



SkinMaker : création interface

Par
Gilles MELLOTT, Responsable Notes Techniques
Note technique 4D-200004-13-FR
Version 1
Date 1 Avril 2000

Résumé

Cet outil permet de créer une bibliothèque de modèles d'interfaces, dont l'utilisation en exploitation, se résumera par deux commandes : Skin_Fill (le chargement) et Skin_Calc (le calcul final de l'interface).

4D Notes techniques

Copyright © 1985-2004 4D SA - Tous droits réservés

Tous les efforts ont été faits pour que le contenu de cette note technique présente le maximum de fiabilité possible. Néanmoins, les différents éléments composant cette note technique, et le cas échéant, le code, sont fournis sans garantie d'aucune sorte. L'auteur et 4D S.A. déclinent donc toute responsabilité quant à l'utilisation qui pourrait être faite de ces éléments, tant à l'égard de leurs utilisateurs que des tiers.

Les informations contenues dans ce document peuvent faire l'objet de modifications sans préavis et ne sauraient en aucune manière engager 4D SA. La fourniture du logiciel décrit dans ce document est régie par un octroi de licence dont les termes sont précisés par ailleurs dans la licence électronique figurant sur le support du Logiciel et de la Documentation afférente. Le logiciel et sa documentation ne peuvent être utilisés, copiés ou reproduits sur quelque support que ce soit et de quelque manière que ce soit, que conformément aux termes de cette licence.

Aucune partie de ce document ne peut être reproduite ou recopiée de quelque manière que ce soit, électronique ou mécanique, y compris par photocopie, enregistrement, archivage ou tout autre procédé de stockage, de traitement et de récupération d'informations, pour d'autres buts que l'usage personnel de l'acheteur, et ce exclusivement aux conditions contractuelles, sans la permission explicite de 4D SA.

4D, 4D Calc, 4D Draw, 4D Write, 4D Insider, 4ème Dimension ®, 4D Server, 4D Compiler ainsi que les logos 4e Dimension, sont des marques enregistrées de 4D SA.

Windows, Windows NT, Win 32s et Microsoft sont des marques enregistrées de Microsoft Corporation.

Apple, Macintosh, Power Macintosh, LaserWriter, ImageWriter, QuickTime sont des marques enregistrées ou des noms commerciaux de Apple Computer, Inc.

Mac2Win Software Copyright © 1990-2002 est un produit de Altura Software, Inc.

4D Write contient des éléments de "MacLink Plus file translation", un produit de DataViz, Inc, 55 Corporate drive, Trumbull, CT, USA.

XTND Copyright 1992-2002 © 4D SA. Tous droits réservés.

XTND Technology Copyright 1989-2002 © Claris Corporation.. Tous droits réservés ACROBAT © Copyright 1987-2002, Secret Commercial Adobe Systems Inc. Tous droits réservés. ACROBAT est une marque enregistrée d'Adobe Systems Inc.

Tous les autres noms de produits ou appellations sont des marques déposées ou des noms commerciaux appartenant à leurs propriétaires respectifs.

Introduction

La mode étant aux "skins", 4D ne sera pas en reste. Cette note technique vous propose de créer vos fonds de formulaires. Ceci, non de manière statique par la création d'une image aux dimensions souhaitées, mais par la gestion d'une bibliothèque de modèles, pouvant être utilisés dynamiquement.

En cas d'intégration à vos développements, vous permettrez à vos clients de changer ou créer leur interface personnelle sans modification du code source.

Le principe

À l'aide de neuf images (4 coins, 4 bordures, un centre) en utilisant les opérations sur les images de 4ème Dimension, nous désirons créer une image finale servant de fond d'écran à un formulaire.

Cette image devant pouvoir être recalculée suivant la fenêtre de réception (nous ne voulons pas ici de redimensionnement de l'image qui nous la déformerait).

Chaque modèle ainsi défini, devra être stocké pour une utilisation inter ou intra-développement.

