VBXpress Demo-Controls

Diese Online-Hilfe gehört zu den Demo-VBX-Dateien, die mit dem Control-Designer VBXpress erstellt wurden. Die Demo-Controls zum Ausprobieren können Sie in Ihren Programmen frei verwenden. Sie werden unverbindlich und ohne irgendwelche Verpflichtungen seitens Marquis Computing oder Zoschke Data zur Verfügung gestellt.

Zum effizienten Ausprobieren finden Sie in dieser Hilfe Informationen zu den Eigenschaften, Ereignisse und Methoden der mit VBXpress erstellbaren VBX-Controls.

Impressum

Zoschke Data GmbH
Bahnhofstraße 3
D-24217 Schönberg/Holstein

Tel. 04344/6166 Fax: 04344/6162

CompuServe E-Mail: 71340,2051

Version

Hilfedatei zu VBX-Controls erzeugt mit VBXpress Version 2.28 - Stand: 8/94.

VBX-Ereignisse

Für die mit VBXpress erstellten Controls stehen folgende Ereignisse zur Verfügung:

ButtonOver1	ButtonPress 2	Click	DblClick
DragDrop1	DragOver	GotFocus 4	KeyDown
KeyPress	KeyUp	LostFocus 4	MouseDown
MouseMove	MouseUp	SliderChange 3	

1 Nur in Visual Basic verfügbar
2 Nicht für Schieberegler verfügbar
3 Nur auf Schieberegler anwendbar.
4 Optional (evtl. nicht vorhanden).

Weitere Informationen über die Verwendung von Ereignissen finden Sie gegebenenfalls in den Handbüchern des von Ihnen verwendeten Windows-Entwicklungssystems.

ButtonOver-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn die Maus ein Control überquert.

Syntax

Visual Basic

Sub MyCtrl_**ButtonOver** (Button **As Integer**)

Anmerkungen

Button ist die Nummer jenes Buttons im Control, über dem sich die Maus gerade befindet.

ButtonPress-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn der Anwender einen Control-Button der für das Control gewählten Aktionsart entsprechend betätigt.

Syntax

Visual Basic

Sub MyCtrl_**ButtonPress** (Button **As Integer**, Mouse **As Integer**)

Anmerkungen

Button ist die Nummer des im Control betätigten Buttons.

Mouse meldet die gedrückte Maustaste (1 oder 2).

Click-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn eine Maustaste über einem Control gedrückt und anschließend wieder losgelassen wird.

Syntax

Visual Basic

Sub MyCtrl_**Click** ()

DblClick-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn auf dem Control mit der Maus ein Doppelklick durchgeführt wird.

Syntax

Visual Basic

```
Sub MyCtrl_DblClick ()
```

DragDrop-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn eine Ziehen-und-Ablegen-Operation abgeschlossen worden ist, d.h. wenn ein Steuerelement über ein Control gezogen und und dann die Maustaste losgelassen wurde.

Syntax

Visual Basic

Sub MyCtrl_**DragDrop** (Quelle **As Control**, X **As Single**, Y **As Single**)

Anmerkungen

Quelle ist das abgelegte Control

X und Y sind die Maus-Koordinaten zum Zeitpunkt des Ablegens.

DragOver-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn eine Ziehen-und-Ablegen-Operation erfolgt. läßt sich verfolgen, wann der Mauszeiger ein gültiges Ziel erreicht oder verläßt bzw. sich direkt darüber befindet. Die Position des Mauszeigers bestimmt, auf welches Zielobjekt dieses Ereignis zutrifft.

Syntax

Visual Basic

Sub MyCtrl_**DragOver** (Quelle **As Control**, X **As Single**, Y **As Single**, Status **As Integer**)

Anmerkungen

Quelle ist das gezogene Control

X und Y sind die Maus-Koordinaten innerhalb des Ziel-Steuerelementes.

Status ist der Übergangszustand des gezogenen Steuerelementes im Verhältnis zu

einer Ziel-Form oder einem Ziel-Steuerelement:

0 - Eintreten (Quell-Steuerelement wird in den Bereich eines Zieles gezogen)

1 - Verlassen (Quell-Steuerelement wird aus dem Bereich eines Zieles

herausgezogen)

2 - Darüber (Quell-Steuerelement wurde von einer Position im Ziel an eine andere

Position gezogen)

GotFocus-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn das Control per Maus oder Tastatur den Eingabefokus bekommt.

Syntax

Visual Basic

Sub MyCtrl_**GotFocus** ()

Anmerkungen

VBX-Controls des vorliegenden Typs können aus einer Vielzahl von Buttons bestehen; da stets das gesamte Control den Tabulator bekommt, erscheint kein Fokus-Rechteck. Dieses Verhalten entspricht z. B. den Buttons der VB- oder Winword-Werkzeugleisten.

KeyDown-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn das Control den Fokus hat und eine Taste gedrückt wird; es liefert die dabei gedrückte(n) Taste(n).

Syntax

Visual Basic

Sub MyCtrl_**KeyDown** (KeyCode **As Integer**, Shift **As Integer**)

Anmerkungen

KeyCode liefert den Code der gedrückten Taste, z. B. KEY_F1 (F1-TASTE) und KEY_HOME (POS1-TASTE). Verwenden Sie für Tastencodes die in der Visual-Basic-Datei CONSTANT.TXT enthaltenen Konstanten.

Shift gibt bitkodiert den Status der UMSCHALTTASTE, STRG-TASTE und ALT-TASTE zur Zeit des Ereignisses an:

UMSCH-TASTE - Bit 0
STRG-TASTE - Bit 1
ALT-TASTE - Bit 2

Je nach gedrückter Tastenkombination können einige, alle oder keine Bits gesetzt sein. Wird beispielsweise die Tastenkombination STRG + ALT gedrückt, hat Shift den Wert 6.

Um die Ansi-Zeichen gedrückter Tasten auszuwerten, verwenden Sie das KeyPress-Ereignis.

KeyPress-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn das Control den Fokus hat und eine Taste gedrückt wird; es liefert den Ansi-Code der gedrückten Taste.

Syntax

Visual Basic

Sub MyCtrl_**KeyPress** (KeyAnsi **As Integer**)

Anmerkungen

KeyAnsi liefert den Ansi-Code der gedrückten Taste.

KeyUp-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn das Control den Fokus hat und eine Taste gedrückt und wieder losgelassen wird; es liefert die freigegebene(n) Taste(n).

Syntax

Visual Basic

Sub MyCtrl_**KeyUp** (KeyCode **As Integer**, Shift **As Integer**)

Anmerkungen

KeyCode liefert den Code der losgelassenen Taste, z. B. KEY_F1 (F1-TASTE) und KEY_HOME (POS1-TASTE). Verwenden Sie für Tastencodes die in der Visual-Basic-Datei CONSTANT.TXT enthaltenen Konstanten.

Shift gibt bitkodiert den Status der UMSCHALTTASTE, STRG-TASTE und ALT-TASTE zur Zeit des Ereignisses an:

UMSCH-TASTE - Bit 0
STRG-TASTE - Bit 1
ALT-TASTE - Bit 2

Je nach gedrückter Tastenkombination können einige, alle oder keine Bits gesetzt sein. Wird beispielsweise die Tastenkombination STRG + ALT gedrückt, hat Shift den Wert 6.

LostFocus-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn das Control den Eingabefokus verliert.

Syntax

Visual Basic

Sub MyCtrl_**LostFocus** ()

MouseDown-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn über dem Control eine der Maustasten gedrückt wird.

