

# Introduction to SpellText

SpellText is a royalty-free OCX (also known as an ActiveX component, or control) that provides all of the functionality of a standard text box, along with spell checking using its internal dictionary of roughly 112,000 English words. You can have any number of custom dictionaries, and all of the standard spell checking functionality (ignore, change, change all, etc) is available. It can easily be used in a Visual Basic program, a Visual C++ program, Microsoft Access, or even in a web page.

The control is extremely fast (checking over 8000 characters of text per second on a Pentium 150) and very efficient taking up only about 800K of disk space.

## What Can I Use SpellText For?

SpellText can be used instead of a standard text box in any new or existing project. You can use it any place you want to have spell checking. As a matter of fact, replacing your current standard text boxes with SpellText is just as simple as deleting the old control and dropping in SpellText.

## Using SpellText in Visual Basic

Using SpellText in a Visual Basic program is easy. Make sure the control is installed on your system (it should be already, since you're reading this document). An OCX can be installed anywhere on your system, although the usual spot for them is in the \Windows\System (\Windows\System32 on Windows NT) directory.

Open up your Visual Basic project, and go to **Components** (in VB5, it's under the **Project** menu). Find the "ChadoSpellText" control, make sure it has a check mark next to it - then hit OK. This adds the component to your toolbox, so you can drag and drop it on your form. The control looks like a yellow text box.

Using the control is simple use it exactly as you would use a text box. SpellText gives you lots of different options you can have it automatically check the spelling when it loses focus (i.e., you TAB out of it, or click on another control on the form), or you can control the checking programmatically. During the checking, you can have the control handle everything including checking as the user types, or waiting for the focus to be lost and then showing the spelling error and offering suggested corrections, or you can have it raise an event in your program, where you can handle the error. Lastly, you can have the control notify the user when the spell check is done, you can raise an event in your code, or you can have it do nothing. The default behavior of the control is to use passive checking.

You can use the control exactly as you would a text box including binding it to a field in a database.

## Using SpellText in Visual C++

Adding SpellText to one of your forms in Visual C++ is straightforward. Simply right-click on the form you wish to add SpellText to, and select Insert ActiveX control from the pop-up menu. Choose the SpellText control from the list of available controls. Then simply resize and position the control as you wish. Right-click on it to get access to its properties, events, and methods.

## Using SpellText in MS Access

Using SpellText in Access is also very easy. Open up the form you want to have SpellText on, and click on Other Controls in the toolbox. This will bring up a list of ActiveX controls on your system. ChadoSpellText should be one of them. Choose that, and place it on your form. Then use it as you would a regular text box. If you go into the properties of the control, you can set all of the standard properties, plus some properties that are specific to SpellText, like checking words that are ALL CAPS, etc.

## Using SpellText in a Web Page

Using SpellText in a web page requires a bit of work. First of all, currently the only browser that supports ActiveX controls of any kind is Internet Explorer. There are third party plug-ins available for Netscape that allow the use of ActiveX controls, but for native support, you are limited to IE.

Embedding an ActiveX control in a web page isn't too difficult - you just use the <OBJECT> tag. Part of that object tag is a URL where the browser can find the control, in case the person viewing the page doesn't already have it installed on their machine. The control and its accompanying support files are packaged up in a CAB file (short for "Cabinet"). We include a CAB file for SpellText in the registered version of the product.

For a detailed example of using SpellText in a web page, visit our web page at <http://www.chado-software.com>.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Packaging SpellText With Your Programs

When you are packaging your program, include the file **ChadoSpellText.OCX**, and be sure it gets registered during installation (see the directions for your installation creation program for details about this). Additionally, the file **MSVBVM60.DLL** must be included for run-time support. This file is required by all programs written in VB 6.0 programs, so it should be included if your program was written in that language. If your program was written in another language, you will need to manually include this file in your project. You will also need to include the file **ChadoSpell.DLL**, which must be installed in the standard Windows or Windows\System directory.

