

# VideoSoft Custom Controls

---

Welcome to the VideoSoft Custom Control, a set of powerful custom controls for [Microsoft Visual Basic](#).

## How to Use



[3D Elastic Control](#)



[IndexTab Control](#)



[AWKSome Control](#)

[Registration](#)

[What do you get?](#)

[Product Support](#)

[Tips and Tricks.](#)



## VideoSoft IndexTab Custom Control

---

The VideoSoft IndexTab control allows you to group controls by subject using a familiar notebook metaphor.

To use the IndexTab, follow these steps:

- 1) Draw the IndexTab.
- 2) Set the caption property to create one tab per control group.
- 3) Draw one container per group inside the IndexTab control.
- 4) Draw the controls inside each container.
- 5) Set other IndexTab properties as desired.

### Properties

[\(About...\)](#)

[AutoScroll](#)

[AutoSwitch](#)

[Caption](#)

[BackColor](#)

[BackTabColor](#)

[ClientHeight](#)

[ClientLeft](#)

[ClientTop](#)

[ClientWidth](#)

[CurrTab](#)

[FirstTab](#)

[ForeColor](#)

[FrontTabColor](#)

[NumTabs](#)

[Position](#)

[Style](#)

### Standard Visual Basic Properties

Align

DragIcon

DragMode

Enabled

FontBold

FontItalic

FontName

FontSize

FontStrikethru

FontUnderline

Height

Index

Left

MousePointer

Name

TabStop

Tag

Top

Visible  
Width

### **Events**

Click  
DblClick  
DragDrop  
DragOver  
GotFocus  
KeyDown  
KeyPress  
KeyUp  
LostFocus  
MouseDown  
MouseMove  
MouseUp  
Scroll

For Standard properties and events check your Visual Basic documentation.

## About ...

---

### Applies to

[3D Elastic](#)

[IndexTab](#)

[Awkesome](#)

### Description

Displays version and copyright information for the control; read-only at design time and not available at run time.

You can click the ellipsis ("...") button to the right of the settings box in the properties window to display the About dialog. The dialog contains the toolbox icon for the control, the version number of the control, and copyright information.

## BackColor, BackTabColor, ForeColor, FrontTabColor Properties

---

### Applies to

IndexTab

### Description

The IndexTab control uses the Microsoft Windows environment RGB scheme of colors. There are four color properties that can be set on an IndexTab control:

BackColor	Determines the background color of the area behind the tabs.
ForeColor	Sets the color of the text.
BackTabColor	Determines the background color of all non-selected the tabs.
FrontTabColor	Determines the color for the currently selected tab and for the background of the client area of the control.

## Caption Property

---

### Applies to

IndexTab

### Description

Determines the number of tabs and the text they contain.

### Usage

#### Visual Basic

```
[form.][control.]Caption = itemstring$
```

### Remarks

To create a new tab place the pipe character ("|") after the text of your previous tab and type the desired caption.

```
IdxTab1.Caption = "1st Quarter|2nd Quarter"
```

To space tabs add the hyphen character ("-") after the pipe and place as many spaces as you wish to space the tabs.

```
IdxTab1.Caption = "1st Quarter|-     |2nd Quarter"
```

## Position Property (IndexTab)

---

### Applies to

IndexTab

### Description

Sets or returns the position of the tabs within the control.

Windows only supports vertical text with some fonts. If you set Position to 2 or 3, make sure you have selected a TrueType font.

### Usage

#### Visual Basic

```
[form.][control.]Position = position%
```

Where position% is:

0	Top
1	Bottom
2	Left
3	Right

## Style Property

---

### Applies to

IndexTab

### Description

Determines the style of the tabs' edges of the IndexTab control.

### Usage

#### Visual Basic

```
[form.][control.]Style = style%
```

Where style% is:

- 0 - Slanted lines;
- 1 - Slanted lines plus 3D effect;
- 2 - Rounded corners.
- 3 - Rounded corners plus 3D effect.

Note: The 3D effect is most visible when Position is set to 1 or 3 (bottom or right).



## NumTabs Property

---

### Applies to

IndexTab

### Description

Returns the number of tabs defined by the Caption property. This property is only available at runtime and is read-only.

