(F) **Visual FEDIT Help Contents**

Description

Visual FEDIT is a superclass of standard Edit control. It provides a variety of editing, validation and formatting capabilities. It supports 5 major styles of data: String, Number, Date/Time, Boolean and Text.   The behavior of each VFEDIT control is similar, even though there some   each data styles possesses a unique features.

File Name

VFEDIT.VBX, FEDIT.DLL

Object Type

VFEDIT

Remarks

In addition to its main purpose of input editing, validation and formatting,   VFEDIT provides many advanced features:

- Initialization;
- Easy set up of Fonts,
- Set up of Foreground and Background colors on the design level;
- Insert/Overwrite modes;
- Clipboard support.

Visual Basic Installation

To add Visual FEDIT into the Visual Basic Toolbox:

- Copy VFEDIT.VBX into your \WINDOWS\SYSTEM directory;
- Select **File/Add File...** command;
- Select VFEDIT.VBX and press OK;
- The VFEDIT.VBX icon

[F] should be visible in the Toolbox.

You can also add VFEDIT.VBX into the AUTOLOAD.MAK file using text editor.   In this case the VFEDIT.VBX will be loaded automatically.

Visual C++ Installation

To add Visual FEDIT into the Visual C++ AppStudio:

- Copy VFEDIT.VBX into your \WINDOWS\SYSTEM directory;
- Select **File/Install Controls...** command;
- Select VFEDIT.VBX into **Control Filename** and press OK;
- Make sure that VFEDIT.VBX is listed in the **Installed Files** list box;
- The VFEDIT.VBX icon
- should be visible in the Control Palette.

The information about VFEDIT custom control is kept in the APSTUDIO.INI file in the [VBX:VFEDIT.VBX;VFEDIT] section.

Distributing Application

There are several items that you have to consider when your application is completed and you create the executable (.EXE) file(s) and DLLs for distribution.

Visual FEDIT is a combination of kernel FEDIT.DLL and a custom control interface -VFEDIT.VBX. Both FEDIT.DLL and VFEDIT.VBX   must be distributed along with your application and installed either in the application's directory (assuming that this directory is included in the PATH) or in the \WINDOWS\SYSTEM directory. If the location of FEDIT.DLL can not be resolved the error message comes up:

FILE NOT FOUND : FEDIT.DLL.

Properties

About	Enabled	HelpContextID	**NoNulls***
Align	FontBold	Index	**ReadOnly***
AutoDrop*	FontItalic	**InitializeWith***	ScrollBars
AutoScroll*	FontName	**InitialText***	TabIndex
AutoValidate*	FontSize	Justification	TabStop
BackColor	FontStrikethru	Left	**Text***
CaseStyle*	FontUnderline	MousePointer	**ThickFrame***
DataField	ForeColor	**MultiLine***	Top
DataSource	**FormatString***	Name	**UserInfo***
DragIcon	**FormatStyle***		Width
DragMode	Height		

Events

Changed	DragDrop	GotFocus	**NullValue***
DblClk	DragOver	LostFocus	**ToggleIns***
			ValidationError*

Property: AutoDrop

Description

This property is used in conjunction with DragMode. When AutoDrop property is set to TRUE the VFEDIT control automatically respond to the DragDrop event. If the dropped control is also an FEDIT of the same data style, then its value is placed into the destination control. The data is formatted according to the rules of the destination control.

Usage

[*form*].**VFEDIT.AutoDrop**[= <*boolean*>]

Remarks

Setting	Description
0	(Default) FALSE
1	TRUE

Type

Integer (Boolean)

Property: AutoScroll

Description

Determines the use of automatic scroll feature. The AutoScroll feature is used when the full text of the control is lager then its visible portion. The text is automatically follows the cursor and makes the hidden parts of the text visible. The AutoScroll can be used instead of the ScrollBars. The most common Data Styles that are used with this property enabled is AlphaNumeric or Unformatted Text. When this property is enabled the HOME, END, PageUp, PageDown keys can be used to position within the full text string.   Please note, that if you use Vertical AutoScroll then the MultiLine property should set to TRUE.

Usage

[*form*].**VFEDIT.AutoScroll** [= *<integer>*]

Remarks

Setting	Description
0	(Default) Disable AutoScroll Feature
1	Horizontal
2	Vertical
3	Both

Type

Integer

Property: AutoValidate

Description

Determines whether to perform the content validation of the input field. If this option is enabled the control keeps the input focus until a valid data is entered. The input that is considered invalid depends on the Format Style:

Style	Description
AlphaNumeric	Empty string in the required positions of the mask
Date/Time	Invalid date or time

Usage

[*form.*]**VFEDIT.AutoValidate** [= *<boolean>*]

Remarks

This property forces a dialog or a window with an invalid data to be temporary system modal.