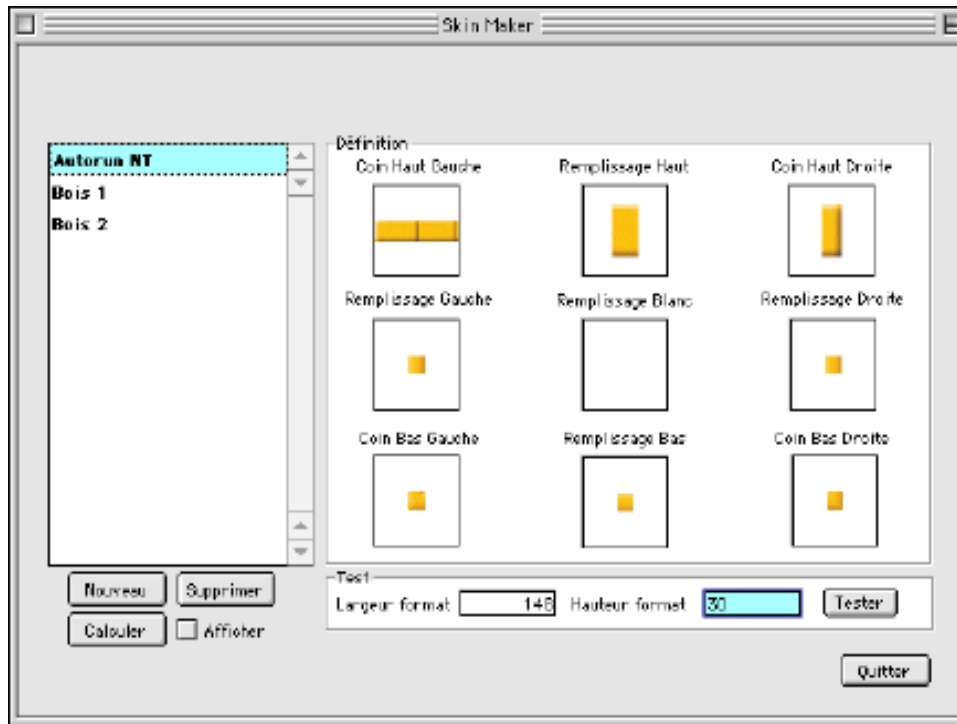
Nous utiliserons à cet usage un blob stocké dans une ressource 'Skin', ne comportant que les éléments devant servir au calcul de l'image finale.

Le calcul final de l'image d'interface sera stocké dans la bibliothèque d'images sous deux formes : le contour et le fond de remplissage du "blanc" intérieur. Ceci nous permettant de gérer les éléments de l'interface séparément. Ces images auront été préalablement placées en page 0 du formulaire et ont les paramètres : agrandir horizontalement et verticalement.

Note : il faudra obligatoirement un couple Contour/Remplissage pour chaque formulaire de dimensions différentes. Ceci afin d'éviter le calcul à chaque changement de formulaire.

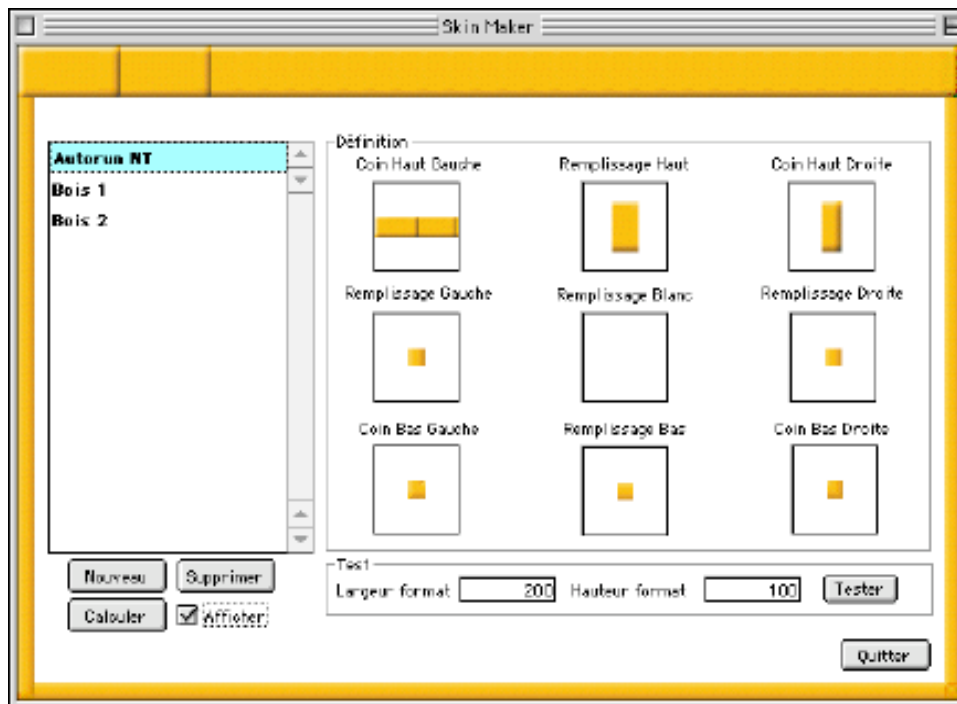
La base exemple

Sélectionnez l'item "SkinMaker" dans le menu Fichier.