Syntax

Visual Basic

Sub MyCtrl_**MouseDown** (Button **As Integer**, Shift **As Integer**, X **As Single**, Y **As Single**)

Anmerkungen

Button liefert bitkodiert die gedrückte Maustaste.

Linke Maustaste - Bit 0
Rechte Maustaste - Bit 1
Mittlere Maustaste - Bit 2

Diese Bits stellen die Werte 1, 2 bzw. 4 dar. Es ist jeweils nur eines dieser Bits gesetzt, das anzeigt, welche Maustaste das Ereignis verursacht hat (z. B. linke Maustaste = 1).

Shift gibt bitkodiert den Status der UMSCHALTTASTE, STRG-TASTE und ALT-TASTE zur Zeit des Ereignisses an:

UMSCH-TASTE - Bit 0
STRG-TASTE - Bit 1
ALT-TASTE - Bit 2

Je nach gedrückter Tastenkombination können einige, alle oder keine Bits gesetzt sein. Wird beispielsweise die Tastenkombination STRG + ALT gedrückt, hat Shift den Wert 6.

X und Y sind die Maus-Koordinaten zum Zeitpunkt des Ereignisses.

MouseMove-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn die Maus über ein Control bewegt wird.

Syntax

Visual Basic

Sub MyCtrl_**MouseMove** (Button **As Integer**, Shift **As Integer**, X **As Single**, Y **As Single**)

Anmerkungen

Button liefert bitkodiert die dabei gedrückte Maustaste:

Linke Maustaste - Bit 0
Rechte Maustaste - Bit 1
Mittlere Maustaste - Bit 2

Diese Bits stellen die Werte 1, 2 bzw. 4 dar. Je nach gedrückten Maustasten können einige, alle oder keine dieser drei Bits gesetzt sein (z. B. linke Maustaste = 1).

Shift gibt bitkodiert den Status der UMSCHALTTASTE, STRG-TASTE und ALT-TASTE zur Zeit des Ereignisses an:

UMSCH-TASTE - Bit 0
STRG-TASTE - Bit 1
ALT-TASTE - Bit 2

Je nach gedrückter Tastenkombination können einige, alle oder keine Bits gesetzt sein. Wird beispielsweise die Tastenkombination STRG + ALT gedrückt, hat Shift den Wert 6.

X und Y sind die Maus-Koordinaten zum Zeitpunkt des Ereignisses.

MouseUp-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn über dem Control eine der Maustasten freigegeben wird.

Syntax

Visual Basic

Sub MyCtrl_**MouseUp** (Button **As Integer**, Shift **As Integer**, X **As Single**, Y **As Single**)

Anmerkungen

Button liefert bitkodiert die losgelassene Maustaste.

Linke Maustaste - Bit 0
Rechte Maustaste - Bit 1
Mittlere Maustaste - Bit 2

Diese Bits stellen die Werte 1, 2 bzw. 4 dar. Es ist jeweils nur eines dieser Bits gesetzt, das anzeigt, welche Maustaste das Ereignis verursacht hat (z. B. linke Maustaste = 1).

Shift gibt bitkodiert den Status der UMSCHALTTASTE, STRG-TASTE und ALT-TASTE zur Zeit des Ereignisses an:

UMSCH-TASTE - Bit 0
STRG-TASTE - Bit 1
ALT-TASTE - Bit 2

Je nach gedrückter Tastenkombination können einige, alle oder keine Bits gesetzt sein. Wird beispielsweise die Tastenkombination STRG + ALT gedrückt, hat Shift den Wert 6.

X und Y sind die Maus-Koordinaten zum Zeitpunkt des Ereignisses.

SliderChange-Ereignis

Beschreibung

Dieses Ereignis tritt auf, wenn sich der Wert (Value) eines Schiebereglers ändert.

Syntax

Visual Basic

Sub MyCtrl_**SliderChange** (Value **As Long**)

Anmerkungen

Der minimal bzw. maximal erreichbare Wert wird durch die Einstellung der Eigenschaften MinVal und MaxVal bestimmt.

VBX-Eigenschaften

Für die mit VBXpress erstellten Controls stehen folgende Eigenschaften zur Verfügung:

About	BodyClick4 5	BorderStyle	BtnCnt2 5
BtnPtr2	BtnState2	Caption	Clicks4 5
Cols2 5	DragIcon1	DragMode1	Enabled
Float5	hBMP	hDC	hWnd
Index1	Left	MaxVal 4	MinVal 4
MousePointer	Name3	Parent1	Rows2 5
Style5	TabIndex	TabStop	Tag
Top	Value 4	Visible 5	

1 Nur in Visual Basic verfügbar.
2 Nicht für Schieberegler verfügbar.
3 Identisch mit der CtlName-Eigenschaft in Visual Basic 1.0 und Visual C++.
4 Nur auf Schieberegler anwendbar.
5 Optional (evtl. nicht vorhanden).

Weitere Informationen über die Verwendung von Eigenschaften finden Sie gegebenenfalls in den Handbüchern des von Ihnen verwendeten Windows-Entwicklungssystems.

About-Eigenschaft

Beschreibung

Stellt für das Control ein systemmodales Fenster zur Anzeige von Versions- und Copyright-Informationen zur Verfügung.

Anmerkungen

Die About-Informationen können vom Benutzer des VBX-Controls nicht verändert werden.

BodyClick-Eigenschaft

Beschreibung

Setzt oder liefert die Information, ob sich der Schieber durch Anklicken des Schiebereglerkörpers vor oder hinter dem Schieber bewegen lassen soll.

Syntax

Visual Basic

[form].MyCtrl.**BodyClick**[= {**True | False**}]

Visual C++

pMyCtrl->**GetNumProperty(BodyClick)**

pMyCtrl->**SetNumProperty(BodyClick,** {**True | False**}**)**

Anmerkungen

Einstellung	Beschreibung
False	Anklicken des Schiebereglers nicht auswerten (Voreinstellung).
True	Anklicken des Schiebereglers auswerten.

Datentyp	Integer (Boolean)

BorderStyle-Eigenschaft

Beschreibung

Legt den Rahmenstil fest oder liefert dessen Einstellung.

Syntax

Visual Basic

[form].MyCtrl.**BorderStyle**[= Wert%]

Visual C++

pMyCtrl->0GetNumProperty(BorderStyle)
pMyCtrl->0SetNumProperty(BorderStyle, **Wert**)

Anmerkungen

Für Wert% sind folgende Einstellungen möglich:

Einstellung	Beschreibung
0	Kein Rahmen
1	Starker Rahmen

Datentyp Integer (enumeriert)

BtnCnt-Eigenschaft

Beschreibung

Liefert die Anzahl der Buttons, über die das Control verfügt (nicht gültig für Schieberegler).

Syntax

Visual Basic

[form].MyCtrl.**BtnCnt**

Visual C++

pMyCtrl->**GetNumProperty(BtnCnt)**

Anmerkungen

Dies ist eine Nur-Lesen-Eigenschaft.

Datentyp	Integer (enumeriert)

BtnPtr-Eigenschaft

Beschreibung

Setzt oder liefert den internen Zeiger auf den Button einer Werkzeugleiste (Toolbar) oder Werkzeugpalette.

