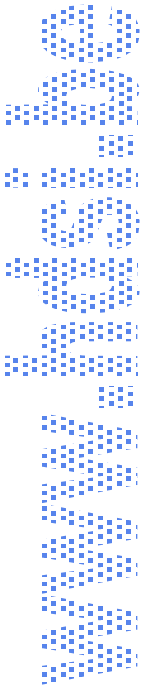
HDSI

HDSI
Rue du Centre, 22
4140 SPRIMONT
Tél. : 04.341.41.05
info@hdsi.be

Standards de programmation HDSI

Version 1.5



HDSI
Rue du Centre, 22
4140 SPRIMONT
Tél. : 04.341.41.05
info@hdsi.be

Visual Basic

Généralités

1. La tabulation doit être réglée sur 2 espaces
2. Les fichiers code source doivent être « *Option Explicit* »

Les subs

L'utilisation des SUB / END SUB est proscrite et sera, à terme, complètement remplacée par des FUNCTION / END FUNCTION

La valeur par défaut du retour de la fonction doit être affectée en tout début de code de la fonction.

Les fonctions, sauf cas particulier, doivent toujours retourner un Boolean. Une valeur de retour égale à True indique que la fonction s'est terminée avec succès. En cas d'échec elle doit retourner False.

Exception :
Les seuls SUB / END SUB autorisés sont ceux autogénérés par le Visual Basic pour les évènements.

Les fonctions

Les noms de fonction doivent représenter au mieux le but de la fonction. Une description sera systématiquement créée en commentaire de la fonction. L'utilisation de Vbdocman est obligatoire[1] afin de pouvoir générer de manière automatique la documentation du code.

Tout appel de fonction devra tester le retour de la fonction appelée et réagir en conséquence.

[1] Sauf en VBA qui ne permet pas d'intégrer cet outil.

HDSI
Rue du Centre, 22
4140 SPRIMONT
Tél. : 04.341.41.05
info@hdsi.be

L'indentation et la mise en forme

1. Le nom de la fonction ou de la procédure ainsi que sa terminaison doivent se trouver contre la marge.
2. Le code de la fonction ainsi que les déclarations de variable locale doivent être décalé d'une tabulation.
3. Les tests conditionnels et leur terminaison ainsi que les divers types de boucles et leur terminaison doivent être au même niveau d'indentation. Le code inclus doit être indenté.
4. Des lignes vides doivent être placées entre les diverses sections de la fonction ou de la procédure. Par exemple entre la déclaration des variables globales et le début du code, ou entre deux parties de code à distinguer.

Exemple :

```
Function MaFonction( iParam1 As Integer, iParam2 As Integer ) as boolean
    Dim strMessage As String

    MaFonction = True

    If iParam1 < iParam2 Then
        strMessage = iParam1 & '' est plus petit que '' & iParam2
    Else
        If iParam1 = iParam2 Then
            strMessage = iParam1 & '' est égal à '' & iParam2
        Else
            strMessage = iParam1 & '' est plus grand que '' & iParam2
        End If
    End If

    MsgBox strMessage, vbOk

End Function
```

HDSI
Rue du Centre, 22
4140 SPRIMONT
Tél. : 04.341.41.05
info@hdsi.be

Les variables

Les variables doivent toutes être nommées d'une manière claire. La seule exception admise concerne les variables utilisées dans les boucles *for…next* qui seront choisies dans i, j, k, l en fonction de leur niveau de profondeur dans le code (boucles imbriquées). Deux boucles de même niveau devront utiliser la même variable d'indice.

Exemple :

```
Dim i as Integer, j as Integer

For i=0 to 10 ' Boucle de premier niveau : profondeur 1
    For j=0 to 5 ' Boucle de second niveau : profondeur 2
        …
    Next j
Next i
…
For i=0 to 20 ' Nouvelle boucle de premier niveau : profondeur 1
    …
Next i
```

Le préfixe sera toujours en minuscule, les divers mots suivants le préfixe commenceront tous par une majuscule, le reste de chaque mot est en minuscule.

Exemple :

```
Dim strNomDeFichier as String
```

HDSI
Rue du Centre, 22
4140 SPRIMONT
Tél. : 04.341.41.05
info@hdsi.be

Type de variable	Préfixe à utiliser	Exemple
Entier court (integer)	i…	iCompteur
Entier long (long)	l…	lValeur
Réel double (double)	d…	dDistance
Chaîne de caractère (string)	str…	strFilename
Tableau d'entier (integer(…))	ai…	aiIndex(10)
Tableau de long (long(…))	al…	alValeur(10)
Tableau de double (double(…))	ad…	adDistance(10)
Tableau de chaîne de caractère (string(…))	as…	asFilename(10)
Objet	o…	oBEEDocs

Les contrôles

Le principe général de dénomination est le même que pour les variables, à savoir : le préfixe en minuscule ensuite les divers mots de la description du contrôle.

Exemple :

frmBoiteDeDialogue

Type de contrôle	Préfixe à utiliser	Exemple
Picture	pct…	pctImage
Label	lbl…	lblMessage
TextBox	txt…	txtValeurAngle
Frame	fra…	fraModeCalcul
Bouton de commande	cmd…	cmdQuitter
Checkbox	chk…	chkAfficherRésultat
OptionButton	opt…	optMode1
ComboBox	cmb…	cmbProducteurs
ListBox	lst…	lstProjets
Image	img…	imgImage

HDSI
Rue du Centre, 22
4140 SPRIMONT
Tél. : 04.341.41.05
info@hdsi.be

Visual C / MDL

Les fonctions

Les noms de fonction doivent représenter au mieux le but de la fonction. Une description sera systématiquement créée en commentaire de la fonction.

Les fonctions, sauf cas particulier, devront retourner un boolean. Une valeur égale à TRUE indiquera que la fonction s'est terminée avec succès, une valeur de retour égale à FALSE indique que la fonction n'a pu être se terminer avec succès.

Les variables locales

Les variables doivent toutes être nommées d'une manière claire. La seule exception admise concerne les variables utilisées dans les boucles *for…next* qui seront choisies dans i, j, k, l en fonction de leur niveau de profondeur dans le code (boucles imbriquées). Deux boucles de même niveau devront utiliser la même variable d'indice.

Exemple :

```
int i, j ;

for(i=0 ; i < 10 ; i++) ‘ Boucle de premier niveau : profondeur 1
{
    for( j=0 ; j < 5 ; j++) ‘ Boucle de second niveau : profondeur 2
    {
        …
    }
}
…
for(i=0 ; i < 10 ; i++) ‘ Nouvelle boucle de premier niveau : profondeur 1
{
    …
}
```

Le préfixe sera toujours en minuscule, les divers mots suivants le préfixe commenceront tous par une majuscule, le reste de chaque mot est en minuscule.

Exemple :

```
char chNomDeFichier[MAXFILELENGTH] ;
```

HDSI
Rue du Centre, 22
4140 SPRIMONT
Tél. : 04.341.41.05
info@hdsi.be

Type de variable	Préfixe à utiliser	Exemple
Entier court (int)	i…	iCompteur
Entier long (long)	l…	lValeur
Réel double (double)	d…	dDistance
Chaîne de caractère (char)	ch…	strFilename
Tableau d'entier (integer[…])	ai…	aiIndex(10)
Tableau de long (long[…])	al…	alValeur(10)
Tableau de double (double[…])	ad…	adDistance(10)
Pointeur	p + préfixe du type + …	*piCompteur