

Visual Basic Entwicklungs-Standards

einmal etwas anders.

Wolfgang Nigischer

Inhalt

- Einführung in die von Microsoft erstellten Namens-Konventionen bei Kontroll-Elementen
- Grundlegende Überlegungen über die gegebenen Namens-Konventionen hinaus weitere Teile des Source-Codes zu standardisieren.

Bereits seit der ersten Version von Visual Basic hat man sich bemüht, durch die Deklarationen in der Datei CONSTANT.TXT den Code lesbarer zu gestalten. Seit der Version 2.0 von Visual Basic haben sich die Namens-Konventionen von Kontroll-Elementen durchgesetzt (MS Knowledge-Base: Q110264). Teile dieses Q110264 Artikels möchte ich hier nur der Vollständigkeit halber anführen. Die Variablen-Deklarationen, die optische Gestaltung des Codes und die Vergabe der Contrex ID bei der Windows Hilfe stammen von :

IHS Professional Services Visual Basic™
Application Development
Standards and Guidelines ⁴

MCS Standard für Objekt-Namen

Präfix	Objekt-Typ	Beispiel
ani	Animation button	aniMailBox
bed	Pen Bedit	bedFirstName
cbo	Combo box and drop down list box	cboEnglish
chk	Checkbox	chkReadOnly
clp	Picture clip	clpToolBar
cmd (3d)	Command button (3D)	cmdOk (cmd3dOk)
com	Communications	comFax
ctr	Control (when specific type unknown)	ctrCurrent
dat	Data control	datBiblio
dir	Directory list box	dirSource
dlg	Common dialog control	dlgFileOpen
drv	Drive list box	drvTarget
fil	File list box	filSource
frm	Form	frmEntry
fra (3d)	Frame (3d)	fraStyle (fra3dStyle)
gau	Gauge	gauStatus
gpb	Group push button	gpbChannel
gra	Graph	graRevenue
grd	Grid	grdPrices
hed	Pen Hedit	hedSignature
hsb	Horizontal scroll bar	hsbVolume
img	Image	imgIcon
ink	Pen Ink	inkMap
key	Keyboard key status	keyCaps
lbl	Label	lblHelpMessage
lin	Line	linVertical
lst	List box	lstPolicyCodes
mdi	MDI child form	mdiNote

⁴ Author: Joe Paquette

IHS Professional Services
19311 -142nd Place SE
Renton, Washington 98058-9446
(206) 630-9866
CIS: 72341,2665
Revised: August 21, 1994

mpm	MAPI message	mpmSendMessage
mps	MAPI session	mpsSession
mci	MCI	mciVideo
mnu ⁵	Menu	mnuFileOpen
opt (3d)	Option Button (3d)	optRed (opt3dRed)
ole	OLE control	oleWorksheet
out	Outline control	outOrgChart
pic	Picture	picVGA
pnl3d	3d Panel	pnl3d
rpt	Report control	rptQtr1Earnings
shp	Shape controls	shpCircle
spn	Spin control	spnPages
txt	Text Box	txtLastName
tmr	Timer	tmrAlarm
vsb	Vertical scroll bar	vsbRate

⁶

Objekt-Namenskonventionen für Datenbank-Objekte

Object Naming Convention for Database Objects

Präfix	Objekt-Typ	Beispiel
db	ODBC Database	dbAccounts
ds	ODBC Dynaset object	dsSalesByRegion
fdc	Field collection	fdcCustomer
fd	Field object	fdAddress
ix	Index object	ixAge
ixc	Index collection	ixcNewAge
qd	QueryDef object	qdSalesByRegion
qry (suffix)	Query (see NOTE)	SalesByRegionQry
ss	Snapshot object	ssForecast
tb	Table object	tbCustomer
td	TableDef object	tdCustomers

Fehlende Controls

Bei den Namens-Konventionen in VB 4.0 fehlen einige Controls, für die sich folgender Namens-Standard herauskristallisieren dürfte⁷.

Präfix	Objekt-Typ
chk	3D check box (same prefix as Checkbox)
cmd	3D command button (same prefix as Command Button)
fra	3D frame (same prefix as Frame)
gra	Graph
iml	ImageList
lsv	ListView

⁵ Bei den Menü`s wird zum Präfix „mnu“ gleich anschließend der Menü - Titel in der verwendeten chronologischen Reihenfolge vergeben.