Syntax

Visual Basic

[form].MyCtrl.**BtnPtr**[= Wert%]

Visual C++

pMyCtrl->**GetNumProperty(BtnPtr)**

pMyCtrl->**SetNumProperty(BtnPtr,** Wert**)**

Anmerkungen

Für Wert% sind folgende Einstellungen möglich:

1-n	(Voreinstellung = 1)

n ist die Anzahl der Buttons (Maximalwert für Wert%). Der Zeiger wird zur Angabe benutzt, auf welchen Button eines Controls mit mehreren Buttons sich das Lesen und programmgesteuerte Setzen des Button-Status mit der BtnState-Eigenschaft beziehen soll.

Datentyp	Integer

BtnState-Eigenschaft

Beschreibung
Setzt oder liefert den aktuellen Status des Buttons, auf den zuvor der Zeiger mit Hilfe der BtnPtr-Eigenschaft eingestellt wurde.

Syntax
Visual Basic
[form].MyCtrl.**BtnState**[= Wert%]

Visual C++
pMyCtrl->**GetNumProperty(BtnState)**
pMyCtrl->**SetNumProperty(BtnState,** Wert**)**

Anmerkungen
Für Wert% sind folgende Einstellungen möglich:

Einstellung	Beschreibung
0	UNGEDRÜCKT (UP)/FREIGEGEBEN
-1	GEDRÜCKT (DOWN)/FREIGEGEBEN
1	GESPERRT

Datentyp Integer (enumeriert)

Wollen Sie das komplette Control sperren oder freigeben, verwenden Sie die Enabled-Eigenschaft.

Caption-Eigenschaft

Beschreibung

Setzt oder liefert den Text, der in der Titelleiste eines verschiebbaren Controls angezeigt wird.

Syntax

Visual Basic

[form].MyCtrl.**Caption**[= text$]

Visual C++

pMyCtrl->**GetStrProperty(Caption)**
pMyCtrl->**SetStrProperty(Caption,** text**)**

Anmerkungen

Die Titelleiste wird nur dann angezeigt, wenn die Float-Eigenschaft auf True eingestellt ist.

Datentyp	String

Clicks-Eigenschaft

Beschreibung

Setzt oder liefert die Information, ob ein Schieberegler-Control bei der Wertänderung durch Bewegen des Schiebers ein akustisches Signal (Klickgeräusch) geben soll.

Syntax

Visual Basic

[form].MyCtrl.**Clicks**[= {**True | False**}]

Visual C++

pMyCtrl->**GetNumProperty(Clicks)**
pMyCtrl->**SetNumProperty(Clicks,** {**True | False**}**)**

Anmerkungen

Einstellung	Beschreibung
False	Kein Klickgeräusch zulassen (Voreinstellung).
True	Klickgeräusch zulassen.

Datentyp Integer (Boolean)

Cols-Eigenschaft

Beschreibung

Liefert die Anzahl der Button-Spalten, über die das Control verfügt (nicht gültig für Schieberegler).

Syntax

Visual Basic

[form].MyCtrl.**Cols**

Visual C++

pMyCtrl->**GetNumProperty(Cols)**

Anmerkungen

Dies ist eine Nur-Lesen-Eigenschaft.

Datentyp	Integer (enumeriert)

DragIcon-Eigenschaft

Beschreibung
Dient zur Einstellung des Symbols, das als Zeiger in einem Ziehen-und-Ablegen-Vorgang (Drag&Drop) angezeigt wird.

Syntax
Visual Basic
[form].MyCtrl.**DragIcon**[= Wert%]

Anmerkungen
Für Wert% sind folgende Einstellungen möglich:

Einstellung	Beschreibung
(Keine)	Es wird kein Symbol angezeigt (Voreinstellung).
(Symbol)	Anzuzeigendes Symbol.
Datentyp	Integer

Das Zuweisen eines DragIcon-Symbols kann auch programmgesteuert zur Laufzeit erfolgen (Beispiel):

```
Source.DragIcon = LoadPicture("C:\SYMBOLE\FRGZEICH.ICO")   ' Bild laden
Source.DragIcon = LoadPicture()       'Bild "entladen"
```

DragMode-Eigenschaft

Beschreibung

Setzt den manuellen oder automatischen Modus für einen Ziehen-und-Ablegen-Vorgang (Drag&Drop).

Syntax

Visual Basic

[form].MyCtrl.**DragMode**[= Wert%]

Anmerkungen

Für Wert% sind folgende Einstellungen möglich:

Einstellung	Beschreibung
0	Manuell
1	Automatisch

Datentyp Integer (enumeriert)

Enabled-Eigenschaft

Beschreibung

Setzt oder liefert den Freigabe-Status des Controls.

Syntax

Visual Basic

[form].MyCtrl.**Enabled**[= {**True | False**}]

Visual C++

pMyCtrl->**GetNumProperty(Enabled)**
pMyCtrl->**SetNumProperty(Enabled,** {**True | False**}**)**

Anmerkungen

Die Enabled-Eigenschaft bestimmt, ob das Steuerelement auf vom Benutzer erzeugte Ereignisse reagieren kann.

Einstellung	Beschreibung
False	Control ist freigegeben (Voreinstellung).
True	Control ist gesperrt.

Datentyp Integer (Boolean)

Die Einstellung beszieht sich auf das komplette Control. Um den Freigabe-Status einzelner Buttons eines Control zu setzen oder zu ermitteln, verwenden sie die BtnState-Eigenschaft.

Float-Eigenschaft

Beschreibung

Setzt oder liefert die Information, ob das Control verschiebbar ("floating") ist.

Syntax

Visual Basic

[form].MyCtrl.**Float**[= {**True | False**}]

Visual C++

pMyCtrl->**GetNumProperty(Float)**
pMyCtrl->**SetNumProperty(Float,** {**True | False**}**)**

Anmerkungen

Einstellung	Beschreibung
False	Control ist verschiebbar (Voreinstellung).
True	Control ist verschiebbar.

Verschiebbare Controls lassen sich vom Programmbenutzer beliebig im Fenster bewegen; sie verfügen über eine Titelleiste, deren Beschriftung mit der Caption-Eigenschaft eingestellt wird. Die Verschiebbarkeit läßt sich auch während der Programmausführung jederzeit ein- oder ausschalten.

Datentyp Integer (Boolean)

hBMP-Eigenschaft

Beschreibung

Liefert eine Handle (Zugriffsnummer), die von der Windows-Umgebung für die Bitmap-Abbildung des Controls bereitgestellt wird.

Syntax

Visual Basic

[form].MyCtrl.**hBMP**

Visual C++

pMyCtrl->**GetNumProperty(hBMP)**

Anmerkungen

Einstellung	Beschreibung
hBmp	Handle-Nr. der Bitmap.
Datentyp	Integer

Diese Eigenschaft ist zur Entwurfszeit nicht verfügbar und kann zur Laufzeit nur gelesen werden. Für Schieberegler bezieht sich hBmp auf das Bild des Schiebereglerkörpers.

hDC-Eigenschaft

Beschreibung

Liefert eine Handle (Zugriffsnummer), die von der Windows-Umgebung für den Gerätekontext des Controls bereitgestellt wird.

Syntax

Visual Basic

[form].MyCtrl.**hDC**

Visual C++

pMyCtrl->**GetNumProperty(hDC)**

Anmerkungen

Einstellung	Beschreibung
hDC	Handle-Nr. des Gerätekontextes.
Datentyp	Integer

Diese Eigenschaft ist zur Entwurfszeit nicht verfügbar und kann zur Laufzeit nur gelesen werden.

hWnd-Eigenschaft

Beschreibung

Liefert die Window-Handle des Controls.