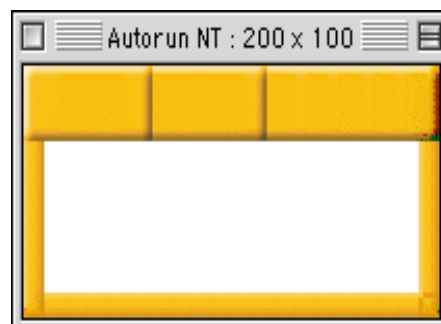
Éléments d'interface :

- Liste des Modèles disponibles. Peuvent être renommés en cliquant sur l'item désiré, la touche commande (MacOS), la touche Alt (Windows) enfoncée.
- Bouton Nouveau : ajoute une entrée dans la liste hiérarchique et dans le ressource fork de la structure.
- Bouton Supprimer : élimine l'item sélectionné de la liste et supprime la ressource correspondante.
- Bouton Calculer : lance la construction de l'image avec les images collées dans la zone centrale.
- Boîte à cocher Afficher : cache ou montre l'interface telle qu'elle a été calculée. En position sélectionnée, toutes les modifications apportées aux images ou toute modification de sélection de modèle entraînera dynamiquement le recalcul de l'interface.



- Largeur format et Hauteur format : dimensions souhaitées de la fenêtre pour la création d'une interface générée par le bouton Tester.

- Bouton Tester : calcule une nouvelle image aux dimensions Largeur format et Hauteur format, puis ouvre une fenêtre aux dimensions souhaitées, afin de visualiser le résultat.



- Groupe Définition : zone de saisie des éléments de l'image finale, éditables par Coller/Copier/Couper. Les images peuvent être de tailles différentes et bénéficient d'un tips affichant leurs dimensions.

- Bouton Quitter : sauve le modèle en cours d'édition et ferme le dialogue.

Les Méthodes

Méthode projet *Skin Save* :

Chargée de construire le blob et de sauvegarder celui-ci dans une ressource 'Skin'.

```
C_BLOB($blob)
```

```
$offset:=0
```

```
VARIABLE VERS BLOB(P_Coin_HD;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Coin_HG;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Coin_BD;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Coin_BG;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Left;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Right;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Top;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Bottom;$blob;$offset)
```

```
VARIABLE VERS BLOB(P_Center;$blob;$offset)
```

```
ECRIRE RESSOURCE("skin";◇Skin_Id{$1};$blob;◇LocalResFile)
```

```
ECRIRE NOM RESSOURCE("skin";◇Skin_Id{$1};◇Skin_name{$1};◇LocalResFile)
```

Méthode Projet Skin_Fill :

Cette méthode lit la ressource 'Skin' avec l'identifiant passé en paramètre et affecte les images avec le contenu du blob.

```
C_BLOB($blob)
```

```
C_ENTIER LONG($offset;$1)
```

```
LIRE RESSOURCE("skin";$1;$blob;◇LocalResFile)
```

```
C_IMAGE($Pict)
```

```
BLOB VERS VARIABLE($blob;P_Coin_HD;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Coin_HG;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Coin_BD;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Coin_BG;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Left;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Right;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Top;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Bottom;$offset)
```

```
BLOB VERS VARIABLE($blob;P_Center;$offset)
```

```
Si (F_PictSize (P_Center))
```

```
  ECRIRE IMAGE DANS BIBLIOTHEQUE(P_Center;1;"BackGround")
```

```
Sinon
```

```
  SUPPRIMER IMAGE DANS BIBLIOTHEQUE(1)
```

```
Fin de si
```

```
  `calcul de la taille minimale
```

```
viHeight:=0
```

```
viWidth:=0
```

```
Skin_VerifSize
```

```
  `construction des tips
```

```
  `inutile en exploitation
```

```
  `ne sert que dans cette note
```

```
$ok:=F_PictSize (P_Coin_HG)
```

```
Calc_Tips (1)
```

```
$ok:=F_PictSize (P_Top)
```

```
Calc_Tips (2)
```

```
$ok:=F_PictSize (P_Coin_HD)
Calc_Tips (3)
$ok:=F_PictSize (P_Left)
Calc_Tips (4)
$ok:=F_PictSize (P_Center)
Calc_Tips (5)
$ok:=F_PictSize (P_Right)
Calc_Tips (6)
$ok:=F_PictSize (P_Coin_BG)
Calc_Tips (7)
$ok:=F_PictSize (P_Bottom)
Calc_Tips (8)
$ok:=F_PictSize (P_Coin_BD)
Calc_Tips (9)
```

Méthode projet Skin_VerifSize :

Calcul de la taille minimale de l'image finale.