Setting	Description
False	(Default) No validation performed
True	Perform validation

Type

Integer (Boolean)

Property: CaseStyle

Description

Determines the case of the control's text. It is applicable when the Format Style property is set to AplhaNumeric or Unformatted Text.

Usage

[*form.*]**VFEDIT.CaseStyle** [= *<integer>*]

Remarks

Determines the font style and font attributes.

Setting	Description
0	(Default) Case insensitive
1	Upper case only
2	Lower case only

Type

Integer

Property: FormatString

Description

Determines the set of characters that are acceptable in this control. At the design time you specify the Formatting Mask that is appropriate for the Format Style. You can also set the Formatting Mask dynamically at the run time.

Usage

[*form.*]**VFEDIT.FormatString** [= *<string>*]

Remarks

Format Style	Mask	Description
AlphaNumeric	**A**	Any alphabetic character ('a'-'z' or 'A'-'Z' range).
	X	Any printable character.
	Z	Any alphanumeric character (combination of 'A' and '9'.
	a	Same as 'A' for *required* input.
	x	Same as 'X' for *required* input.
	z	Same as 'Z' for *required* input.
	9	Any digit (0 - 9).
	0	A *required* digit (0 - 9).
Numeric	**9**	Any digit (0-9). This character will be replaced with an actual digit.
	S	Represents a sign. Whenever user presses plus/minus key VFEDIT control will place the sign in the appropriate position.
	,	Represents a comma delimiter that usually separates thousands. A comma is not shown if all digits to the left of it are zeroes.
	.	Represents a decimal point. The decimal point separates the decimal part of the number from the whole one.
Date/Time	**mm**	A month in a decimal form, e.g. '08'.
	dd	A day of a month.
	yy	A year without century, e.g. '92' .
	yyyy	A year with century, e.g. '1992'.
	hh	An hour.
	xx	A minute.
	ss	A second.
	Mmm	A month in a case sensitive string form, e.g. 'Aug'.
	Www	A day of the week in a case sensitive string form, e.g. 'Tue'. The Day of the Week is set automatically according to the date value and cannot be modified by a user.
	am / AM **pm / PM**	Represents a part of the day. When FEDIT finds any of these strings it represent time in the 12 hours mode.
Boolean	**Y**	Represents Y/N (Y is default)
	YYY	Represents YES/NO (Yes is default)

N	Represents Y/N	(N is default)
NNN	Represents YES/NO	(No is default)
T	Represents T/F	(T is default).
TTTTT	Represents TRUE/FALSE	(TRUE is default).
F	Represents T/F	(F is default).
FFFFF	Represents TRUE/FALSE	(FALSE is default).

Note:

By using uppercase M, D, H, N, S in the Date/Time format style you can request control to use spaces instead of zeroes in the specified positions
(e.g. .for January 1, 1993   <Mm/Dd/Yy>   will be < 1/ 1/93> and
                                          <mm/dd/yy>   will be <01/01/93>.

All other characters can be used as literals.

Type
String

Property: FormatStyle

Description

Determines the Data Style of the control. Each Data Style has it's own set of Formatting characters that define the behavior.

Usage

[*form.*]**VFEDIT.FormatStyle** [= *<integer>*]

Remarks

Setting	Description
0	(Default) AlphaNumeric;
1	Numeric;
2	Date / Time;
3	Boolean;
4	Unformatted Text

Type

Integer

Property: InitializeWith

Description
The initial state of the control:

Value	Setting	Description
0	**Default value**	Initial value depends on the FormatStyle (see below);:
1	**Null value**	The NULL value causes FEDIT to suppress output of the Formatting mask and display an empty string;
2	**InitialText**	Display a text specified in the InitialText property.

Default value:

FormatStyle	Description
AlphaNumeric	String with delimiters in the specified positions. ???
Numeric	Zero.
Date / Time	Current date/time.
Boolean	TRUE for the 'Y' and 'T' formats, FALSE otherwise.
Unformatted Text	Empty string.

Type
Integer

Property: InitialText

Description
This property is used when **InitializeWith** property is set to InitialText.