Help.Contents	mnuHelpContents
File.Open	mnuFileOpen
Format.Character	mnuFormatCharacter
File.Send.Fax	mnuFileSendFax
File.Send.Email	mnuFileSendEmail

⁶ Anmerkung:

Für Controls von drittherstellern, z.B. Microhelp wird eine vierte Stelle im Präfix eingeführt.

Beispielsweise: Um Verwechslungen mit dem Command Button von Microhelp und dem von VB mitgelieferten zu vermeiden, könnte man „cmd“ für das Original - VB und „cmd“ für das Control - Button von der Fa. Microhelp verwenden

⁷ Diese Liste ist noch inoffiziell! Anregungen nimmt Herr Marc Rochkind (CIS ID.:72540,2273) gerne entgegen.

msk	Masked edit
opt	3D option button (same prefix as Option Button)
otl	Outline
prg	ProgressBar
rch	RichTextBox
sld	Slider
sst	SSTab
sts	Status
tbr	ToolBar
tbs	TabStrip
trv	TreeView

Namenskonventionen für Variablen

Diese setzen sich wie folgt zusammen:⁸

Bei Variablen besteht der Präfix aus zwei klein geschriebenen Buchstaben, gefolgt von den eigentlichen (aussagekräftigen) Variablennamen.

Der erste Buchstabe des Präfix gibt Auskunft über den Gültigkeitsbereich der Variablen:

Global	g
Module	m
Form	f
Local	l
Parameter	p (Argumente in Prozeduren und Funktionen)

Der zweite Buchstabe gibt Auskunft über den Datentyp:

Integer	i
Boolean	b (bis VB 3.0 integer)
Long	l
Single	s
Double	d
Currency	c
String (text)	t
Variant	v
User Defined Type	u (user defined type structures)

Beispiele:

Global string (text) variable	gtAppName
Form level integer variable	fiObjectCount
Module lever single variable	msPercentComplete
Global variant variable	gvSystemObject
Form level currency variable	fclInvoiceTotal
Global user defined type	guUserDefaults
Local integer variable	liCounter
Parameter control	pctlSourceControl

Optische Gestaltung des Code

Sämtlicher Code zwischen Anfang und Ende einer Prozedur soll mindestens einen Tab-Stop⁹ eingezogen sein.

Bei If...EndIf, Select Case...End Select, For...Next, und Do...Loop ist eine weitere Einrückung von einem weiteren Tab-Stop vorzunehmen. Zusätzlich sollte man noch durch die entsprechende Anordnung von Leerzeilen logische Abschnitte schaffen.

⁸ Hierbei möchte ich den Entwurf von IHS - Standards anführen, da mir dieser etwas besser gefällt. Die MS Variante kann jeder Internet oder in CompuServe Knowledge - Base unter: Q110264 finden.

⁹ Standardeinstellung: 1 Tabulator = 4 Zeichen

Wenn Sie ein neues Modul entwerfen, so sollte nach der ersten Beschreibung der Eintrag Option Explicit sein.

Das könnte dann so aussehen:

Module Header:

```
' WorkFile:      MODULE.BAS
' Copyright:    IHS Professional Services, 1994
' Created:     15 Mar 94
' Author:      Joe Programmer
' Description: This is a brief description of
'              MODULE.BAS. One or two lines should
'              be sufficient.
```

Option Explicit

```
' =====
' Global Level Constant Declaration Section
' =====
Global Const QQ_OPEN = 0      ' Open
Global Const QQ_CLOSED = 1   ' Closed
' =====
' Type Definition Declaration Section
' =====
Type guMyTypeDef              ' My Type Def
tStringVar1 As String        ' Variable 1
tStringVar2 As String        ' Variable 2
End Type
' =====
' API/DLL Procedure Declaration Section
' =====
Declare Function giReadIniFile Lib "XYZ.DLL" Alias_
"ReadINI" (ByVal FullPath As String, ByVal FileName As_
String) As Integer
```

```
' =====
' Global Level Variable Declaration Section
' =====
Dim guUserType As UserDefinedType 'User defined type
Dim gtString As String            'String variable
```

```
' =====
' Module Level Constant Declaration Section
' =====
Const MM_ON = 1                'On
Const MM_OFF = 2               'Off
```

```
' =====
' Module Level Variable Declaration Section
' =====
Dim muUserType as UserDefinedType 'User defined type
Dim mtString as String            'String variable
```

