

## Analyseur de champs Blob

---

Par  
Gilles MELLOTT, Responsable Notes Techniques  
Note technique 4D-200002-06-FR  
Version 1  
Date 1 Février 2000

### Résumé

---

Cette note technique présente un utilitaire permettant de visualiser le contenu de tous les champs blob d'une structure en mode hexadécimal et Ascii. Ce module comporte en outre un convertisseur de bases numériques.

### 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

L'utilisation de champs blob pour stocker des structures de données autres que les types courants de 4ème dimension, posent un problème de lisibilité lors du développement.

Cette note technique vous propose un petit utilitaire sous forme de librairie 4D Insider (un seul formulaire et une seule méthode), vous permettant de visualiser le contenu de ces champs sous forme hexadécimale et Ascii.

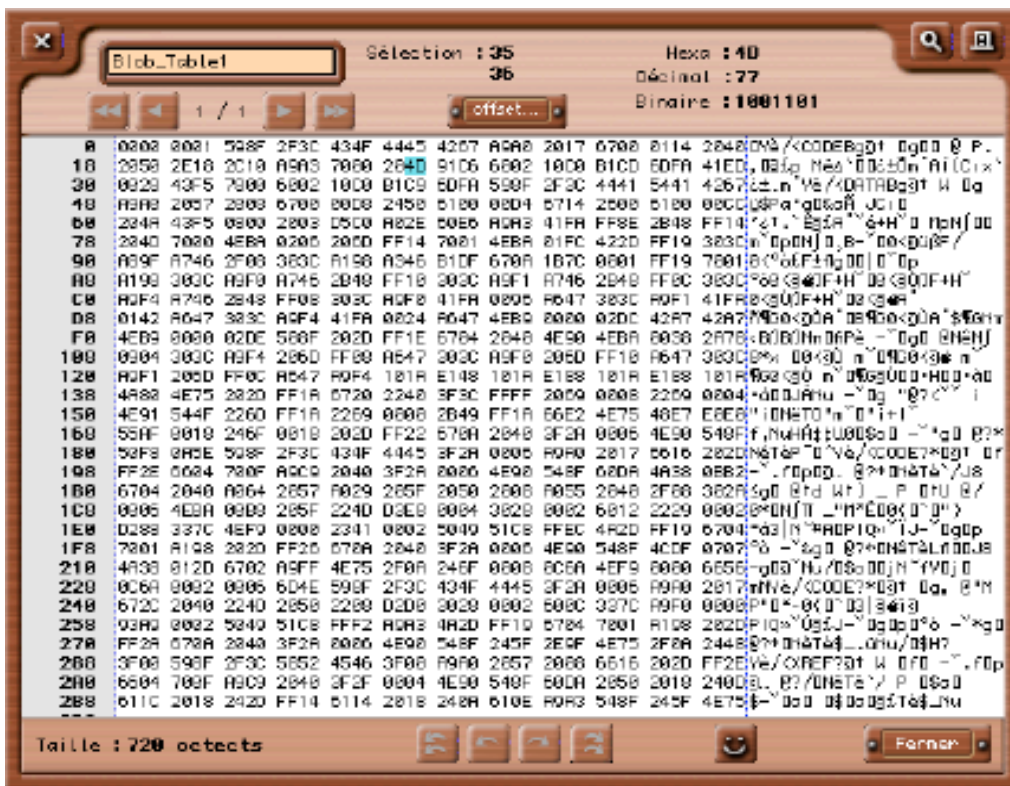
# Fonctionnement

La librairie fournie, peut être utilisée telle quelle dans toute structure de base 6.5x, et ou intégrée comme assistant dans un utilitaire d'aide au développement de type POPEUR (notes techniques 99-35 Décembre, 2000-05 Février et 2000-07 Mars).

L'objectif à atteindre : pouvoir naviguer dans les enregistrements et dans tous les champs de type blob, contenus dans une structure, ceci avec une seule méthode et un seul formulaire pour une facilité d'intégration.

Dans la base exemple, l'initialisation de deux variables effectuée dans la méthode Base Sur ouverture, permet l'utilisation du formulaire en utilisation directe sans lancer de process spécifique. Sinon, en mode Menus créés, sélectionnez l'item de menu Test...

