To add a function, enter the text below and validate.

```
## This works returns the preferences folder for the application
## depending on the platform
## OS X -> / Users / User name / Library / Preferences
## Windows XP -> C: / Documents and Settings / User Name / Applica
tion Data
## Windows 9x -> C: / windows / system
## Linux -> HOME / users folder
function ReturnPrefsFolder
  switch the platform
    "MacOS" box

    ## MacOS X
    put specialFolderPath ("preferences") into spfp
    break
  "Win32" box

  ## Everything> = Win2000
  if the systemversion contains "NT" then
    put specialFolderPath (26) into spfp
  else

    ## Win95, 96, 97, 98 :- )
    put specialFolderPath ("system") into spfp
  end if
  break
default

  ## Linux
  put $ HOME into spfp
  break
end switch

## Creation of a sub-folder with the name of the application
## if it does not exist
if there is not a folder (spfp & "/ LiveCode_Clapi") then
  create folder (spfp & "/ LiveCode_Clapi")
end if
## We return the path including the slash
return (spfp & "/ LiveCode_Clapi /")
end ReturnPrefsFolder
```

This function will determine the folder where the configuration file will be located depending on the platform used.

```

1  ## Cette fonctionne retourne le dossier préférences pour l'application
2  ## en fonction de la plateforme
3  ## OS X -> /Users/User name/Library/Preferences
4  ## Windows XP -> C:/Documents and Settings/User Name/Application Data
5  ## Windows 9x -> C:/windows/system
6  ## Linux -> Dossier HOME/users
7  function ReturnPrefsFolder
8      switch the platform
9          case "MacOS"
10
11         ## MacOS X
12         put specialFolderPath("preferences") into spfp
13         break
14     case "Win32"
15
16     ## Everything >= Win2000
17     if the systemversion contains "NT" then
18         put specialFolderPath(26) into spfp
19     else
20
21     ## Win95, 96, 97, 98 :- )
22     put specialFolderPath("system") into spfp
23     end if
24     break
25 default
26
27     ## Linux
28     put $HOME into spfp
29     break
30 end switch
31
32 ## Création d'un sous dossier du nom de l'application
33 ## si celui-ci n'existe pas
34 if there is not a folder (spfp & "/LiveCode_Clapi") then
35     create folder (spfp & "/LiveCode_Clapi")
36 end if
37 ## On retourne le chemin incluant le slash
38 return (spfp & "/LiveCode_Clapi/")
39 end ReturnPrefsFolder
  