Invisible tabs, used to separate regular tabs, are also counted.

This read-only property is set whenever you change the Caption.

## ClientLeft, ClientTop, ClientWidth, ClientHeight Properties

---

### Applies to

IndexTab

### Description

Return the internal dimensions of the IndexTab excluding the rectangle used to show the tabs themselves.

These properties are useful if you want to size frames or picture controls so that they occupy the whole client area of the control.

You only need to use these properties if AutoSwitch is off.

These properties are read-only.

For example:

```
Sub Form_Resize ()  
    ' make sure index tab fills form  
    IdxTab1.Move 0, 0  
    IdxTab1.Width = ScaleWidth  
    IdxTab1.Height = ScaleHeight  
  
    ' make sure picture fills index tab  
    Picture1.Move 0, IdxTab1.ClientTop  
    Picture1.Width = IdxTab1.Width  
    Picture1.Height = IdxTab1.ClientHeight  
End Sub
```

## AutoScroll Property

---

### Applies to

IndexTab

### Description

Determines whether the control should automatically scroll the list of tabs so that the current one is always visible.

You can find out when scrolling happened by checking the Scroll event.

### Usage

```
[form.][control.]AutoScroll = TRUE|FALSE
```

## AutoSwitch Property

---

### Applies to

IdxTab

### Description

Determines whether the control should automatically resize, show, and hide child windows when the current tab changes.

To use AutoSwitch, you need to design your IdxTab so that it has one child control per tab, including comment tabs.

The child controls must be arranged by Z-order. The back child should correspond to the first tab, the second-to-back to the second tab, and so on until the front child, which corresponds to the last tab.

The reverse order was chosen because new children are created in front of old ones, so if you create them in order you'll get it right.

This property only acts in run mode.

### Usage

```
[form.][control.]AutoSwitch = TRUE|FALSE
```

## CurrTab Property

---

### Applies to

IndexTab

### Description

Sets or returns the number of the currently selected tab.

The first tab is numbered zero. Invisible tabs, used to separate regular tabs, are also counted.

Setting this property to a negative value or to a value greater than the number of tabs causes all tabs to be de-selected. No error occurs.

Changes to this property trigger the Click event.

### Usage

```
[form.][control.]CurrTab = setting%
```

# FirstTab Property

---

## Applies to

IndexTab

## Description

Sets or returns the number of the first tab to be displayed. The first tab is numbered zero. Invisible tabs, used to separate regular tabs, are also counted. This property is useful if you want to implement scrolling in addition to the AutoScroll capability.

Changes to this property trigger the Scroll event.

## Usage

```
[form.][control.]FirstTab = setting%
```

## Scroll Event

---

### Applies to

IndexTab

### Description

Occurs whenever the FirstTab changes either through code, mouse, or keyboard action.

### Usage

```
Sub CtlName_Scroll ()
```



## VideoSoft Elastic Custom Control

---

The VideoSoft Elastic control is a versatile smart container. It can save you hundreds of lines of VB code by allowing you to:

- 1) Give regular controls a 3-D look.
- 2) Automatically resize the Elastic to the left, right, top, bottom, or to fill its container, be it a form or another control.
- 3) Automatically resize the controls inside the Elastic, evenly or unevenly, vertically or horizontally.
- 4) Allow the user to resize controls inside the Elastic at run time, using the mouse.
- 5) Create multi-line labels, 3-D gauges, or both at the same time in a single Elastic.

Use nested Elastics to create complex screens that resize automatically when the user changes the size of the form.

NOTE: Check the align property for automatic resizing.

### Properties

[\(About...\)](#)  
[Align](#)  
[AutoSizeChildren](#)  
[BevelInner](#)  
[BevelInnerWidth](#)  
[BevelOuter](#)  
[BevelOuterWidth](#)  
[BorderWidth](#)  
[CaptionPos](#)  
[ChildSpacing](#)  
[FloodColor](#)  
[FloodDirection](#)  
[FloodPercent](#)  
[Splitter](#)

### Standard Visual Basic Properties

BackColor  
Caption  
DragIcon  
DragMode  
Enabled  
FontBold  
FontItalic  
FontName  
FontSize  
FontStrikethru  
FontUnderline  
ForeColor  
Height  
Index  
Left  
MousePointer  
Name  
TabIndex



TabStop  
Tag  
Top  
Visible  
Width

### **Events**

Click  
DblClick  
DragDrop  
DragOver  
GotFocus  
KeyDown  
KeyPress  
KeyUp  
LostFocus  
MouseDown  
MouseMove  
MouseUp  
RealignFrame  
ResizeChildren

For Standard properties and events check your Visual Basic documentation.