Bei den Prozedur-Headern wäre dann folgender Vorschlag angebracht:

```
Sub gProcName (ByVal ptString As String, pbLogical As Integer)
' Description: This is a brief description of a Sub
'              procedure.
' Pass:       ptString String to convert
'              pbLogical Logical flag
' Return:     N/A
```

```
' Date       Developer      Comments
' 03/15/94   J. Programmer   Initial creation
```

```
' =====
' Local Constant/Vari able Decl arati on Secti on
' =====
Const P_CON = 0                ' Constant
```

```
Dim luUserType as UserDefinedType 'User defined
Dim ltString            'String var
```

```
Function giFuncName (ByVal ptString As String,
pbLogical As Integer) as Integer
' Description: This is a brief description of a
'              Functi on procedure.
' Pass:       ptString String to convert
'              pbLogical Logical flag
' Return:     gi FuncName Function return value
```

```
' Date       Developer      Comments
' 03/15/94   J. Programmer   Initial creation
```

```
' =====
' Local Constant/Vari able Decl arati on Secti on
```

```
' =====
Const F_CON = 0          ' Constant
Dim lUserType as UserDefi nedType ' User defi ned
Dim l tString           ' String var
```

In-Line-Comments

In-Line-Comments beschreiben den entsprechenden Teil des Source-Codes, wobei man nach Möglichkeit vorher beschreiben sollte, was hinterher im Source-Code passiert.

Auch sollten die Zeilen nicht zu lange werden, damit man die Bemerkungen ohne Verwendung der Horizontalen Scroll-Balken in einem sehen kann¹⁰.

```
' Comment goes here. Use as many lines as necessary,
' being sure to start a new line so the comment is
' visible in a typically sized code window. One
' formatted comment line should precede and follow
' the actual comment.
```

Nun zu den einzelnen Modulen

Im anschließenden Kapitel geht es um eine Übersichtliche Aufteilung von Modulen (*bas-Dateien¹¹) von Applikations-Spezifischen Deklarationen:

<EXENAME>. BAS

Dieses Modul, das den selben Namen, wie die fertige Applikation hat, beinhaltet die anwendungsspezifischen Konstanten.

WI NAPI . BAS

Dem Namen nach nicht schwer zu erraten ist diese Datei für die Windows-Deklarationen (APIs) zuständig.

Auch API-spezifische Sus' s und Funktionen wären hier richtig aufgehoben.

GLOBAL. BAS

Hier findet man die applikationsweit gültigen Variablen und generell gültige Prozeduren

Last but not least einige Worte zur Kontext sensitiven Hilfe:

IHS Professional Services schlägt hier folgende Vorgehensweise vor:

Die Help Contrex ID besteht aus einer Nummer, die sich aus 6 Zahlen zusammensetzt.

Menü	1 B B P P S S
1	Menü-Identifikation
B B	Menü-Option (beginnend mit 01 beim linken oberen Menüpunkt)
P P	serielle Menü-Option (beginnend mit 02 pro Inkrement)
S S	Menü-Sup-Option (beginnend mit 02 pro Inkrement)

Objekte:	2 F F F C C
2	Objekt Identifikation
F F F	Seriennummer der Form
C C	Seriennummer des Control-Elements (00=Form, 01= Control 1, 02=Control 2....)

Alias:	999XXX
999	= Alias Identifikation
XXX	Alias Serien nr.

Seasoned professional

```
=====
#include <iostream.h>
#include <string.h>

class string
{
private:
int size;
char *ptr;

public:
string() : size(0), ptr(new char('\0')) {}

string(const string &s) : size(s.size)
{
ptr = new char[size + 1];
strcpy(ptr, s.ptr);
}

~string()
{
delete [] ptr;
}

friend ostream &operator <<
(ostream &, const string &);
string &operator=(const char *);
};

ostream &operator<<
(ostream &stream, const string &s)
{
return(stream << s.ptr);
}

string &string::operator=(const char *chrs)
{
if (this != &chrs)
{
delete [] ptr;
size = strlen(chrs);
ptr = new char[size + 1];
strcpy(ptr, chrs);
}
return(*this);
}

int main()
{
string str;

str = "Hello World";
cout << str << endl;

return(0);
}
```

¹⁰ ca. 45 Anschläge (das ist ungefähr die Breite des Code - Fensters die dessen ersten Darstellung)

¹¹ Ein Klassenstandard für VB 4.0 wird von der Firma IHS noch erarbeitet.