## Le formulaire :

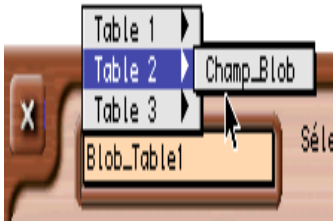


## Les Objets :

Case de fermeture du formulaire.



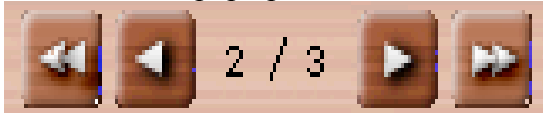
Pop up menu hiérarchique, permettant de naviguer entre les champs blob de la structure.



Bouton atteindre, sélectionne directement l'offset désiré.



En dessous du pop up menu les boutons de navigation entre les enregistrements.



## Éléments informatifs

- position de l'enregistrement dans la sélection
- offset de début de sélection hexadécimale
- valeur de la sélection en hexa
- valeur de la sélection en décimal
- valeur de la sélection en binaire
- taille totale du blob

Bouton ouvrant un dialogue de recherche de chaîne hexadécimale ou de chaîne ascii.





Impression du champ blob



Boutons de navigation dans le blob (pages de 720 octets)



Convertisseur de bases numériques



Enfin, un bouton Quitter



## Les méthodes

---

### La méthode projet Wood Blob :

Cette méthode, appelée depuis un menu ne reçoit pas de paramètres, et lance donc un nouveau process avec un paramètre.

Cette même méthode est appelée par des objets du formulaire, ou par la méthode formulaire elle même. Chaque appel peut être assimilé à une méthode en soi, et fonctionne comme telle. Ces "sous méthodes" sont regroupées uniquement dans le but de faciliter le portage du module.

```
◇Out:=1
Si (Nombre de parametres=0)
  ◇WoodBLOB:=Nouveau process("Wood_BLOB";64000;"Wood_BLOB";"Launch";*)
  PASSER AU PREMIER PLAN(◇WoodBLOB)
Sinon
  Au cas ou

  : ($1="Launch")
  `===== Votre table interface =====
  ◇ptrTable:=->[Table 1]
  `=====
  FORMULAIRE ENTREE(◇ptrTable->"Wood_Blob";*)
  Creer fenetre(30;50;-1;-1;2;"Wood Blob")
  DIALOGUE(◇ptrTable->"Wood_Blob")

  : ($1="Quit")
  `fermeture du process
  ◇WoodBLOB:=0

  : ($1="F_Dec2Base")
  `fonction F_Dec2Base
  `méthode de conversion d'une chaîne décimale en chaîne de base $3
```

Note : le tableau tabHexa est déclaré dans la méthode formulaire au cycle : Sur chargement.

```
C_TEXTE($2) `chaîne décimale à convertir
C_ENTIER LONG($3) `base
```

```
$str:=""
A:=Num($2)
Tant que (A>=$3)
  B:=Ent(Modulo(A;$3))
  $str:=$str+tabHexa{B+1}
  A:=Arrondi((A\3);0)
Fin tant que
  $str:=$str+tabHexa{A+1}
```

```

$L:=Longueur($str)
$0:=""
Boucle ($i;$L;1;-1)
  $0:=$0+$str≤$i≥
Fin de boucle

: ($1="F_Base2Dec")
  `fonction F_Base2Dec
  `méthode de conversion d'une chaîne de base x en chaîne décimale

```

```

C_TEXTE($2) `chaîne à convertir
C_ENTIER LONG($3) `base

```

```

$L:=Longueur($2)
$Dec:=0
Boucle ($i;0;$L-1)
  $char:=Sous chaîne($2;$L-$i;1)
  $v:=Num($char)
  Si (($v=0) & ($char#"0"))
    $v:=Code ascii($char)-55
  Fin de si
  $Dec:=$Dec+($v*($3^$i))
Fin de boucle
$0:=Chaîne($Dec)

```

```

: ($1="F_Ascii2Hexa")
  `F_Ascii2Hexa
  `méthode de conversion d'une chaîne ascii en chaîne hexadécimale

```

```

$source:=$2
$separator:=Caractere($3)
$L:=Longueur($source)
$Max:=Ent(32000/(4+Longueur($separator)))