## Align Property

---

### Applies to

Elastic

### Description

Determines whether an elastic box can appear in any size anywhere on a form, or whether it appears at the top, bottom, left or right of the form automatically sized to fit the form's width or height.

The Elastic's Align property is similar to the standard Align property, except for two things:

- 1) The Elastic's Align has three additional settings: Left, Right, and Fill Container.
- 2) The Elastic's Align must be triggered from the Form\_Resize subroutine with a statement such as  
VSElastic1.Left = 0

If you have many Elastics in a form, use the top left one in the statement. All others will be resized too.

### Usage

[form.]VSElastic.Align = numericexpresion

### Setting Description:

#### **0        None.**

(Default) Nonesize and location can be set at design time or in code.

#### **1        Top.**

Elastic box is at the top of the form and its width is equal to the form's ScaleWidth property setting.

#### **2        Bottom.**

Elastic box is at the bottom of the form and its width is equal to the form's ScaleWidth property setting.

#### **3        Left.**

Elastic box is at the left of the form and its width is equal to the form's ScaleHeight property setting.

#### **4        Righth.**

Elastic box is at the right of the form and its width is equal to the form's ScaleHeight property setting.

#### **5        Fill container.**

The Elastic box is aligned to the available space in your form. You can set the elastic to always match the size of the entire form.

### Example

VSElastic1.Align = 0      'User defined alignment

VSElastic1.Align = 1      'Aligns to the top

## AutoSizeChildren Property

---

### Applies to

Elastic

### Description

Controls how the Elastic resizes the controls inside it.

This property is useful at design time, to create forms that look professional, and at run time, to make all controls inside the Elastic resize themselves when the form is resized.

### Usage

```
[form.]VSElastic.AutoSizeChildren = numericexpresion
```

### Setting Description:

- 0 - No automatic children sizing
- 1 - Even Horizontally
- 2 - Uneven Horizontally
- 3 - Even Vertically
- 4 - Uneven Vertically

The Even Horizontally option makes all controls in the Elastic have the same width and spreads them evenly across the Elastic.

The Even Vertically option makes all controls in the Elastic have the same height and spreads them evenly down the Elastic.

The Uneven Horizontally option stretches only the top control inside the Elastic (highest Z-Order) and spreads all child controls evenly across the Elastic.

The Uneven Vertically option stretches only the top control inside the Elastic (highest Z-Order) and spreads all child controls evenly across the Elastic.

The maximum number of children an Elastic can resize is 30. If you need more, use nested Elastics.

### Example

```
VSElastic1.AutoSizeChildren = 0    'User defined alignment
```

```
VSElastic1.AutoSizeChildren = 1    'Autosize children even horizontally
```

## CaptionPos Property

---

### Applies to

Elastic

### Description

Sets or returns the alignment of the caption text in the elastic box. The caption property is multi-line and it will proportionally align the text according to CaptionPos.

### Usage

```
[form.]VSElastic.CaptionPos = numericexpresion
```

### Setting Description:

- |   |  |
|---|--|
| 0 | (Default) Caption appears left-justified at the top. |
| 1 | Caption appears left-justified in the middle.        |
| 2 | Caption appears left-justified at the bottom.        |
| 3 | Caption appears centered at the top.                 |
| 4 | Caption appears centered in the middle.              |
| 5 | Caption appears centered at the bottom.              |
| 6 | Caption appears right-justified at the top.          |
| 7 | Caption appears right-justified in the middle.       |
| 8 | Caption appears right-justified at the bottom.       |

### Example

```
VSElastic1.CaptionPos = 0      '
```

```
VSElastic1.CaptionPos = 1      '
```

## BevelInner Property

---

### Applies to

Elastic

### Description

Determines the style of the Inner Bevel of the elastic box.