**NOTE:** Do not include your license file (**SpellText.LIC**) in your final distribution! SpellText will work perfectly on a users machine without the license file. If you distribute the license file, you are illegally giving away a copy of the control. You are allowed only to distribute the run-time portion of the control - i.e., the OCX file itself.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Miscellaneous Issues

## **Custom Dictionaries**

You can have as many custom dictionaries as you like. When you set the CustomDictionaryFName property to a fully qualified file name, it saves the current custom dictionary (if it has been changed) and loads the specified one (if it exists). The custom dictionary is always saved when the program using the SpellText control terminates.

## **Multiple Instances on the Same Form**

You can have multiple instances of SpellText on the same form. All SpellText controls included in the same program share the same custom dictionary, ignore-all list, and change-all list. Therefore, memory usage is kept at a minimum, and speed is kept at a maximum. Instances of SpellText in separate programs do not share any data.

## **Standard Dictionary Contents**

Weve done our best to provide you with a useful dictionary. Along with most common English words, the standard dictionary also has some common names in it - typically things like names of states, countries, religions, religious figures, common first and last names, etc. Our goal is to provide you with a control that catches spelling errors, while at the same time allows your users to type naturally without having to stop all of the time to add words to the custom dictionary. Please contact us if you feel you have found words that belong in the dictionary.

# AddWordsToCustom Method

## **AddWordsToCustom (sNewWords As String)**

This method allows you to add one or more words to the custom dictionary. All words passed in must be separated by blanks, tab characters, or line breaks. Words that are placed in the custom dictionary are saved when the control is unloaded. Custom dictionaries are loaded at the time the program using the control is run.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# CheckSpellingNow Method

## **CheckSpellingNow()**

This method causes the SpellText control to immediately check the spelling of the current text. You can use this instead of, or in addition to, the CheckOnFocusLoss property.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# CheckString Method

**Checkstring (sToCheck as string)**

This method allows you to check the spelling of any arbitrary string. The string can have as many words in it as you like, separated by white space (spaces, tab characters, and/or line breaks).

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# IgnoreAll Method

## **IgnoreAll (sToIgnore As String)**

This method allows you to add one or more words to the ignore-all list. All words passed in must be separated by blanks, tab characters, or line breaks. Words that are placed in the ignore-all list will always be ignored during spell checking.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# SpellText Help

[Introduction to SpellText](#)

[Miscellaneous Issues](#)

[Packaging SpellText With Your Programs](#)

[Technical Support](#)

## Methods

[AddWordsToCustom](#)

[CheckSpellingNow](#)

[CheckString](#)

[ClearIgnoreList](#)

[IgnoreAll](#)

[LinkPoke](#)

[LinkRequest](#)

[LinkSend](#)

[OLEDrag](#)

[Refresh](#)

## Events

[DebugEvent](#)

[SpellCheckDone](#)

[SpellingError2](#)

[Change](#)

[Click](#)

[DbtClick](#)

[KeyDown](#)

[KeyPress](#)

[KeyUp](#)

[MouseDown](#)

[MouseMove](#)

[MouseUp](#)

[OLECompleteDrag](#)

[OLEDragDrop](#)

[OLEDragOver](#)

[OLEGiveFeedback](#)

[OLESetData](#)

[OLEStartDrag](#)

[Validate](#)

## Properties

AllowCancel  
AutoHandleErrors  
CheckOnFocusLoss  
CustomDictionaryFName  
DebugOption  
DoneAction  
EnteredText  
IgnoreAllCaps  
IgnoreInternetAddrs  
IgnoreWithNumbers  
MisspellingBackColor  
MisspellingForeColor  
PassiveCheckColor  
PassiveChecking  
Style  
SuggestCorrections

BackColor  
CausesValidation  
Enabled  
Font  
FontBold  
FontItalic  
FontName  
FontSize  
FontStrikeThru  
FontUnderline  
ForeColor  
hWnd  
LinkItem  
LinkMode  
LinkTimeout  
LinkTopic  
Locked  
MaxLength  
Mouselcon  
MousePointer  
OLEDragMode  
OLEDropMode  
PasswordChar  
RightToLeft  
SelLength  
SelStart  
SelText  
Text