```

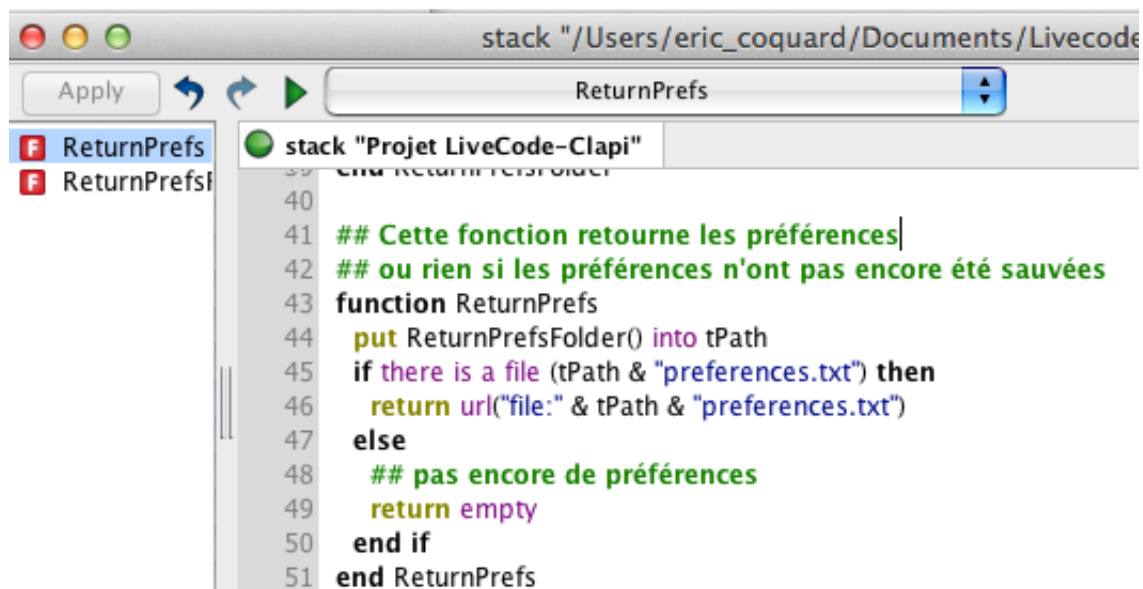
ReturnPrefsFolder function code

La fonction suivante permet de récupérer les informations contenues dans le fichier de configuration, s'il existe.

```

## Cette fonction retourne les préférences
## ou rien si les préférences n'ont pas encore été sauées
function ReturnPrefs
  put ReturnPrefsFolder() into tPath
  if there is a file (tPath & "preferences.txt") then
    return url("file:" & tPath & "preferences.txt")
  else
    ## pas encore de préférences
    return empty
  end if
end ReturnPrefs

```



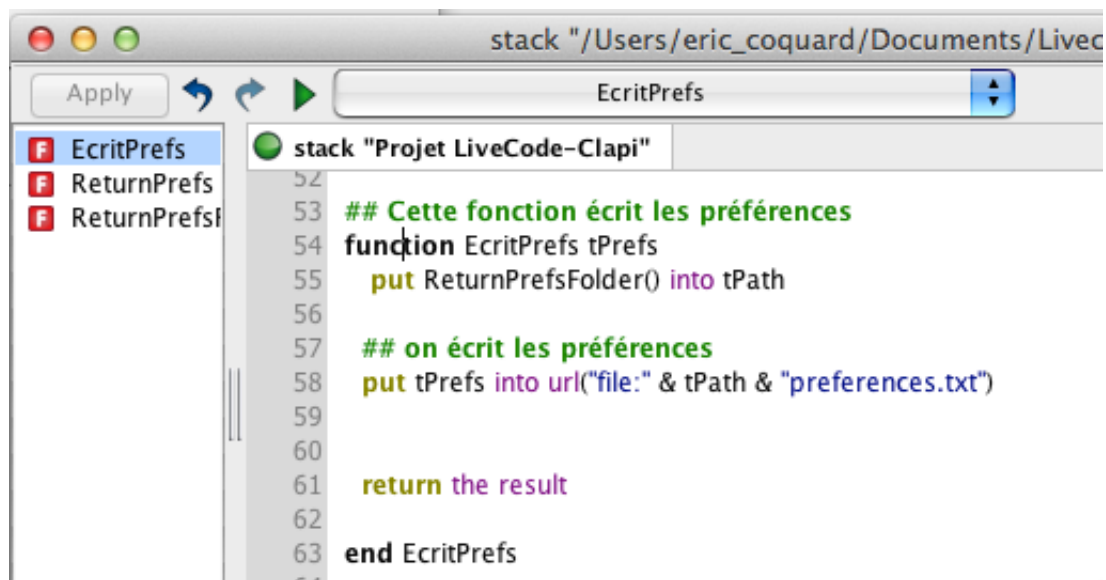
Code de la fonction ReturnPrefs

La dernière fonction écrira les informations saisies dans les zones de texte vers le fichier de configuration. Attention, actuellement le mot de passe est sauvegardé en clair.

```

## Cette fonction retourne les préférences
## ou rien si les préférences n'ont pas encore été sauées
function ReturnPrefs
  put ReturnPrefsFolder() into tPath
  if there is a file (tPath & "preferences.txt") then
    return url("file:" & tPath & "preferences.txt")
  else
    ## pas encore de préférences
    return empty
  end if
end ReturnPrefs

```



Fonction EcritPrefs

Script sauvegarde préférence et lecture des préférences

Le code pour lire les préférences sera placé dans le bouton permettant l'accès à la configuration du programme (BtnConf). Le code pour la sauvegarde des préférences sera placé dans le bouton permettant le retour à la carte principale (BtnMain).