```

```

$0:=""
Boucle ($i;1;$L;2)
  $char:=Sous chaîne(Chaîne(Code ascii($source≤$i≥);"&$00");2)
  $char:=$char+Sous chaîne(Chaîne(Code ascii($source≤$i+1≥);"&$00");2)
  Si ($separator="")
    $0:=$0+$char
  Sinon
    $0:=$0+$char+($separator*Num($i<$Max))
  Fin de si
Fin de boucle

```

```

C_ALPHA(8;$hex)
$fullRange:="" *6
Range=""
Boucle ($i;0;29)
  $hex:=Wood_BLOB ("F_Dec2Base";Chaîne($i*24+(720*◇cpt));16)
  Range:=Range+Sous chaîne($fullRange;1;6-Longueur($hex))+$hex+Caractere(13)
Fin de boucle

```

```

: ($1="UpDate")

```

```

  `redéfinition du contenu de l'affichage dans le formulaire de visualisation,
  `lors du changement de page à l'intérieur d'un blob. Pour une raison
  `d'interface, les pages sont fixées à 720 octets

```

```

Si (◇cpt>0)
  ACTIVER BOUTON(PreviousSeg)
  ACTIVER BOUTON(FirstSeg)
Sinon
  INACTIVER BOUTON(PreviousSeg)
  INACTIVER BOUTON(FirstSeg)
Fin de si

$offset:=(720*◇cpt)
Si ($offset>theSize)
  $offset:=TheSize-(theSize%720)
Fin de si

Ascii_txt:=BLOB vers texte(Champ(◇Table;◇field)->Texte sans longueur ;$offset;720)
Hexa_txt:=Wood_BLOB ("F_Ascii2Hexa";Ascii_txt;32)
Ascii_txt:=Remplacer chaine(Ascii_txt;Caractere(13);Caractere(0))

theSize:=Taille BLOB(Champ(◇Table;◇field)->)
Si (theSize>0)
  ACTIVER BOUTON(*;"navAction@")
Fin de si

Si (◇cpt<(theSize\720))
  ACTIVER BOUTON(NextSeg)
  ACTIVER BOUTON(LastSeg)
Sinon
  INACTIVER BOUTON(NextSeg)
  INACTIVER BOUTON(LastSeg)
Fin de si

$fiche:=Numero dans selection(Table(◇Table)->)
$totfiches:=Enregistrements dans table(Table(◇Table)->)
aSelect:=Chaine($fiche)+" / "+Chaine($totfiches)
Si ($fiche>1)
  ACTIVER BOUTON(*;"navEnregBack@")
Sinon
  INACTIVER BOUTON(*;"navEnregBack@")
Fin de si
Si ($fiche<$totFiches)
  ACTIVER BOUTON(*;"navEnregNext@")
Sinon
  INACTIVER BOUTON(*;"navEnregNext@")
Fin de si

Fin de cas

Fin de si

```

## La méthode Formulaire Wood Blob :

```

Si (◇Out<2) & (Page formulaire courante=1)

Au cas ou

```

: (**Evenement formulaire=**Sur chargement )

`déclaration du tableau utilisé dans la méthode projet Wood\_Blob

**C\_ENTIER LONG**(theSelect;theEnd;theSize)

**TABLEAU ALPHA**(1;TabHexa;16)

TabHexa{1}:="0"

TabHexa{2}:="1"

TabHexa{3}:="2"

TabHexa{4}:="3"

TabHexa{5}:="4"

TabHexa{6}:="5"

TabHexa{7}:="6"

TabHexa{8}:="7"

TabHexa{9}:="8"

TabHexa{10}:="9"

TabHexa{11}:="A"

TabHexa{12}:="B"

TabHexa{13}:="C"

TabHexa{14}:="D"

TabHexa{15}:="E"

TabHexa{16}:="F"

**C\_ALPHA**(21;vB;vT;vQ;vO;vD;vH)

- ` analyse de la structure hôte et
- ` construction du popUp hiérarchique
- ` contenant les champs blob

**C\_ENTIER LONG**(Blob\_Choice;\$SubList)

\$nt:=**Nombre de tables**

**Boucle** (\$i;1;\$nt)

\$exist:=**Faux**

\$nc:=**Nombre de champs**(\$i)