Syntax

Visual Basic

[form].MyCtrl.**hWnd**

Visual C++

pMyCtrl->**GetNumProperty(hWnd)**

Anmerkungen

Einstellung	Beschreibung
hWnd	Window-Handle des Controls.
Datentyp	Integer

Diese Eigenschaft ist zur Entwurfszeit nicht verfügbar und kann zur Laufzeit nur gelesen werden.

Windows identifiziert jede Form und jedes Steuerelement einer Anwendung anhand einer Zugriffsnummer, die als hWnd bezeichnet wird . hWnd wird im Zusammenhang mit Windows API-Aufrufen verwendet (diverse Funktionen erfordern die hWnd-Zugriffsnummer des aktuellen Objekts als Aufruf-Argument).

Achtung: Der Wert des hWnd-Eigenschaft kann sich während des Programmablaufs ändern. Sie sollten den Wert daher nicht in einer Variablen speichern.

Index-Eigenschaft

Beschreibung

Legt die Nummer fest, die ein Steuerelement eindeutig in einem Steuerelementefeld (Control Array) identifiziert.

Syntax

Visual Basic

[form].MyCtrl.**Index**[= Wert%]

Anmerkungen

Nur dann verfügbar, wenn das Steuerelement Teil eines Steuerelementefeldes ist. Zur Laufzeit schreibgeschützt.

Datentyp	Integer

Left -Eigenschaft

Beschreibung

Setzt oder liefert den Abstand zwischen dem inneren linken Rand des Controls und dem linken Rand des Fensters, zu dem es gehört.

Syntax

Visual Basic

[form].MyCtrl.**Left**[= Wert%]

Visual C++

pMyCtrl->**GetNumProperty(Left)**

pMyCtrl->**SetNumProperty(Left,** Wert**)**

Anmerkungen

Beachten Sie, daß dieser Wert bei Visual Basic von der ScaleMode-Eigenschaft des Eltern-Forms abhängt (Twips, Pixel usw.).

Datentyp	Integer

MaxVal-Eigenschaft

Beschreibung

Setzt oder liefert den Maximalwert für ein Schieberegler-Control.

Syntax

Visual Basic

[form].MyCtrl.**MaxVal**[= Wert&]

Visual C++

pMyCtrl->**GetNumProperty(MaxVal)**

pMyCtrl->**SetNumProperty(MaxVal,** Wert**)**

Anmerkungen

Wert& muß zwischen -2,147,483,648 und +2,147,483,647 liegen.

Datentyp	Long

MinVal-Eigenschaft

Beschreibung

Setzt oder liefert den Minimalwert für ein Schieberegler-Control.

Syntax

Visual Basic

[form].MyCtrl.**MinVal** [= Wert&]

Visual C++

pMyCtrl->**GetNumProperty(MinVal)**

pMyCtrl->**SetNumProperty(MinVal,** Wert**)**

Anmerkungen

Wert& muß zwischen -2,147,483,648 und +2,147,483,647 liegen. Die Grundeinstellung ist null; negative Werte lassen sich jedoch in der Entwicklungsumgebung oder per Programmcode einstellen.

Datentyp	Long

MousePointer-Eigenschaft

Beschreibung

Setzt oder liefert den Typ des Mauszeiges, wenn sich die Maus zur Laufzeit über dem Control befindet.

Syntax

Visual Basic

[form].MyCtrl.**MousePointer**[= Wert%]

Visual C++

pMyCtrl->**GetNumProperty(MousePointer)**

pMyCtrl->**SetNumProperty(MousePointer,** Wert**)**

Anmerkungen
Die Einstellungen für die MousePointer-Eigenschaft sind:

Einstellung	Beschreibung
0	Standard-Pfeilsymbol (Voreinstellung)
1	Pfeil
2	Kreuz
3	Einfügemarke
4	Symbol (kleines Rechteck innerhalb eines anderen)
5	Pfeil N S W O (Vierfachpfeil in alle vier Richtungen)
6	Pfeil NO SW (Doppelpfeil, der nach Nordosten und Südwesten zeigt)
7	Pfeil N S (Doppelpfeil, der nach Norden und Süden zeigt)
8	Pfeil NW SO (Doppelpfeil, der nach Nordwesten und Südosten zeigt)
9	Pfeil W O (Doppelpfeil, der nach Westen und Osten zeigt)
10	Aufwärtspfeil
11	Sanduhr (Warten)
12	Kein Ablegen (für Ziehen und Ablegen)

Datentyp Integer (enumeriert)

Name-Eigenschaft

Beschreibung

Liefert den Namen, der zur Identifikation des Controls im Programmcode verwendet wird.

Syntax

Visual Basic

[form].MyCtrl.**Name**

Visual C++

pMyCtrl->**GetStrProperty(Name)**

Anmerkungen

Diese Eigenschaft steht zur Laufzeit nicht zur Verfügung.

Datentyp	String

Parent-Eigenschaft

Beschreibung

Liefert die Form, die das Control enthält.

Syntax

Visual Basic

[form].MyCtrl.**Parent**

Anmerkungen

Diese Eigenschaft ist zur Entwicklungszeit nicht verfügbar und zur Laufzeit schreibgeschützt.

Verwenden Sie die Parent-Eigenschaft , um auf Eigenschaften, Methoden oder Steuerelemente einer einem Steuerelement übergeordneten Form zuzugreifen. Hier zwei VB-Beispiele:

Slider1.Parent.MousePointer = 4.

ToolBar1.Parent.Print "Hallo"

Datentyp	Integer

Rows-Eigenschaft

Beschreibung

Liefert die Anzahl der Button-Zeilen, über die das Control verfügt (nicht gültig für Schieberegler).

Syntax

Visual Basic

[form].MyCtrl.**Rows**

Visual C++

pMyCtrl->**GetNumProperty(Rows)**

Anmerkungen

Dies ist eine Nur-Lesen-Eigenschaft.

Datentyp	Integer (enumeriert)

Style-Eigenschaft

Beschreibung

Setzt oder liefert die Aktionsart des Controls, die dessen Verhaltensweise bestimmt.

Syntax

Visual Basic

[form].MyCtrl.**Style**[= Wert%]

Visual C++

pMyCtrl->**GetNumProperty(Style)**

pMyCtrl->**SetNumProperty(Style,** Wert**)**

Anmerkungen

Die Einstellungen für die Style-Eigenschaft sind:

Einstellung	Beschreibung
0	COMMANDBUTTON
1	COMMANDBUTTON+NOWAIT
2	TOGGLEBUTTON
3	TOGGLEBUTTON+NOWAIT
4	OPTIONBUTTON
5	TOGGLEBUTTON+NOWAIT
6	DELAYBUTTON
7	SPINBUTTON

Datentyp Integer

Button-Controls können alle acht Einstellungen haben, Schieberegler nur 0 oder 1. Bei 1 geben Schieberegler bei Bewegung des Schiebers sofort Rückmeldung, bei 0 erst bei Absetzen des Schiebers.

TabIndex-Eigenschaft

Beschreibung

Bestimmt die Position eines Controls in der Tab-Reihenfolge seiner übergeordneten Form.

Syntax

Visual Basic

[form].MyCtrl.**TabIndex**[= Wert%]

Visual C++

pMyCtrl->**GetNumProperty(TabIndex)**

pMyCtrl->**SetNumProperty(TabIndex,** Wert**)**

Anmerkungen

Wert% liegt im Bereich von 0 bis (n - 1), wobei n die Anzahl der Controls auf dem Form ist.

Datentyp Integer (enumeriert)

Hinweis

VBX-Controls des vorliegenden Typs können aus einer Vielzahl von Buttons bestehen; da stets das gesamte Control den Tabulator bekommt, erscheint kein Tab-Rechteck. Dieses Verhalten entspricht z. B. den Buttons der VB- oder Winword-Werkzeugleisten.