### Usage

```
[form.]VSElastic.BevelInner = numericexpresion
```

### Setting Description:

- |   |   |
|---|---|
| 0 | (Default) None. No inner bevel is drawn.  |
| 1 | Raised. The inner bevel appears raised off the screen.                                      |
| 2 | Raised Outlined. The inner bevel appears raised off the screen with a border as an outline. |
| 3 | Inset. The inner bevel appears inset into the screen.                                       |
| 4 | Inset Outlined. The inner bevel appears inset into the screen with a border as an outline.  |

### Example

```
VSElastic1.BevelInner = 0      ' No inner bevel  
VSElastic1.BevelInner = 3      ' Inner bevel inset
```

## BevelOuter Property

---

### Applies to

Elastic

### Description

Determines the style of the Outer Bevel of the elastic box.

### Usage

```
[form.]VSElastic.BevelOuter = numericexpresion
```

### Setting Description:

- |   |   |
|---|---|
| 0 | (Default) None. No outer bevel is drawn.  |
| 1 | Raised. The outer bevel appears raised off the screen.                                      |
| 2 | Raised Outlined. The outer bevel appears raised off the screen with a border as an outline. |
| 3 | Inset. The outer bevel appears inset into the screen.                                       |
| 4 | Inset Outlined. The outer bevel appears inset into the screen with a border as an outline.  |

### Example

```
VSElastic1.BevelOuter = 0      ' No outer bevel  
VSElastic1.BevelOuter = 3      ' Outer bevel inset
```

## BorderWidth Property

---

### Applies to

Elastic

### Description

Sets or returns the width of the border, the distance between the outer and inner bevels of the elastic box.

### Usage

```
[form.]VSElastic.BorderWidth = Width%
```

### Remarks:

The setting for this property determines the number of pixels between the inner and outer bevels that surround the elastic box.

Use this property in conjunction with the BevelInner, BevelOuter, BevelInnerWidth and BevelOuterWidth properties.

### Example

```
VSElastic1.BorderWidth = 5 ' sets borderwidth to 5
```



## FloodDirection Property

---

### Applies to

Elastic

### Description

Determines if and how the elastic is used as a status or progress indicator.

### Usage

[form.]VSElastic.FloodDirection = setting%

### Remarks:

**0 (Default) None.**

Elastic has no flooding capability and the caption (if any) is displayed.

**1 RIGHT.**

Elastic will be painted in the FloodColor from the right inner bevel to the left as the FloodPercent property increases.

**2 LEFT.**

Elastic will be painted in the FloodColor from the left inner bevel to the right as the FloodPercent property increases.

**3 UP.**

Elastic will be painted in the FloodColor from the bottom inner bevel upward as the FloodPercent property increases.

**4 DOWN.**

Elastic will be painted in the FloodColor from the top inner bevel downward as the FloodPercent property increases.

## FloodColor Property

---

### Applies to

Elastic

### Description

Sets or returns the color used to paint the area inside the elastic box's inner bevel when the elastic box is used as a status or progress indicator (that is, when the FloodDirection property setting is other than none).

### Usage

```
[form.]VSElastic.FloodColor = color&
```

### Remarks:

Use this property with FloodPercent and FloodDirection to cause the panel to display a colored status bar indicating the degree of completion of a task. At design time you can set this property by entering a hexadecimal value in the Settings box or by clicking the three dots that appear at the right of the Settings box. Clicking this button displays a dialog box that allows you to select a FloodColor setting from a palette of colors similar to the Visual Basic Color Palette window.

## FloodPercent Property

---

### Applies to

Elastic

### Description

Sets or returns the percentage of the painted area inside the elastic box's inner bevel when the elastic box is used as a status or progress indicator (that is, FloodDirection property setting other than none).

### Usage

```
[form.]VSElastic.FloodPercent = percent%
```

### Remarks:

The FloodPercent property can be set to an integer value between 0 and 100. Use this property in conjunction with FloodColor and FloodDirection to cause the elastic box to display a colored status bar, indicating the degree of completion of a task.