\$sublist:=**Nouvelle liste**

**Boucle** (\$j;1;\$nc)

\$field:=**Champ**(\$i;\$j)

**Si** (**Type**(\$field->)=Est un BLOB )

\$exist:=**Vrai**

**AJOUTER A LISTE**(\$SubList;**Nom du champ**(\$Field);\$j+(\$i\*100))

**Fin de si**

**Fin de boucle**

**Si** (\$exist)

**Si** (**Non**(**Liste existante**(Blob\_Choice)))

Blob\_Choice:=**Nouvelle liste**

**Fin de si**

**AJOUTER A LISTE**(Blob\_Choice;**Nom de la table**(\$i);\$i;\$subList;**Faux**)

**Fin de si**

**Fin de boucle**

**Si** (**Liste existante**(Blob\_Choice))

**REDESSINER LISTE**(Blob\_Choice)

**CHOIX VISIBLE**(\*;"NoBlob";**Faux**)

**Sinon**

**CHOIX VISIBLE**(Blob\_Choice;**Faux**)

**CHOIX VISIBLE**(\*;"NoBlob";**Vrai**)

**Fin de si**

`IMPORTANT : obligatoirement sur chargement du formulaire



` afin de pouvoir gérer l'évènement Sur minuteur

**FIXER MINUTEUR**(10)

Hexa\_txt:=""

Ascii\_txt:=""

OldSel:=""

vH:=""

vD:=""

theSize:=0

theSelect:=0

**INACTIVER BOUTON**(\*;"nav@")

: (**Evenement formulaire**=Sur minuteur )

**Au cas ou**

: (**Dernier objet**=(->Hexa\_Txt))

- ` Gestion de la sélection
- ` dans le texte hexadécimal

**TEXTE SELECTIONNE**(Hexa\_txt;\$deb;\$fin)

\$select:=**Sous chaine**(Hexa\_txt;\$deb;\$fin-\$deb)

**Si** (\$select#OldSel)

**Si** (**Modulo**(\$fin-\$deb;2)#0)

**SELECTIONNER TEXTE**(Hexa\_txt;\$deb;\$deb)

vH:=""

vB:=""

vD:=""

**Sinon**

theSelect:=(((\$deb\5)\*2)+((\$deb%5)\2)+(720\* $\diamond$ cpt)

theEnd:=(((\$fin\5)\*2)+((\$fin%5)\2)+(720\* $\diamond$ cpt)

**Si** (**Sous chaine**(Hexa\_txt;\$deb;1)#" ") & (**Sous chaine**(Hexa\_txt;\$deb;1)#" ")

- ` NOTE : réactiver la ligne SELECTIONNER TEXTE(Ascii\_txt...
- ` dès lors que vous désiriez visualiser la correspondance
- ` dans la zone texte ascii
- ` attention toutefois, car vous perdrez la sélection de la zone hexadécimale
- ` (on ne peut avoir deux focus dans un même formulaire)

` SELECTIONNER TEXTE(Ascii\_txt;theSelect+1;theEnd+1)

vH:=**Remplacer chaine**(**Sous chaine**(Hexa\_txt;\$deb;(\$fin-\$deb));" ";" ")

vD:=Wood\_BLOB ("F\_Base2Dec";vH;16)

vB:=Wood\_BLOB ("F\_Dec2Base";vD;2)

**Sinon**

**SELECTIONNER TEXTE**(Hexa\_txt;\$deb;\$deb)

vH:=""

vB:=""

vD:=""

**Fin de si**

**Fin de si**

oldSel:=\$select

**Fin de si**

**Fin de cas**

: (**Evenement formulaire**=Sur appel extérieur )

## Au cas ou

: (◇MESSAGE="Calc")

  ` Convertisseur de bases décimales

◇MESSAGE=""

◇Out:=2

**COORDONNEES FENETRE**(\$left;\$top;\$right;\$bottom;**Fenetre premier plan**)

awind:=**Creer fenetre**(\$left+9;\$top+74;\$left+9+236;\$top+74+265;2)

**DIALOGUE**(◇ptrTable->"Wood\_Blob")

: (◇MESSAGE="Find")