TabStop-Eigenschaft

Beschreibung

Legt fest, ob bei Verwendung der Tab-Taste auf dem Control angehalten oder ob es übergangen wird.

Syntax

Visual Basic

[form].MyCtrl.**TabStop**[= {**True | False**}]

Visual C++

pMyCtrl->**GetNumProperty(TabStop)**

pMyCtrl->**SetNumProperty(TabStop,** {**True | False**}**)**

Anmerkungen

Einstellung	Beschreibung
False	Tab-Stop ist freigegeben (Voreinstellung).
True	Tab-Stop ist abgeschaltet.
Datentyp	Integer (Boolean)

Tag-Eigenschaft

Beschreibung

Stellt für das Control Speicherplatz zum Ablegen beliebiger String-Informationen zur Verfügung.

Syntax

Visual Basic

[form].MyCtrl.**Tag**[= Wert$]

Visual C++

pMyCtrl->**GetNumProperty(Tag)**

pMyCtrl->**SetNumProperty(Tag,** Wert**)**

Anmerkungen

Datentyp String

Top-Eigenschaft

Beschreibung

Setzt oder liefert den Abstand zwischen dem inneren oberen Rand des Controls und dem linken Rand des Fensters, zu dem es gehört.

Syntax

Visual Basic

[form].MyCtrl.**Top**[= Wert%]

Visual C++

pMyCtrl->**GetNumProperty(Top)**

pMyCtrl->**SetNumProperty(Top,** Wert**)**

Anmerkungen

Beachten Sie, daß dieser Wert bei Visual Basic von der ScaleMode-Eigenschaft des Eltern-Forms abhängt (Twips, Pixel usw.).

Datentyp Integer

Value-Eigenschaft

Beschreibung

Setzt oder liefert den aktuellen Wert eines Schieberegler-Controls. Für Button-Controls ist diese Eigenschaft nicht verfügbar.

Syntax

Visual Basic

[form].MyCtrl.**Value**[= Wert&]

Visual C++

pMyCtrl->**GetNumProperty(Value)**

pMyCtrl->**SetNumProperty(MinVal,** Wert**)**

Anmerkungen

Das Setzen dieser Eigenschaft bewirkt, daß der Schieber auf dem Schiebereglerkörper entsprechend positioniert wird.

Wert& muß zwischen -2,147,483,648 und +2,147,483,647 liegen.

Datentyp Long

Visible-Eigenschaft

Beschreibung

Bestimmt, ob das Control sichtbar oder ausgeblendet ist.

Syntax

Visual Basic

[form].MyCtrl.**Visible**[= {**True | False**}]

Visual C++

pMyCtrl->**GetNumProperty(Visible)**

pMyCtrl->**SetNumProperty(Visible,** {**True | False**}**)**

Anmerkungen

Einstellung	Beschreibung
True	Das Control wird angezeigt (Voreinstellung).
False	Das Control ist ausgeblendet.

Datentyp	Integer (Boolean)

Methoden

Für die mit VBXpress erstellten Controls stehen folgende Methoden zur Verfügung:

Drag
Move
Refresh
SetFocus

Weitere Informationen über die Verwendung von Methoden finden Sie gegebenenfalls in den Handbüchern des von Ihnen verwendeten Windows-Entwicklungssystems.

Drag-Methode

Beschreibung
Diese Methode beginnt oder beendet das Ziehen von Steuerelementen oder bricht es ab.

Syntax
Visual Basic
[form].MyCtrl.**Drag** [Funktion%]

Anmerkungen
MyCtrl ist der Name des Controls.
Funktion ist ein Integerwert (0, 1 oder 2), der die auszuführende Aktion bestimmt. Wenn Sie diesen Parameter weglassen, wird mit dem Ziehen des Steuerelementes begonnen.

Funktion	**Beschreibung**
0	Zieh-Operation abbrechen.
1	Mit dem Ziehen von MyCtrl beginnen.
2	Ziehen beenden, also MyCtrl ablegen.

Die Verwendung der Drag-Methode zum Ziehen und Ablegen eines Controls ist nur notwendig, wenn die DragMode-Eigenschaft des entsprechenden Steuerelementes auf 0 (=Manuell) gesetzt ist. Sie können Drag jedoch auch bei Controls verwenden, deren DragMode-Eigenschaft auf 1 (=Automatisch) gesetzt ist. Soll sich der Mauszeiger während des Ziehvorgangs ändern, verwenden Sie die DragIcon- oder MousePointer-Eigenschaft zur Definition Ihres Mauszeigers. Beachten Sie, daß die MousePointer-Eigenschaft nur genutzt wird, wenn DragIcon nicht festgelegt wurde.

Move-Methode

Beschreibung
Diese Methode verschiebt das im Aufruf angegebene Control.

Syntax
Visual Basic
[form].MyCtrl.**Move** links[, oben[, Breite[, Höhe]]]

Anmerkungen
MyCtrl ist der Name des zu verschiebenden Controls.

Außerdem werden folgende Parameter angegeben:

Parameter	**Beschreibung**
links	Einfach genauer Wert, der die horizontale Koordinate für den linken Rand von MyCtrl angibt.
oben	Einfach genauer Wert, der die vertikale Koordinate für den oberen Rand von MyCtrl angibt.
Breite	Einfach genauer Wert, der die neue Breite von MyCtrl angibt.
Höhe	Einfach genauer Wert, der die neue Höhe von MyCtrl angibt.

Refresh-Methode

Beschreibung
Erzwingt das sofortige Neuzeichnen des angegebenen Controls.

Syntax
Visual Basic
[form].MyCtrl.**Refresh**

SetFocus-Methode

Beschreibung
Diese Methode setzt den Eingabe-Fokus auf das angegebene Control.

Syntax
Visual Basic
[form].MyCtrl.**SetFocus**

Anmerkungen
VBX-Controls des vorliegenden Typs können aus einer Vielzahl von Buttons bestehen; da stets das gesamte Control den Tabulator bekommt, erscheint kein Fokus-Rechteck. Dieses Verhalten entspricht z. B. den Buttons der VB- oder Winword-Werkzeugleisten.

Was ist VBXpress?

Eines der wichtigsten Bestandteile eines Programms ist die Verbindung zum Anwender, die sogenannte Benutzerschnittstelle. Je komfortabler und intuitiver sie ist, um so einfacher läßt sich ein Programm bedienen - ein wichtiger Wettbewerbsvorteil im heiß umkämpften Markt der Windows-Anwendungen. Die wichtigsten Eingabe-Elemente wie Buttons, Menüs und Textfelder stellt Visual Basic Ihnen in seiner Toolbox bereit; auch andere Windows-Entwicklungssysteme verfügen über ein Grundarsenal an solchen Steuerelementen. Jedoch wurde das Graphical User Interface (GUI) gerade in jüngster Zeit immer weiter verfeinert und um Raffinessen erweitert; diverse Bedienungselemente, die man von kommerziellen Windows-Anwendungen kennt und schätzt, stehen in Visual Basic und anderen Windows-Entwicklungssystemen standardmäßig nicht zur Verfügung.

Und genau hier rüsten Sie Ihre Programmierumgebung mit VBXpress auf - vorausgesetzt, sie erlaubt den Einsatz von Steuerelementen in form von VBX-Controls, denn diese lassen sich mit dem VBXpress-Control-Designer extrem einfach erstellen.