## Splitter Property

---

### Applies to

Elastic

### Description

Determines whether the user is allowed to resize Elastic child windows by dragging the bar between two child windows.

### Usage

[form.]VSElastic.Splitter = option%

### Remarks:

If Splitter is set to true, the Elastic will monitor the mouse at run time. Whenever the mouse passes over the space between controls inside the Elastic, the pointer changes to a resize icon and the user is allowed to drag the border between the adjacent controls to a new location.

This property is only active if AutoChildSpacing is set to Uneven Vertical or Uneven Horizontal.

## BorderInnerWidth Property

---

### Applies to

Elastic

### Description

### Usage

```
[form.]VSElastic.BorderInnerWidth = Width%
```

### Remarks:

The setting for this property determines the number of pixels of the inner bevel that surround the elastic box. Use this property in conjunction with the BevelInner, BevelOuter, BevelInnerWidth and BevelOuterWidth properties.

### Example

```
VSElastic1.BorderInnerWidth = 2    ' sets borderinnerwidth to 2
```

## BorderOuterWidth Property

---

### Applies to

Elastic

### Description

### Usage

```
[form.]VSElastic.BorderOuterWidth = Width%
```

### Remarks:

The setting for this property determines the number of pixels of the Outer bevel that surround the elastic box. Use this property in conjunction with the BevelInner, BevelOuter, BevelInnerWidth and BevelInnerWidth properties.

### Example

```
VSElastic1.BorderOuterWidth = 2    ' sets BorderOuterWidth to 2
```

## ChildSpacing Property

---

### Applies to

Elastic

### Description

### Usage

```
[form.]VSElastic.ChildSpacing = spacing%
```

### Remarks:

Sets or returns the spacing between child controls inside the Elastic.

The child spacing is measured in pixels.

This property is only relevant when AutoResizeChildren is set to a non- zero value.

### Example

```
VSElastic1.ChildSpacing = 6 ' sets ChildSpacing to 2
```

## RealignFrame Event

---

### Applies to

Elastic

### Description

Occurs whenever the Elastic is resized.

Sub CtlName\_RealignFrame ()



## ResizeChildren Event

---

### Applies to

Elastic

### Description

Fired when the Elastic's children are resized.

Sub CtlName\_ResizeChildren ()



## VideoSoft AWKsome Custom Control

---

The VideoSoft Awk control allows you to quickly scan and parse text files. It is the ideal tool for doing simple data manipulation -- changing its format, checking its validity, retrieving items, generating reports, and the like -- without having to write a lot of code to scan and parse files.

The VideoSoft AWKsome control is similar to the original AWK language in concept and purpose, but it is substantially different in syntax, since VSAwesome is embedded in Visual Basic. The most apparent omission is a pattern matching mechanism, which is provided by VB's Like operator.

### Properties

[\(About...\)](#)

[Action](#)

[Case](#)

[CurrPos](#)

[Error](#)

[F \(Field\)](#)

[FieldAt](#)

[Filename](#)

[FS \(Field Separator\)](#)

[L \(Line\)](#)

[NF \(Number of Fields\)](#)

[PercentDone](#)

[RN \(Record Number\)](#)

### Standard Visual Basic Properties

Index

Left

Name

Tag

Top

Width

### Events

[Begin](#)

[End](#)

[Error](#)

[Scan](#)

For Standard properties and events check your Visual Basic documentation.

## Action Property

---

### Applies to

AWKsome

### Description

Specifies an action to be performed by the VSAwksome control.

Valid actions are:

- 0 - Scan File
- 1 - Next Line
- 2 - Close File

The Scan File action causes VSAwk to take the following actions:

- 1) Open the file specified by the FileName property;
- 2) Fire the Begin event;
- 3) Read the file one line at a time, parsing each line and firing a Scan event for each one;
- 4) Close the file;
- 5) Fire the End event.

The Next Line action causes VSAwk to read the next line from the current data file. It is often useful in the VSAwk\_Scan subroutine. The CClose File action causes VSAwk to close the current file and stop scanning it.

## Case Property

---

### Applies to

AWKsome

### Description

Determines whether VSAwk should change the case of each line as it is read.