Usage
[*form.*]**VFEDIT.InitialText** [= *<string>*]

Remarks
The InitialText value very much depends on the Format Style of the control. VFEDIT tries to make the most sense out of the data, but if the data is unrecognizable VFEDIT will set it to the default.

Type
String

Property: MultiLine

Description
Determines if control consist of multiple lines.

Usage
[*form.*]**VFEDIT.MultiLine** [= *<boolean>*]

Remarks

This property is valid only with FormatStyle property set to Unformatted Text.

Setting	Description
FALSE	(Default) A single line mode;
TRUE	A MultiLine mode.

Type
Boolean

Property: NoNulls

Description
Determines whether data is expected to be entered. This feature is useful for required fields.

Usage
[*form.*]**VFEDIT.NoNull** [= *<boolean>*]

Remarks
VFEDIT automatically checks the value of the control and doesn't allow user to remove focus from it until data is entered. This mechanism   forces a dialog or a window with an empty control to be temporary system modal.

Setting	Description
FALSE	(Default) Data is not required;
TRUE	A NoNull Mode - data is required..

Type
Boolean

Property: ReadOnly

Description

Makes the control Read only with no editing capabilities.

Usage

[*form.*]**VFEDIT.ReadOnly** [= *<boolean>*]

Remarks

VFEDIT automatically checks the value of the control and doesn't allow user to change data in the control.

Setting	Description
FALSE	(Default) The data in the control is editable;
TRUE	A Read Only mode - no editing is allowed.

Type

Boolean

Property: Text

Description

This property holds the string value of the control as it appears on the screen.

Usage

[*form.*]**VFEDIT.Text**[= *<string>*]
<string_variable> = [*form.*]**VFEDIT.Text**

Remarks

This property can be used to transfer data between VFEDIT controls.

Type

String

Property: ThickFrame

Description

The FEdit controls have a distinctive thick frame when conrol receives focus. In order to accomodate a uniform look and feel with other controls that may be used in your program you can disable this feature.

Usage

[*form.*]**VFEDIT.ThickFrame** [= *<boolean>*]

Remarks

This property is set for all FEdit controls in a window. You do not have to reset this property for every FEdit control.

Setting	Description
TRUE	(Default) Display Thick Frame.
FALSE	Disable Thick Frame

Type

Boolean

Property: UserInfo

Description

You can use this property to hold any information that pertains to the control..

Usage

[*form.*]**VFEDIT.UserInfo** [= *<string>*]

Type

String

Event: NullValue

Description

Occurs when an VFEDIT control with **NoNulls** property set to **TRUE** is about to loose input focus and the control's **Text** property value is still a NULL (either as a result of the initialization to NULL or user pressing the keys   CTRL-DEL). The VFEDIT will retain the focus until a key (other then TAB key) is pressed to populate the Text property.

Syntax

Sub *VFEditControlName*_**NullValue()**

Event: ToggleIns

Description

The Override is the default mode of operation when an VFEDIT gets the input focus. This event occurs when user presses an INSERT key. Based on   standard behavior of the VFEDIT control, the cursor is changed from blinking bar into the non-blinking bold bar. You can override or extend this behavior in this subroutine.

Syntax

Sub *VFEditControlName***_ToggleIns()**

Event: ValidationError

Description

This event occurs when an VFEDIT control is either AlphaNumeric or Date/Time under the following conditions:

Style	Description
AlphaNumeric	Empty string in the required positions of the mask
Date/Time	Invalid date or time

Syntax

Sub *VFEditControlName*_**ValidationError()**

Methods

FVBSetDouble
 Sets the Text property of the FormattedEdit control using a double value.

FVBGetDouble
Retrieves the value of the Text property as a double-precision number.

FVBSetLong
Sets the Text property of the FormattedEdit control using a long value.

FVBGetLong
Retrieves the value of the text property as a long value.

FVBGetString
Retrieves the value of the text property as an unformatted string.

FVBSetString
Sets the value of the text property using an unformatted string.

FVBGetStringAs
Retrieves the value of the text property in the specified format.

FVBSetStringAs
Sets the value of the text property using a string in the specified format.

FVBGetValue
Retrieves the value of the text property.

FVBSetValue
Sets the value of the text property.

FVBSetNull
Sets the value of the control to NULL.

FVBIsNull
This function tests the value of the control.

FVBTranslateString
The **FVBTranslateString** function translates the value of the *sourceString* parameter in the *sourceMask* format into *destMask* format.