C_ENTIER LONG(viWidth;viHight)

```
$ok:=F_PictSize (P_Coin_HG)
$large1:=◇Large
$haut1:=◇Haut
```

```
$ok:=F_PictSize (P_Coin_HD)
$large2:=◇Large
$haut2:=◇Haut
```

```
$ok:=F_PictSize (P_Coin_BG)
$haut3:=◇Haut
$Large3:=◇Large
```

```
$ok:=F_PictSize (P_Coin_BD)
$haut4:=◇Haut
$Large4:=◇Large
```

```
$ok:=F_PictSize (P_Top)
$haut5:=◇Haut
$Large5:=◇Large
```

```
$ok:=F_PictSize (P_Bottom)
$haut6:=◇Haut
$Large6:=◇Large
```

```
$ok:=F_PictSize (P_Left)
$haut7:=◇Haut
$Large7:=◇Large
```

```
$ok:=F_PictSize (P_Right)
$haut8:=◇Haut
$Large8:=◇Large
```

```
$miniLH:=$large1+$large2+$large6
$miniLB:=$large3+$large6+$large4
```

```
S i ($miniLH<$miniLB)
$miniLH:=$miniLB
```

```

Fin de si

$miniHG:=$haut1+$haut7+$haut3
$miniHD:=$haut2+$haut8+$haut4
Si ($miniHG<$miniHD)
  $miniHG:=$miniHD
Fin de si

Si (viWidth<$miniLH)
  viWidth:=$miniLH
Fin de si
Si (viHight<$miniHG)
  viHight:=$miniHG

Fin de si

```

Méthode projet *Skin Calc* :

Cette méthode est le moteur de cette note. Ici nous calculons l'image finale aux dimensions passées en paramètres. Nous n'utilisons que les opérateurs sur les images.

```

C_IMAGE($0;$pict;$atemp)
C_ENTIER LONG($1;$2) `largeur, hauteur

$pict:=◇PictVide

$sizeTop_Hor:=$1
$sizeLeft_vert:=$2
$sizeBottom_Hor:=$1
$sizeRight_vert:=$2

$pict:=◇pictvide
$offsetLeft:=0
$offsetRight:=0
$offsetTop:=0
$offsetBottom:=0

Si (F_PictSize (P_Coin_HG))
  $pict:=$pict | P_Coin_HG
  $sizeTop_Hor:=$sizeTop_Hor-◇Large
  $sizeLeft_vert:=$sizeLeft_vert-◇Haut
  $offsetLeft:=◇Haut
  $offsetTop:=◇Large
Fin de si

Si (F_PictSize (P_Coin_HD))
  $sizeTop_Hor:=$sizeTop_Hor-◇Large
  $sizeRight_vert:=$sizeRight_vert-◇Haut
  $aTemp:=P_Coin_HD+($1-◇Large)
  $pict:=$pict | $atemp
  $offSetRight:=◇Haut
Fin de si

Si (F_PictSize (P_Coin_BG))
  $sizeLeft_vert:=$sizeLeft_vert-◇Haut

```

```
$sizeBottom_Hor:=$sizeBottom_Hor-◇Large
$aTemp:=P_Coin_BG/($2-◇Haut)
$pict:=$pict | $atemp
$offsetBottom:=◇Large
```

Fin de si

```
Si (F_PictSize (P_Coin_BD))
  $sizeRight_vert:=$sizeRight_vert-◇Haut
  $sizeBottom_Hor:=$sizeBottom_Hor-◇Large
  $aTemp:=P_Coin_BD/($2-◇Haut)
  $atemp:=$atemp+($1-◇Large)
  $pict:=$pict | $atemp
```