ToolTipText  
WhatsThisHelpID

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# LinkPoke Method

## **LinkPoke()**

This method transfers the contents of the control to the source application via DDE.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# LinkRequest Method

## **LinkRequest()**

This method requests a destination application to update the contents of the control via DDE.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# LinkSend Method

## **LinkSend()**

This method transfers the contents of the control to a destination application via DDE.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLEDrag Method

## **OLEDrag()**

This method causes the control to initiate an OLE drag/drop operation.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Refresh Method

## **Refresh()**

This method causes the contents of the text box to be immediately redrawn. Generally, painting the control will be done automatically, but there may be circumstances where you want to force a refresh of the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# SpellCheckDone Event

## **SpellCheckDone ()**

This event is raised when the spell checking for a control finishes, if the DoneAction property was set to FIRE\_DONE\_EVENT. If the DoneAction property was set to SHOW\_MSG\_BOX or DO\_NOTHING, then the event will not be raised.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# SpellingError2 Event

(SpellingError is obsolete - do not use)

**SpellingError2 (sBadWord As String, sSuggestions As String, OffsetOfWord As Long, WhatToDo As eWhatToDo)**

This event is raised for each misspelled word, but only if the AutoHandleErrors property is set to False. The parameters are as follows:

**sBadWord** is the misspelled word. If you decide to change the word to another, you should put the new word into this string to pass it back to the control. This value will only be examined if WhatToDo is set to either CHANGE\_WORD or CHANGE\_ALL.

If the SuggestCorrections property is True, then **sSuggestions** is a list of suggestions, separated by ASCII character 13 (Return). If the SuggestCorrections property is False, then sSuggestions will be an empty string.

**OffsetOfWord** is the numeric offset of the misspelled word in the string being checked (starting at 0).

**WhatToDo** allows you to specify how the control should handle the spelling error. It can be set to one of the following values:

- IGNORE\_THIS\_ONE, which means that this single instance of the misspelled word should be ignored, but other occurrences should be flagged as misspelled.
- IGNORE\_ALL, which means that the word will be added to the ignore-all list, so all future occurrences of the word (throughout the life of the program) will be ignored.
- ADD\_TO\_CUSTOM causes the word to be added to the custom dictionary.
- CHANGE\_WORD means that the word should be changed, and sBadWord will be set to the new word. This change will only occur once.
- CHANGE\_ALL will change the word to the value in sBadWord, and will force this change to occur for the rest of the life of the program.
- CANCEL\_CHECKING stops the checking immediately.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Change Event

## **Change ()**

This event is raised when the contents of the SpellText control are changed in any way.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Click Event

## **Click()**

This event is raised when the SpellText control is clicked on.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# DbClick Event

## **DbClick()**

This event is raised when the SpellText control is double clicked on.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# KeyDown Event

**KeyDown(Keycode As Integer, Shift As Integer)**

This event is raised when a key is pressed while the SpellText control has focus. KeyCode returns a value for the key pressed, and Shift returns a value for what combination of the shift(1), alt (4), and control (2) keys are pressed.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# KeyPress Event

## **KeyPress (KeyAscii As Integer)**

This event is raised when a key is pressed while the SpellText control has focus. KeyAscii returns the ASCII value for the key.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# KeyUp Event

## **KeyUp (Keycode As Integer, Shift As Integer)**

This event is raised when a key is released after it has been pressed while the SpellText control has focus. KeyCode returns a value for the key pressed, and Shift returns a value for what combination of the shift(1), alt (4), and control (2) keys are pressed.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# MouseDown Event

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

This event is raised when the mouse is pressed in the SpellText control. Button returns a value for which mouse buttons were pressed (1 for left, 2 for right, and 4 for middle). Shift returns a value for what combination of the shift(1), alt (4), and control (2) keys are pressed. X and Y are the offsets within the SpellText control that the mouse was pressed. These values are expressed in the Scale units of the container of the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# MouseMove Event