Valid setting for this property are:

- 0 - No Change
- 1 - Convert to Upper Case
- 2 - Convert to Lower Case

## Begin Event

---

### Applies to

AWKsome

### Description

Fired when VSAwk opens a file for scanning, after Action is set to 0 (Scan File).

### Syntax

Sub CtlName\_Begin ()

## **CurrPos Property**

---

### **Applies to**

AWKsome

### **Description**

Sets or returns the position of the current line within the file.

This property is useful when you want to save the position of current line so that you can quickly return to it later.

### **Usage**

```
[form.]VSAwk.CurrPos = numericexpresion&
```

## End Event

---

### Applies to

AWKsome

### Description

Fired when VSAwk is done scanning a file.

### Syntax

```
Sub CtlName_End ()
```

## Error Event

---

### Applies to

AWKsome

### Description

Fired when VSAwk encounters an error. See the Error property for a list of possible errors.

### Syntax

```
Sub CtlName_Error ()
```



## Error Property

---

### Applies to

AWKsome

### Description

Returns a code that identifies the last error trapped by VSAwk.

Possible error codes are:

- 0 - No Error
- 1 - Can't open file
- 2 - Line too long
- 3 - Field too long
- 4 - Invalid field index set
- 5 - Invalid field index get

The Can't open file error occurs when the file specified by the FileName property can't be found.

The Line too long error occurs when the file contains lines longer than 255 characters. In this case, VSAwk truncates the line.

The Field too long error occurs when you try to make the current line longer than 255 characters.

The Invalid field index set error occurs when you try to set the value of a field that doesn't exist, i.e., smaller than zero or greater than FS. The Invalid field index get error occurs when you try to get the value of a field that doesn't exist, i.e., smaller than zero or greater than FS.

### Usage

[form.]VSAwk.Error

## F (Field) Property

---

### Applies to

AWKsome

### Description

Sets or returns the value of a field within the current line (record).

Fields are numbered from 1 to NF. As in the AWK language, field 0 corresponds to the whole line (record).

If you try to set or read an invalid field, VSAwk fires the Error event.

If you set a field to a string that contains field separators, the line is automatically re-parsed, and the NF property is updated.

### Usage

```
[form.]VSAwk.F(index%) = itemstring$
```

### Example

```
VSAwk.F(0) = "This is a string"
Debug.Print VSAwk.NF, VSAwk.F(0)
4      This is a string
Debug.Print VSAwk.F(4)
string

VSAwk.F(4) = "plain long ASCII string"
Debug.Print VSAwk.NF, VSAwk.F(0)
7      This is a plain long ASCII string
```

## FileName Property

---

### Applies to

AWKsome

### Description

Sets or returns the name of the file to be scanned by the VSAwk control.

If VSAwk can't find the specified file, it fires the Error event.

### Usage

```
[form.]VSAwk.FileName [ = filename$ ]
```

## FieldAt Property

---

### Applies to

AWKsome

### Description

Returns the number of the field at the specified position.

This property was designed to be used with VB's InStr function. It is useful when you want to process a field based on its contents rather than on its position.

### Usage

```
[form.]VSAwk.FieldAt(position%)
```

### Example

```
VSAwk.F(0) = "This is a string"  
i% = InStr (VSAwk.F(0), "string")  
Debug.Print i%  
11  
Debug.Print VSAwk.FieldAt (11)  
4  
Debug.Print VSAwk.F(VSAwk.FieldAt (11))  
string
```

## FS (Field Separator) Property

---

### Applies to

AWKsome

### Description

Sets or returns the string of characters to be used by VSAwk as field separators.

The default value for this property is a string containing a space, a comma, and the tab character.

If you change this property, the current string is automatically re-parsed.

### Usage

```
[form.]VSAwk.FS [ = string$ ]
```

## L (Line) Property

---

### Applies to

AWKsome

### Description

Sets or returns the value of the current line (record).

Using this property is exactly the same as using the F(0) property.

### Usage

```
[form.]VSAwk.L [ = string$ ]
```

### Example

```
VSAwk.F(0) = "This is a string"  
Debug.Print VSAwk.L  
This is a string
```

## NF (Number of Fields) Property

---

### Applies to

AWKsome

### Description

Returns the number of fields in the current line (record).