Auch wenn die nächste Windows-Version zusätzlich zu VBX den OLE-Custom-Control-Typ ".OCX" unterstützt - der Fachpresse ist bereits eindeutig zu entnehmen, daß "Chicago" weiterhin auch den VBX-Standard voll unterstützen wird (bedenkt man, daß allein der Visual Basic Produktkatalog Sommer 1994 rund 70 Toolboxen mit zusammen Hunderten VBX-Controls enthält, ist dies auch nicht weiter verwunderlich).

Control-Typen

VBXpress ermöglicht Ihnen die blitzschnelle und problemlose Erstellung folgender VBX-Control-Typen:

- **Werkzeug-Paletten (Floating Tool Palettes).**
- **Horizontale und vertikale Button-Werkzeugleisten (Toolbars).**
- **Individuelle Befehlsschaltflächen (Command Buttons).**
- **Spin-Buttons.**
- **Wippschalter.**
- **Realistisch aussehende Schieberegler, Mischpult-Bedienfelder und Meßinstrumente.**

Zu allen genannten Typen finden Sie anbei Beispiel-Controls!

Das Verhalten dieser Steuerelemente kann flexibel angepaßt werden - beispielsweise kann durch Ändern einer einzigen Eigenschaft bestimmt werden, ob Buttons einer Werkzeugleiste sich gegenseitig auslösen sollen oder mehrere gedrückt werden können.

Alle Controls können auch verschiebbar ("floating") genutzt werden - sie erhalten dann eine Teitelleiste und sind im Fenster frei verschiebbar. Mit Hilfe der Float-Eigenschaft kann diese Funktionalität jederzeit ein- oder augeschaltet werden.

VBXpress bietet zudem den Vorteil, daß sich die VBX-Dateien auf die Benutzung in der VB-Entwicklungsumgebung beschränken lassen, oder umgekehrt auf den Einsatz mit kompilierten EXE-Programmen. Ersteres ermöglicht die problemlose Weitergabe von Controls "zum Ausprobieren", letzteres verhindert die ungewollte Verwendung der Controls für Fremdanwendungen.

Der VBXpress-Control-Designer

Die Gestaltung der GUI-Steuerelemente (Controls) geschieht interaktiv mit dem komfortablen **VBXpress-Control-Designer**, für den **Windows-Online-Hilfe** sowie eine **umfangreiche Grafik-Bibliothek** zur Verfügung steht. Sie benötigen **weder C, CDK noch intime Windows-Kenntnisse**! Als fertiges Produkt erhalten Sie auf Knopfdruck Windows-konforme VBX-Dateien, die von bedeutenden Windows-Entwicklungssystemen unterstützt werden, zum Beispiel:

- Microsoft Visual Basic (alle Versionen)
- Microsoft Visual C++
- Borland dBASE for Windows
- Borland C++ 4.0
- und ... und ...

Für jeden Schritt der VBX-Erstellung bietet VBXpress Hilfestellung an. Mit dem VBXpress-Control-Designer erstellen Sie mit Bildern versehene Command-Buttons sowie maßgeschneiderte horizontale oder vertikale Toolbars und Schieberegler in wenigen Minuten. Die Controls "leben" bereits in der VBXpress-Entwicklungsumgebung. Ein einziger Menübefehl genügt, um daraus eine **vollwertige VBX-Datei** zu erstellen!

Da Visual Basic VBX-Controls bereits seit 1991 verwendet (für diese Sprache wurde der VBX-Standard ursprünglich entwickelt), stellen VB-Entwickler zugleich auch die Mehrheit der VBX-Anwender dar. Aus diesem Grund wurden in VBXpress speziell für VB-Programmierer zahlreiche Beispiele vorgesehen.

Egal, ob Sie genau jene Controls erstellen wollen, die Ihre Anwendung erfordert, oder Control-Sammlungen für andere Entwickler erstellen wollen: **Mit VBXpress produzieren Sie VBX-Dateien im Handumdrehn!**

VBX-Controls

VBX-Controls sind spezielle DLLs, die dem Windows-Entwickler viel Komfort bieten: Einmal in die Werkzeugsammlung von Visual Basic oder in die Tool-Palette des VC++ App-Studio geladen, kann er sie beliebig auf seine Formulare ziehen und mit wenig Programmieraufwand benutzen. Wie andere DLLs auch wird das VBX-Control nicht Bestandteil der kompilierten EXE-Datei, sondern bleibt eine separate Datei.

Zoschke Data GmbH
Bahnhofstraße 3
D-24217 Schönberg/Holstein

Tel. 04344/6166 Fax: 04344/6162

CompuServe E-Mail: 71340,2051

Weitere Tools bei Zoschke Data

Bei Zoschke Data bekommen Sie jede Menge nützlicher Tools rund um Visual Basic for Windows und zur Windows-Hilfe-Entwicklung:

Toolboxen rund um Visual Basic for Windows

Toolboxen zur Windows-Hilfe-Entwicklung

Für folgende Tool-Hersteller ist Zoschke Data Exklusiv-Distributor für Deutschland, Österreich und die Schweiz:

- Apex Software
- Crescent Software
- Desaware
- Marquis Computing
- Sheridan Software Systems
- Software Interphase

Toolboxen rund um Visual Basic for Windows

CCF-Cursors 2.0 von Desaware
Custom Control Factory 2.0 von Desaware
Data Widgets von Sheridan Software Systems
db/VB (=Visual/db) von Marquis Computing
Designer Widgets von Sheridan Software Systems
NetPak Professional von Crescent Software
PDQComm for Windows von Crescent Software
QuickPak Professional for Windows von Crescent Software
QuickPak Scientific for Windows von Crescent Software
SpyWorks-VB von Desaware
SpyNotes #2 von Desaware
TrueGrid Pro von Apex Software
VBAssist von Sheridan Software Systems
VB/magic Controls von Marquis Computing
VersionStamper-VB von Desaware
3-D Widgets von Sheridan Software Systems
VBDD (Data Dictionary) von Crescent Software
VBXpress von Marquis Computing
XREF for Visual Basic von Crescent Software

Toolboxen zur Windows-Hilfe-Entwicklung

Windows Help Magician von Software Interphase

Windows Bitmap Magician von Software Interphase

Winhelp-Makros von Zoschke Data

CCF-Cursors 2.0 von Desaware

CCF-Cursors erlaubt die Erstellung und Verwendung beliebiger Cursor anhand von Symbolen, wie man sie z. B. mit dem im VB enthaltenen Icon-Editor erstellen kann. Zur Umwandlung ins Cursor-Format wird eine komfortable Utility mitgeliefert. Hinzu kommen Erweiterungen der Maus- und Cursor-Steuerung, automatische Cursor-Animation, Überwachung der Menüauswahl-Ereignisse und nützliche Anwendungsbeispiele. Deutsches Handbuch. Keine Lizenzgebühren.

Custom Control Factory 2.0 von Desaware

Die Custom Control Factory ist die erweiterte Version des "Animated Button" (VB2.0/3.0 Professional) und dient zur Erstellung nahezu beliebiger Buttons allein durch die Angabe von Eigenschaften eines einzigen VBX. Highlights: Beliebige Bitmaps (auch 256 Farben) als Button-Bilder, mehrzeilige Beschriftungen mit wählbarer Ausrichtung und Position, automatischer 3D-Effekt, Bilddaten-Kompression, anwenderdefinierbare Ablaufzyklen und DDE-Unterstützung. Deutsches Handbuch. Keine Lizenzgebühren.