Voici le code de sauvegarde des paramètres :

```
on mouseUp
  put "field" & TAB & "FldServeur" & TAB & fld "FldServeur" & CR in
  put "field" & TAB & "FldUserServeur" & TAB & fld "FldUserServeur"
  put "field" & TAB & "FldUserClapi" & TAB & fld "FldUserClapi" & C
  put "field" & TAB & "FldPasswordClapi" & TAB & fld "FldPasswordC
  put "field" & TAB & "FldCmdSSH" & TAB & fld "FldCmdSSH" & CR afte
  put "field" & TAB & "FldPathClapi" & TAB & fld "FldPathClapi" & C

  ## Stockage des valeurs
  put EcritPrefs(tPrefs) into tResult

  ## vérification
  if tResult <> empty then
    answer "Problème lors de la sauvegarde des préférences" & CR &

  end if

  go to card "Main"
end mouseUp
```

```

button "BtnMain" of card "Configuration" of stack "/Users/eric_coquard/Documents/Livecode/Cl
Apply mouseUp
H mouseUp
1 on mouseUp
2   put "field" & TAB & "FldServeur" & TAB & fld "FldServeur" & CR into tPrefs
3   put "field" & TAB & "FldUserServeur" & TAB & fld "FldUserServeur" & CR after tPrefs
4   put "field" & TAB & "FldUserClapi" & TAB & fld "FldUserClapi" & CR after tPrefs
5   put "field" & TAB & "FldPasswordClapi" & TAB & fld "FldPasswordClapi" & CR after tPrefs
6   put "field" & TAB & "FldCmdSSH" & TAB & fld "FldCmdSSH" & CR after tPrefs
7   put "field" & TAB & "FldPathClapi" & TAB & fld "FldPathClapi" & CR after tPrefs
8
9   ## Stockage des valeurs
10  put EcritPrefs(tPrefs) into tResult
11
12  ## vérification
13  if tResult <> empty then
14    answer "Problème lors de la sauvegarde des préférences" & CR & tResult
15  end if
16
17  go to card "Main"
18 end mouseUp
19

```

Sauvegarde des paramètres

Et le code permettant la lecture des paramètres du bouton btnConf :


```
on mouseUp
  ## récupère les informations du fichiers des préférences
  put ReturnPrefs() into tPrefs

  ## on vérifie la présence d'informations
  if not tPrefs = empty then
    ## figeage de l'affichage
    lock screen

    ##on défini la tabulation pour séparer les items
    set itemdel to TAB

    ##Pour chaque ligne on récupère les informations
    ##item 1 type de contrôle
    ##item 2 nom du contrôle
    ##item 3 propriété (contents) du contrôle
    repeat for each line i in tPrefs
      put item 1 of i into tType
      put item 2 of i into tName
      put item 3 of i into tValue

      ##contrôle Text Field
      switch tType
        case "field"
          put tValue into fld tName of card "Configuration"
          break
      end switch

      ##il est possible de rajouter d'autre contrôle
    end repeat
    ##on libère l'affichage
    unlock screen
  end if

  ## on affiche la carte configuration
  go to card "Configuration"
end mouseUp
```

```

on mouseUp
  ## récupère les informations du fichiers des préférences
  put ReturnPrefs() into tPrefs

  ## on vérifie la présence d'informations
  if not tPrefs = empty then
    ## figeage de l'affichage
    lock screen

    ##on défini la tabulation pour séparer les items
    set itemdel to TAB

    ##Pour chaque ligne on récupère les informations
    ##item 1 type de contrôle
    ##item 2 nom du contrôle
    ##item 3 propriété (contents) du contrôle
    repeat for each line i in tPrefs
      put item 1 of i into tType
      put item 2 of i into tName
      put item 3 of i into tValue

      ##contrôle Text Field
      switch tType
        case "field"
          put tValue into fld tName of card "Configuration"
          break
      end switch

      ##il est possible de rajouter d'autre contrôle
    end repeat
    ##on libère l'affichage
    unlock screen
  end if

  ## on affiche la carte configuration
  go to card "Configuration"
end mouseUp

```

Lecture des paramètres

Et voilà, votre programme mémorise automatique les valeurs saisies dans la fenêtre de configuration. Il restera à encoder le mot de passe pour assurer un maximum de sécurité. Avant de compiler votre application, n'oubliez pas de mettre les valeurs des zones de texte à blanc, car LiveCode a la particularité de sauvegarder dans les contrôles la dernière valeur saisie.