The number of fields changes whenever the current line (L property) or field separators (FS property) change.

### Usage

[form.]VSAwk.NF

## PercentDone Property

---

### Applies to

AWKsome

### Description

Returns a number between 0 and 100 that indicates how much of the current file has been scanned.

This property was designed to give you an easy way to report progress while processing long files. It can be used, for example, with the VSElastic FloodPercent property.

### Usage

[form.]VSAwk.PercentDone



## **RN (Record Number) Property**

---

### **Applies to**

AWKsome

### **Description**

Returns the number of the current line (record) being scanned by VSAwk.

### **Usage**

[form.]VSAwk.RN

## Scan Event

---

### Applies to

AWKsome

### Description

Fired for each line in the input file while VSAwk scans a file. This is where you put the code that does the actual processing.

You may close the current file or get the next input line while in this subroutine by setting the Action property to 2 or 3, respectively.

### Syntax

```
Sub CtlName_Scan ()
```

Microsoft is a registered trademark of Microsoft Corporation.  
Windows, DOS, and Visual Basic are trademarks of Microsoft Corporation.

## Product Support

---

Product Support for VideoSoft Custom Controls is available to REGISTERED USERS through the following channels:

**Compuserve:**

CIS 71552,3052

**VideoSoft Tech Support:**

2625 Alcatraz Avenue, Suite 271  
Berkeley, California 94705  
(510) 547-7295

## **What do you get?**

---

- \* The latest release of VideoSoft Custom Control Library.**
- \* The developers license file that eliminates the VideoSoft banner when you load the controls.**
- \* More sample projects that show how to use the controls.**
- \* Product Support thru CompuServe, mail, or telephone.**
- \* Notification of new releases and special upgrades offers.**
- \* The satisfaction of helping us develop high-quality software for smart, honest people like you.**

## VideoSoft Product Registration

---

You can register the product directly from Compuserve

Type    GO SWREG  
Product Registration ID = 1084

To get a hard copy of the following form select file menu, Print Topic.

### VideoSoft Custom Controls

Please register my copy of VideoSoft Custom Controls, I am enclosing a check or money order for the amount of:  
\_\_\_\_\_ (CA residents add 8.25% sales tax)

Name: \_\_\_\_\_  
Company: \_\_\_\_\_  
Address1: \_\_\_\_\_  
Address2: \_\_\_\_\_  
City, State, ZIP: \_\_\_\_\_

Country: \_\_\_\_\_  
Phone: \_\_\_\_\_  
Fax: \_\_\_\_\_

### Where did you hear about VideoSoft Custom Controls?

\_\_\_\_\_

### Registration Fees

Single user	\$45.00
Site License 5 users	\$200.00
Site License 20 users	\$350.00
Site License 100 users	\$650.00

---

Amount enclosed \_\_\_\_\_  
(California Residents add 8.25% sales tax)  
ROYALTY FREE

### Send completed form with payment to:

VideoSoft  
2625 Alcatraz Avenue, Suite 271  
Berkeley, California 94705

## Tips and Tricks

---

Follow these tips for a more productive application development using VideoSoft Custom Controls.

### Elastic

\* To force all elastics in a form to resize themselves when the form size changes, insert the following line in your Form\_Resize subroutine:

```
VSElastic1.Left = 0
```

\* If you put several graphical controls in an Elastic, screen redraws may become slow or create flicker. To avoid this, put the graphical control alone in a new Elastic, then drop the new Elastic into the original one. The maximum number of children an Elastic can resize is 30. If you need more, use nested Elastics.

\* If you have an elastic with the Autosize children set to either even horizontally or even vertically, you can change the order of the controls by clicking on the control and dragging it to the position desired.

### IndexTab

\* An elastic can be used inside the indextab to resize controls.

\* An indextab can be used inside another indextab.

\* Use the & character to create shortcuts on the tabs.

### AWKsome

\* If you have a blank line in the file or string to be parsed, the awk is unable to parse it and an error event is trigger. Code any special behavior inside the error event.