Data Widgets von Sheridan Software Systems

Data Widgets ist eine Sammlung gebundener Controls, die das Erstellen von Datenbank-Anwendungen mit VB 3.0 nahezu ohne zusätzlichen Programmcode ermöglicht. DataGrid ist ein Gitternetz mit dem "Look and Feel" von Microsoft Access, das direktes Editieren in den Zellen, eingebautes Add/Update/Delete, indiviuelle Zellenfarben und Unterstützung für Daten > 64K bietet. Außerdem: DataButton, DataDropDown, DataCombo, DataOption und erweitertes Datenbank-Control mit Lesezeichen und variabler Datensatzanzahl zum Blättern. Keine Lizenzgebühren.

db/VB 3.x (=Visual/db) von Marquis Computing

db/VB ist eine Datenbank-Toolbox für VB2.0/3.0, die das Dateiformat dBase III+/IV nutzt. Die Bibliothek mit Funktionen zum Anlegen, Durchsuchen, Auswerten und Bearbeiten von DBF-, NDX- und DBT-Dateien wurde komplett in VB geschrieben, daher ist keine DLL-Engine erforderlich. Für Indizieren, Packen, Anzeigen, Browse etc. bietet db/VB einsatzbereite Anwendungen. Expression Evaluator mit vielen dBASE- und BASIC-Funktionen. Code Optimizer zur Programmoptimierung. Handbuch und Online-Hilfe deutsch. Keine Lizenzgebühren.

Designer Widgets von Sheridan Software Systems

Designer Widgets bietet Custom Controls, mit denen VB-Programme das "Look and Feel" von Office-Anwendungen erhalten: Die Dockable Toolbar ermöglicht verschiebbare Button-Leisten mit optionaler Ballon-Hilfe. Mit den Index Tabs lassen sich effiziente Tab-Dialoge erstellen; die Tabs können an allen vier Seiten erscheinen. Mit FormFX kann man das Aussehen von Forms verbessern, z. B. 3D-Titelleiste mit Bildern und variabler Höhe und frei wählbarer Schriftart. Keine Lizenzgebühren.

NetPak Professional von Crescent Software

Mit VB und NetPak Professional lassen sich Anwendungen mit Netzwerk-Funktionalität schnell erstellen. Die Toolbox enthält DLLs und VBX-Controls sowohl für Novell Netware als auch Microsoft Windows for Workgroups. Für NetWare-Server und -Workstations bietet NetPak z. B. Druckfunktionen, Bindery-, QMS-, TTS-, Nachrichten- und Verzeichnis-Funktionen sowie NetWare-386-spezifische Funktionen (F2). Für WfWg lassen sich unter anderem die darin enthaltenen Benutzerdialoge auf einfache Weise aktivieren. Sämtliche Quelltexte. Keine Lizenzgebühren.

PDQComm for Windows von Crescent Software

PDQComm for Windows enthält eine erheblich erweiterte Version des ebenfalls von Crescent entwickelten MSComm-Controls (VB2.0/3.0 Professional). Highlights: Terminal-Emulation (ANSI, TTY, VT52 und VT100) mit automatischem Terminal-Fenster mit Scrollback-Puffer, Hintergrund-Dateiübertragung (ASCII, CompuServe B+, Kermit, XModem-Checksum, XModem-CRC, XModem-1K, YModem-Batch, YModem-G und ZMdem), Scriptsprache mit Recorder sowie Modem-Datenbank mit Daten zu mehr als 440 Modems. Sämtliche Quelltexte. Keine Lizenzgebühren.

QuickPak Professional for Windows von Crescent Software

QuickPak Professional 3.x for Windows ist die umfassende Sammlung von Profi-Tools für Visual Basic. In einem Paket "aus einem Guß" finden Sie hier mehr als 30 Custom Controls (z.B. ein mächtiges Hypertext-Control und ein gebundenes TrueColor-Bildfeld für PCX/BMP/ GIF/TGA mit Zoom), über 400 DLL-Routinen, mehr als 2,7 Mbyte Programmbeispiel-Code, über 1000 Seiten Dokumentation, mächtige Programmier-Utilities (z. B. FormWizard) und sämtliche Quelltexte (auch DLL und VBX). TrueGrid 2.0 (Standard) von Apex als Bonus. Keine Lizenzgebühren.

QuickPak Scientific for Windows von Crescent Software

QuickPak Scientific for Windows enthält mathematische Routinen für (fast) jeden Zweck, z. B. Differential-Gleichungssysteme, Funktions-Minima/-Maxima, komplexe Zahlen, Kurvenglättung, Vektoren, Matrizenrechnung, Lineare Algebra, numerische Differentiation und Integration, Trigonometrie, Statistik, Fourier-Transformation, Wurzelberechnung, Interpolation, Optimierung, Gamma- und Besselfunktion, Gauß-Fehlerfunktion u.v.a.m. Komplett in VB geschrieben. Deutsches Handbuch. Keine Lizenzgebühren.

SpyWorks-VB von Desaware

SpyWorks-VB ermöglicht erfahrenen Visual-Basic-Programmierern mit Windows-Kenntnissen mächtige Windows-Techniken, die bisher C/SDK-Programmierern vorbehalten waren. Die Toolbox enthält Custom Controls für Subclassing, Windows-Hooks und Callbacks. Hinzu kommen mächtige Debugging-Tools und Spy-Utilities. SpyNotes #1 mit interessanten Anwendungsbeispielen als Bonus enthalten. Keine Lizenzgebühren.

SpyNotes #2 von Desaware

SpyNotes #2 - The Common Dialog Toolkit ist eine Ergänzung zu SpyWorks-VB und ermöglicht VB-Programmierern, auf die Windows-Standarddialoge zuzugreifen und sie den Erfordernissen entsprechend maßzuschneidern (z. Erweiterung mit VB-Controls, freie Wahl der Position und Abmessungen einer Standarddialogbox, Registrierung und Subclassing anwenderdefinierter und registrierter Nachrichten, nichtmodale Verwendung von Standarddialogen). Viele Beispiele. Keine Lizenzgebühren.

TrueGrid Pro von Apex Software

TrueGrid Pro ist ein vielseitiges gebunden und ungebunden einsetzbares Gitternetz-Steuerelement für VB3-Datenbank-Anwendungen nahezu ohne zusätzlichen Programmcode. Highlights: Für Datenbanken beliebiger Größe und alle Formate, die auch VB kennt. WYSIWYG-Layout Editor. Zur Laufzeit änderbare Spaltenbreite, -Reihenfolge und Splits. Volle Format$-Unterstützung. Auch Bilder in Grid-Zellen. Drag & Drop von Zellen. Spaltensummen und berechnete Spalten. Comboboxen und Optionsfelder in Zellen. Value Lists. Keine Lizenzgebühren.

VBAssist 3.0 von Sheridan Software Systems

VBAssist 3.0 erweitert die VB-Entwicklungsumgebung um viele nützliche Hilfsmittel zur Produktivitätssteigerung, z. B. Control-Ausrichtung, Control-Array-Manipulation, Änderung der Tab-Folge, Code-Verwaltung, Eigenschaften-Schablonen, themenorientierte Eigenschaften-Listen, Datenbank-Assistent zur Control-Anbindung per Drag & Drop, FormWizard zur automatischen Erzeugung kompletter Datenbank-Forms, Screen-Capture, scrollfähige Forms und Bildfelder und vieles mehr. Macht aus VB ein "Super-VB".