Encodage du mot de passe

Pour encoder et décoder le mot de passe, nous aurons besoin de deux fonctions. Ces fonctions sont inspiré de [cet article](http://livecodejournal.com/tutorials/handy-handlers-) (<http://livecodejournal.com/tutorials/handy-handlers->

[005.html](#)). Installez celles-ci dans la pile (stock). Je vous conseille de mettre votre chaîne personnalisée à la place de "A\$èdti4s#op"

Fonction Cryptage

```
function ReturnCrypte pData
  local tMd5, tRandSeed, tMax, tOffsetsA
  local tStackName, tSaveVis, tFile
  local pPassword

  put "A$èdti4s#op" into pPassword
  --
  if pPassword is empty then
    -- Simply mark the data as not password-protected
    -- and compress it without further modification:
    put "00"&compress(pData) into pData
  else
    -- Get MD5 digest:
    put md5digest(pPassword) into tKeyString
    put len(tKeyString) into tKeyStringLen
    --
    put compress(pData) into pData
    --
    -- Apply it with Xor to the data one byte at a time:
    put 0 into i
    put empty into tCryptoText
    repeat for each char k in pData
      add 1 to i
      if i > tKeyStringLen then put 1 into i
      put char i of tKeyString into tKeyChar
      put numtochar( chartonum(k) bitxor chartonum(tKeyChar))\
        after tCryptoText
    end repeat
    --
    -- Mark the data as password-protected:
    put "01" & tCryptoText into pData
  end if
  -- Convert to common low ASCII:
  return base64encode(pData)
end ReturnCrypte
```

The screenshot shows a LiveCode stack editor window titled "stack \"/>

```

65 function ReturnCrypte pData
66   local tMd5, tRandSeed, tMax, tOffsetsA
67   local tStackName, tSaveVis, tFile
68   local pPassword
69
70   put "A§èdti4s#op" into pPassword
71   --
72   if pPassword is empty then
73     -- Simply mark the data as not password-protected
74     -- and compress it without further modification:
75     put "00"&compress(pData) into pData
76   else
77     -- Get MD5 digest:
78     put md5digest(pPassword) into tKeyString
79     put len(tKeyString) into tKeyStringLength
80     --
81     put compress(pData) into pData
82     --
83     -- Apply it with Xor to the data one byte at a time:
84     put 0 into i
85     put empty into tCryptoText
86     repeat for each char k in pData
87       add 1 to i
88       if i > tKeyStringLength then put 1 into i
89       put char i of tKeyString into tKeyChar
90       put numtochar( charnum(k) bitxor charnum(tKeyChar))\
91         after tCryptoText
92     end repeat
93     --
94     -- Mark the data as password-protected:
95     put "01" & tCryptoText into pData
96   end if
97   -- Convert to common low ASCII:
98   return base64encode(pData)
99 end ReturnCrypte

```

function ReturnCrypte

Fonction Décryptage

```

function ReturnDecrypte pData
  local tEncryptionMethod
  local tMd5, tRandSeed, tMax, tOffsetsA
  local pPassword
  put "A§èdti4s#op" into pPassword

```

```
--
-- Convert from base64 back to binary:
put base64decode(pData) into pData
-- Check and remove password-protection flag:
put char 1 to 2 of pData into tEncryptionMethod
delete char 1 to 2 of pData
--
switch tEncryptionMethod
case "00" --no encryption
  break
--
case "01" -- mdx
  -- Get MD5 digest:
  put md5digest(pPassword) into tKeyString
  put len(tKeyString) into tKeyStringLen
  --
  -- Apply it with Xor to the data one byte at a time:
  put 0 into i
  put empty into tClearText
  repeat for each char k in pData
    add 1 to i
    if i > tKeyStringLen then put 1 into i
    put char i of tKeyString into tKeyChar
    put numtochar( chartonum(k) bitxor chartonum(tKeyChar))\
      after tClearText
  end repeat
  put tClearText into pData
  break
--
end switch
--
-- Attempt to decompress data, throwing an error if not valid:
try
  get decompress(pData)
  catch errNo
  throw "Mauvaise Clé"
  exit to top
finally
  return it
end try
end ReturnDecrypte
```