  ` recherche d'une chaîne

**COORDONNEES FENETRE**(\$left;\$top;\$right;\$bottom;**Fenetre premier plan**)

◇MESSAGE=""

◇Out:=3

awind:=**Creer fenetre**(\$right-246;\$top+74;\$right-10;\$top+74+151;2)

aValue=""

OkFind:=0

**DIALOGUE**(◇ptrTable->"Wood\_Blob")

**Si** (OkFind=1) & (aValue#"" )

**Si** (fHexa=2)

    \$ptr:=->Hexa\_Txt

**Sinon**

    \$ptr:=->Ascii\_Txt

**Fin de si**

    \$p:=**Position**(aValue;\$ptr->)

**SELECTIONNER TEXTE**(\$ptr->,\$p;\$p+**Longueur**(aValue))

**Fin de si**

: (◇MESSAGE="Goto")

  ` sélection d'un offset

**COORDONNEES FENETRE**(\$left;\$top;\$right;\$bottom;**Fenetre premier plan**)

◇MESSAGE=""

◇Out:=4

\$middle:=(\$right-\$left)\2

awind:=**Creer fenetre**(\$middle-123;\$top+74;\$middle+123;\$top+74+118;2)

aValue=""

OkFind:=0

**DIALOGUE**(◇ptrTable->"Wood\_Blob")

**Si** (OkFind=1) & (aValue#"" )

  \$value:=(**Num**(aValue))

    ` calcul du segment

  \$page:=\$value\720

**Si** (\$page#◇cpt)

    ◇cpt:=\$page

    Wood\_BLOB ("Update")

**Fin de si**

    \$value:=( \$value-(720\*◇cpt))\*2

    \$offset:=\$value+( \$value\4)+1

**SELECTIONNER TEXTE**(Hexa\_Txt;\$offset;\$offset+2)

**Fin de si**

: (◇MESSAGE="Print")

```

`impression

FORMULAIRE SORTIE(◊ptrTable->"Wood_Blob")
IMPRIMER ENREGISTREMENT(◊ptrTable->)
awind:=0
◊MESSAGE:=""
Fin de cas

◊Out:=1
FERMER FENETRE(awind)
ALLER A PAGE(1)
FIXER MINUTEUR(10)

Fin de cas
Sinon
Si(Page formulaire courante=1)
  FIXER MINUTEUR(0)
  ALLER A PAGE(◊Out)
Fin de si
Fin de si

```

## Appels

Deux types d'appels sont générés par les objets du formulaire :

- Appel à la méthode principale (changement de segment, conversion, etc.) du type :

**ENREGISTREMENT SUIVANT**(**Table**(◊Table)->)  
Wood\_BLOB ("Update")

- Appel à la méthode du formulaire (recherche, convertisseur, etc.) du type :

◊MESSAGE:="Calc"  
**APPELER PROCESS**(**Numero du process courant**)

## Initialisation variables :

A inclure dans vos projets, pour utilisation en mode Utilisation directe :

```

◊Out:=1

`===== Votre table interface =====
◊ptrTable:=->[Table 1]

`=====

```

Dans la base test ce code est situé dans la méthode base : Sur ouverture.

Conversions :

Pour chaque type de conversion de base numérique, nous passons d'abord la valeur à convertir en base décimale, puis, pour chacune des bases numériques, nous convertissons cette valeur décimale dans cette base.

Exemple de conversion d'une chaîne hexadécimale vers les autres bases décimales :

```
vD:=Wood_BLOB ("F_Base2Dec";Self->;16)
v4D:=Chaine(Num(vD);"&x")
vB:=Wood_BLOB ("F_Dec2Base";vD;2)
vT:=Wood_BLOB ("F_Dec2Base";vD;3)
vQ:=Wood_BLOB ("F_Dec2Base";vD;5)
vO:=Wood_BLOB ("F_Dec2Base";vD;8)
```

## Note

---

Pour une intégration plus légère, l'interface peut être avantageusement remplacée par une interface "standard". Et si vous en avez le besoin, il vous sera facile de transformer ce module en éditeur de blob.

L'interface est celle du thème " TreasureChest " du navigateur NeoPlanet™.  
<http://www.neoplanet.com>