VB/magic Controls von Marquis Computing

Mit den VB/magic Controls stehen Ihnen attraktive Eingabe-Elemente zur Verfügung, z. B. verschiebbare Werkzeug-Paletten und Toolbars, bebilderte Befehlsschaltflächen (sogar runde), realistische, benutzerdefinierte Dialogboxen, Wippschalter sowie Popup-, Fly-Out- und Rollo-Menüs. VB/magic-Controls sind pures VB, kommen daher ohne zusätzliche VBX- oder DLL-Dateien aus. Der Entwurf der Controls geschieht geschieht per Control-Designer mit deutscher Benutzerführung. Deutsches Handbuch. Keine Lizenzgebühren.

VersionStamper-VB von Desaware

VersionStamper-VB ermöglicht es endlich, in ausführbare VB3.0-Programme Versionsinformationen für die EXE-Datei und alle von der Anwendung benutzten DLL- und VBX-Dateien einzubetten. Damit lassen sich unkompatible VBX- oder DLL-Versionen auf dem Zielsystem umgehend erkennen und Versionskonflikte vermeiden. Die gelieferten Versionsinformationen sind zu allen Windows-basierten Installations-Programmen kompatibel. Außerdem wird gezeigt, wie sich der mit Visual Basic gelieferte Setup Toolkit diesbezüglich verbessern läßt. Keine Lizenzgebühren.

3-D Widgets von Sheridan Software Systems

3-D Widgets ist eine Erweiterung des ebenfalls von Sheridan entwickelten THREED.VBX (VB2.0/3.0 Professional). Im 3D-Look und mit erweiterter Funktionalität unter anderem enthalten: Auswahlkästchen, Befehlsschaltfläche, Rahmen, Panel, Ribbon sowie Listenfelder (mit Mehrfachspalten, Mehrfachauswahl, Bildern und horizontalem Rollen). Auch Kombi-, Laufwerk-, Verzeichnis- und Datei-Listenfelder wurden funktionell und optisch erweitert. Ein weiteres Control erlaubt die Umsetzung vorhandener Menüleisten in 3D-Look, sogar mit Bildern.

VBDD von Crescent Software

VBDD - The Data Dictionary for Visual Basic - macht die VB-Datenbank-Programmierung einfacher, schneller und effektiver. VBDD öffnet die Access-Engine und macht VB mächtige Funktionen wie QueryDefs mit Parametern zugänglich. Außerdem ergänzt VBDD wichtige Funktionen wie volle referentielle Integrität, Pseudo-Datenfelder, Bereichs- und Wertüberprüfung und umfangreiche Tabellenwartung. QueryDef Builder für ausgeklügelte Abfragen. Zum Einbau ins Programm werden maßgeschneiderte einsatzbereite Code-Clips generiert. Keine Lizenzgebühren.

VBXpress von Marquis Computing

Mit dem VBXpress-Control-Designer entwerfen Sie GUI-Controls wie Toolbars, verschiebbare Tool-Paletten, Grafik-Buttons, Spinbuttons und realistische Schieberegler. Dann wählen Sie "VBX erstellen..." und erstellen daraus echte VBX-Dateien, die Sie lizenzkostenfrei verwenden und verbreiten können - ohne C und CDK! Für Visual Basic, VC++, Borland C++ 4.0 und andere Sprachen, die VBX unterstützen. Mehr als 15 Beispiel-Controls sind bereits enthalten. Deutsche Benutzerführung, Online-Hilfe und Handbuch.

XREF for Visual Basic von Crescent Software

XREF for Visual Basic ist ein leistungsfähiges Tool zur Dokumentation, Programm-Optimierung und Fehlersuche. Aussagefähige Reports informieren über Forms, Controls, Eigenschaften, Ereignisse, Variablen, Konstanten, Prozeduren, wer was aufruft usw. Auch eine Aufstellung unbenutzter Routinen und Symbole sowie Aufruf-Baumdiagramme und formatierte Programm-Listings mit Seitentiteln, Datum und Uhrzeit lassen sich erstellen. Alle REMs und/oder Stringtexte lassen sich zum Übersetzen bzw. Prüfen extrahieren und automatisch wieder einfügen. Deutsches Handbuch.

Windows Help Magician von Software Interphase

Windows Help Magician ist ein komfortables Entwicklungssystem zum Erstellen von Windows-Hilfedateien. Eigener Texteditor mit Funktionen zum Einrichten von Sprüngen und Popups, Einfügen von Bitmaps, Sounds, Videos und Winhelp-Makros, Auswählen von Browse-Folgen, Im-/Exportieren von RTF- und Textdateien (z. B. Handbücher) usw. Einfache menügesteuerte Bedienung; selbst der Help-Compiler wird automatisch aufgerufen. Der Testmodus erlaubt komplettes Testen ohne Kompilierung. Textverarbeitungs-unabhängig. Deutsches Handbuch.

Windows Bitmap Magician von Software Interphase

Der Windows Bitmap Magician automatisiert den Prozeß, Zeichen beliebiger Fonts in Help-Dateien zu verwenden. Hierzu werden die Zeichen in Bitmaps umgewandelt. Jetzt können Sie in Ihrer Online-Hilfe z.B. Symbole aus Wingdings, Dingbats etc. als Blickpunkte verwenden. Weiterer Vorteil: Die Benutzer Ihrer Hilfe brauchen die Spezial-Fonts nicht installiert zu haben; Ihre Help-Dateien bleiben somit voll kompatibel zum Rest der (Windows-)Welt.

Winhelp-Makros von Zoschke Data

Mit der in Winhelp integrierten Befehlssprache läßt sich Windows-Hilfe individuell erweitern und maßschneidern, z. B. um eigene Buttons, Menüs und sogar Sounds und Video-Clips. Die deutsche Buch-Toolbox Winhelp-Makros - Befehlsreferenz, Tips und Tools macht Hilfe-Autoren mit Syntax und Anwendung von Winhelp-Makrobefehlen vertraut und geht dabei sowohl auf die Hilfe-Entwicklung mit Winword als auch mit dem Windows Help Magician ein. Die Begleitdiskette enthält Hilfe-Beispiele und nützliche Winhelp-Utilities.

VBXpress-Lieferumfang

- **VBXpress-Control-Designer mit deutscher Benutzeroberfläche** - Komplettes System zum Entwurf und zur Erstellung von VBX-Controls.
- **Control-Beispiele** - Für alle von VBXpress unterstützten Control-Typen wurden Beispiele vorgesehen, und zwar sowohl als editierbare VBXpress-Ressourcen-Dateien als auch als einsatzbereite VBX-Dateien.
- **Programmbeispiele** - Für Visual Basic wurden Programmbeispiele vorgesehen, die den Einsatz mit VBXpress erstellter Controls demonstrieren.
- **Deutsche Online-Hilfe** - Die Toolbox enthält eine kontextbezogene Online-Hilfe für den Controls-Designer und die damit erstellbaren VBX-Controls.
- **VBX-Online-Hilfe zur Weitergabe** - Eine neutrale VBX-Hilfedatei, die registrierte VBXpress-Anwender zusammen mit den von ihnen erstellten VBX-Controls lizenzkostenfrei verbreiten dürfen.
- **Ausführliches deutsches Handbuch** (Ringordner, ca. 140 Seiten mit Stichwortverzeichnis und vielen Abbildungen).
- **Lizenz** - Der VBXpress-Kaufpreis berechtigt den Lizenznehmer, die von ihm mit dem VBXpress-Control-Designer erzeugten VBX-Controls ohne Zahlung von Lizenzgebühren zu verbreiten.

Zoschke Data Bestell-Nr. 850