Fin de si

```
Si (F_PictSize (P_Left))
  $reste:=$sizeLeft_vert
  Boucle ($i;0;$sizeLeft_vert;◇Haut)
    Si ($reste>=◇Haut)
      $aTemp:=P_Left/($i+$offsetLeft)
      $pict:=$pict | $atemp
      $reste:=$reste-◇Haut
    Sinon
      $coef:=$reste/◇Haut
      $aTemp:=P_Left*/($coef)
      $aTemp:=$aTemp/($i+$offsetLeft)
      $pict:=$pict | $atemp
```

Fin de si

Fin de boucle

Fin de si

```
Si (F_PictSize (P_Right))
  $reste:=$sizeRight_vert
  Boucle ($i;0;$sizeRight_vert;◇Haut)
    Si ($reste>=◇Haut)
      $aTemp:=P_Right/($i+$offsetRight)
      $atemp:=$atemp+($1-◇Large)
      $pict:=$pict | $atemp
      $reste:=$reste-◇Haut
    Sinon
      $coef:=$reste/◇Haut
      $aTemp:=P_Right*/($coef)
      $aTemp:=$aTemp/($i+$offsetRight)
      $atemp:=$atemp+($1-◇Large)
      $pict:=$pict | $atemp
```

Fin de si

Fin de boucle

Fin de si

```
Si (F_PictSize (P_Top))
  $reste:=$sizeTop_Hor
  Boucle ($i;0;$sizeTop_Hor;◇Large)
    Si ($reste>=◇Large)
      $aTemp:=P_Top+($i+$offsetTop)
      $pict:=$pict | $atemp
      $reste:=$reste-◇Large
    Sinon
      $coef:=$reste/◇Large
      $aTemp:=P_Top*+($coef)
```



```

    $aTemp:=$aTemp+($i+$offsetTop)
    $pict:=$pict | $atemp
    Fin de si
    Fin de boucle
    Fin de si

    Si (F_PictSize (P_Bottom))
    $reste:=$sizeBottom_Hor
    Boucle ($i;0;$sizeBottom_Hor;◇Large)
    Si ($reste>=◇Large)
    $aTemp:=P_Bottom+($i+$offsetBottom)
    $atemp:=$atemp/($2-◇Haut)
    $pict:=$pict | $atemp
    $reste:=$reste-◇Large
    Sinon
    $coef:=$reste/◇Large
    $aTemp:=P_Bottom*+($coef)
    $aTemp:=$aTemp+($i+$offsetBottom)
    $atemp:=$atemp/($2-◇Haut)
    $pict:=$pict | $atemp
    Fin de si
    Fin de boucle
    Fin de si

    Si (F_PictSize ($pict))
    Si ($3)
    ECRIRE IMAGE DANS BIBLIOTHEQUE($pict;2;"Skin")
    Sinon
    ECRIRE IMAGE DANS BIBLIOTHEQUE($pict;4;"Skin_Test")
    Fin de si
    Sinon
    SUPPRIMER IMAGE DANS BIBLIOTHEQUE(2)
    SUPPRIMER IMAGE DANS BIBLIOTHEQUE(4)
    Fin de si

    Si (F_PictSize (P_Center))
    Si ($3)
    ECRIRE IMAGE DANS BIBLIOTHEQUE(P_Center;1;"BackGround")
    Sinon
    ECRIRE IMAGE DANS BIBLIOTHEQUE(P_Center;3;"BackGround_Test")
    Fin de si
    Sinon
    SUPPRIMER IMAGE DANS BIBLIOTHEQUE(1)
    SUPPRIMER IMAGE DANS BIBLIOTHEQUE(3)
    Fin de si

    `affichage si besoin
    CHOIX VISIBLE(*;"theSkin@";(bFond=1))
    `redessiner le formulaire

    ALLER A PAGE(Page formulaire courante)

```

Méthode objet du bouton 'Test' :

Crée une fenêtre et une interface aux coordonnées désirées.

En fait, ce serait la méthode utilisée en exploitation, à condition d'avoir chargé un modèle avec la commande *Skin_Fill*.

```
Skin_Calc (viWidth;viHight;Faux)  
  
$aWind:=Creer fenetre(50;50;50+viWidth;50+viHight;4;aSkinName+" : "+Chaine(viWidth)  
+" x "+Chaine(viHight);"U_Close")  
DIALOGUE([Interface];"INT_Skin")  
  
FERMER FENETRE($aWind)
```

Note : le troisième paramètre ici un booléen, sert de bascule pour l'enregistrement dans la bibliothèque d'images : en effet cette note ne possède que deux formulaires. Ce paramètre devra être remplacé par un ou deux entiers longs par exemple, pour la gestion de multiples formulaires de tailles différentes.