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

This event is raised when the mouse is moved over the SpellText control. Button returns a value for which mouse buttons were pressed during the movement (1 for left, 2 for right, and 4 for middle). Shift returns a value for what combination of the shift(1), alt (4), and control (2) keys are pressed. X and Y are the offsets within the SpellText control that the mouse is currently at. These values are expressed in the Scale units of the container of the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# MouseUp Event

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

This event is raised when the mouse is released after being pressed in the SpellText control. Button returns a value for which mouse buttons were pressed (1 for left, 2 for right, and 4 for middle). Shift returns a value for what combination of the shift(1), alt (4), and control (2) keys are pressed. X and Y are the offsets within the SpellText control that the mouse was pressed. These values are expressed in the Scale units of the container of the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLECompleteDrag Event

## **OLECompleteDrag (Effect As Long)**

Occurs when a SpellText control is dropped onto a target component, informing it that a drag action was either performed or canceled. If Effect is 0, this cancels the drop event. If Effect is 1, then the SpellText control is copied to the destination component. If Effect is 2, then the SpellText control is moved to the destination component.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLEDragDrop Event

**OLEDragDrop (Data As DataObject, Effect As Long, Button As Integer, Shift As Integer, X As Single, Y As Single)**

Occurs when a SpellText control is dropped onto a target component when the control determines that a drop can occur. This only occurs if the OLEDropMode property of the control is set to 1. Data is an object of type DataObject that contains information about formats that the control will provide, and, in some cases, actual data. Effect has the same values as in the OLECompleteDrag event. Button, Shift, X, and Y have the same values that they do in the MouseDown event.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLEDragOver Event

**OLEDragOver (Data As DataObject, Effect As Long, Button As Integer, Shift As Integer, X As Single, Y As Single, State As Integer)**

Occurs when a SpellText control is dragged over a target component. Data is an object of type DataObject that contains information about formats that the control will provide, and, in some cases, actual data. Effect has the same values as in the OLECompleteDrag event. Button, Shift, X, and Y have the same values that they do in the MouseDown event. State is either 0 (meaning the control is entering the target), 1 (meaning the control is leaving the target), or 2 (meaning that the control is currently over the target.)

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLEGiveFeedback Event

## **OLEGiveFeedback (Effect As Long, DefaultCursors As Boolean)**

Occurring after every OLEDragOver event, this event allows the SpellText control to provide visual feedback to the user. Effect has the same values as in the OLECompleteDrag event. If DefaultCursors is True, then the default cursor will be used. If it is False, then the screens mouse pointer will be used.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLESetData Event

**OLESetData (Data As DataObject, DataFormat As Integer)**

Occurs on a SpellText control when a target component performs the GetData method on the controls DataObject object, but the data for the specified format has not yet been loaded. Call SetData on the DataObject to fill in the data, and set DataFormat to the correct format of the data you are sending.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# OLEStartDrag Event

**OLEStartDrag(Data As DataObject, AllowedEffects As Long)**

Occurs when the SpellText controls OLEDrag method is performed, or when an OLE drag/drop operation is started when the OLEDragMode property is set to Automatic. This event specifies the data formats and drop effects that the SpellText control supports. It can also be used to insert data into the DataObject object.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# AutoHandleErrors Property

## **AutoHandleErrors As Boolean**

*Default Is True*

This property controls whether the SpellText control will handle spelling errors or not. If the value of this property is True, then the control will completely handle all spelling errors. If the value of this property is False, then the control will pass all misspelled words to the SpellingError2 event.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# CheckOnFocusLoss Property

## **CheckOnFocusLoss As Boolean**

*Default Is False*

This property controls whether the SpellText control will check for spelling errors when the control loses focus. If the value of this property is True, then the control will check spelling as soon as another control on the parent form is clicked on, or the Tab key is used to move to another control. If the value of this property is False, then the control will only check for spelling errors when the program issues a CheckSpellingNow method call.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# CustomDictionaryFName Property