The screenshot shows a LiveCode IDE window with the title bar 'stack "/>

```

101 function ReturnDecrypte pData
102   local tEncryptionMethod
103   local tMd5, tRandSeed, tMax, tOffsetsA
104   local pPassword
105   put "A§èdti4s#op" into pPassword
106   --
107   -- Convert from base64 back to binary:
108   put base64decode(pData) into pData
109   -- Check and remove password-protection flag:
110   put char 1 to 2 of pData into tEncryptionMethod
111   delete char 1 to 2 of pData
112   --
113   switch tEncryptionMethod
114   case "00" --no encryption
115     break
116     --
117   case "01" -- mdx
118     -- Get MD5 digest:
119     put md5digest(pPassword) into tKeyString
120     put len(tKeyString) into tKeyStringLength
121     --
122     -- Apply it with Xor to the data one byte at a time:
123     put 0 into i
124     put empty into tClearText
125     repeat for each char k in pData
126       add 1 to i
127       if i > tKeyStringLength then put 1 into i
128       put char i of tKeyString into tKeyChar
129       put numtochar( chartonum(k) bitxor chartonum(tKeyChar))\
130         after tClearText
131     end repeat
132     put tClearText into pData
133     break
134     --
135   end switch
136   --
137   -- Attempt to decompress data, throwing an error if not valid:
138   try
139     get decompress(pData)
140     catch errNo
141     throw "Mauvaise Clé"
142     exit to top
143   finally
144     return it
145   end try
146 end ReturnDecrypte

```

function ReturnDecrypte

Modification de la gestion des préférences

Modifions les scripts des boutons BtnMain et BtnConf. Commencez par le script BtnMain qui cryptera le mot de passe. Remplacez la ligne appropriée.

```
put "field" & TAB & "FldPasswordClapi" & TAB & ReturnCrypte(fld "F
ldPasswordClapi") & CR after tPrefs
```

```
put "field" & TAB & "FldPasswordClapi" & TAB & ReturnCrypte(fld "FldPasswordClapi") & CR after tPrefs
function mouseUp de BtnMain
```

Continuez par le script BtnConf qui décryptera le mot de passe. On ajoutera la condition pour le contrôle Field FldPasswordClapi.

```
switch tType
case "field"
    if tName = "FldPasswordClapi" then
        put ReturnDecrypte (tValue) into fld tName of card "C



    else
        put tValue into fld tName of card "Configuration"
    end if
    break
end switch
```

```
switch tType
case "field"
    if tName = "FldPasswordClapi" then
        put ReturnDecrypte(tValue) into fld tName of card "Configuration"
    else
        put tValue into fld tName of card "Configuration"
    end if
    break
function mouseUp from BtnConf
```

You can check the result by listing the preferences file as below:

```
field FldServer 172.16.209.174
field FldUserServer vmdebian
field FldUserClapi admin
field FldPasswordClapi MDGsa4Hrx2NGwd1XP18CaIjenKqI6xIlhPTVVBQX
field FldCmdSSH / usr / bin / ssh
field FldPathClapi / usr / local / centreon / www / modules / cent
reon-clapi / core / centreon
```

0 Commentaires Sugarbug  Règles de confidentialité de Disqus  1 S'identifier ▼

 Recommander  Tweet  Partager Les meilleurs ▼



Commencer la discussion...

S'IDENTIFIER AVEC

OU INSCRIVEZ-VOUS SUR DISQUS 

Nom

Soyez le premier à commenter.

 S'abonner  Ajoutez Disqus à votre site web !Ajouter DisqusAjouter

© 2020 Eric Coquard [Contact me \(mailto:eric.coquard@sugarbug.fr\)](mailto:eric.coquard@sugarbug.fr) Last modified: 06/30/2020