## **CustomDictionaryFName As String**

*Default Is CHCUSTOM.DIC*

This property controls the name of the custom dictionary. When you set the name of this control, any current custom dictionary will be saved, and if the property is the name of an existing file, the custom dictionary will be loaded. Note that since all instances of SpellText controls in the same project share the same custom dictionary, changing this property for one control changes it for all of the controls.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# DoneAction Property

## **DoneAction As eDoneAction**

*Default Is DO\_NOTHING*

This property controls the action that the SpellText control takes after a spell check is complete. If the value is DO\_NOTHING, then nothing will be done after a spell check is complete. If the property value is FIRE\_DONE\_EVENT, then the SpellCheckDone event will be raised at the end of the checking. If the property value is SHOW\_MSG\_BOX, the control will show a simple message box that states that the spell check is complete.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# IgnoreAllCaps Property

**IgnoreAllCaps As Boolean**

*Default Is True*

This property controls whether the SpellText control will ignore words that are spelled with all capital letters.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# IgnoreInternetAddrs Property

**IgnoreInternetAddrs As Boolean**

*Default Is True*

This property controls whether the SpellText control will ignore words that are Internet web or e-mail addresses.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# IgnoreWithNumbers Property

**IgnoreWithNumbers As Boolean**

*Default Is True*

This property controls whether the SpellText control will ignore words that have numeric digits in them.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# SuggestCorrections Property

## **SuggestCorrections As Boolean**

*Default Is True*

This property controls whether the SpellText control will generate a list of suggestions for each misspelled word. The generation of the words takes a small amount of time, which is why you may wish to disable it.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# BackColor Property

**BackColor As OLE\_COLOR**

*Default Is White*

This property controls the background color of the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Enabled Property

**Enabled As Boolean**

*Default Is True*

This property controls whether the SpellText control is enabled.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Font Property

**Font As Font**

*Default Is MS Sans Serif*

This property controls the font that the SpellText control uses.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# FontBold Property

**FontBold As Boolean**

*Default Is False*

This property controls whether font that the SpellText control uses is displaying characters in bold.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# FontItalic Property

**FontItalic As Boolean**

*Default Is False*

This property controls whether font that the SpellText control uses is displaying characters in italics.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# FontName Property

## **FontName As String**

*Default Is "MS Sans Serif"*

This property returns or sets the name of the font that the SpellText control is using.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# FontSize Property

**FontSize As Single**

*Default Is 8.25*

This property returns or sets the size of the font that the SpellText control is using.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# FontStrikeThru Property

## **FontStrikeThru As Boolean**

*Default Is False*

This property controls whether font that the SpellText control uses is displaying characters that have strikethru lines in them.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# FontUnderline Property

## **FontUnderline As Boolean**

*Default Is False*

This property controls whether font that the SpellText control uses is displaying characters that are underlined.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# ForeColor Property

**ForeColor As OLE\_COLOR**

*Default Is Black*

This property controls whether the color of the text that the SpellText control will display.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# hWnd Property

**hWnd As Long**

*No Default Value*

This property returns the handle to the window of the underlying text control for the given SpellText control. You can use this value in conjunction with Win32 calls.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# LinkItem Property

## **LinkItem As String**

*No Default Value*

This property returns or sets the data passed to a destination control in a DDE conversation with another application.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# LinkMode Property

## **LinkMode As Integer**

*Default Is 0*

This property returns or sets the type of link used for a DDE conversation. It allows a destination control on a Visual Basic form to initiate a conversation, as specified by the control's LinkTopic and LinkItem properties.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# LinkTimeout Property

**LinkTimeout As Integer**

*Default Is 50*

This property returns or sets the amount of time the control waits for a response to a DDE message.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# LinkTopic Property

## **LinkTopic As String**

*No Default Value*

This property is used during the establishment of a DDE conversation from another application to the SpellText control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# Locked Property

**Locked As Boolean**

*Default Is False*

If this property is True, then the user will not be able to enter data into the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# MaxLength Property

## **MaxLength As Long**

*Default Is 0*

This property controls the maximum number of characters that a user can enter into a SpellText control. If it is set to 0, the actual value is somewhere around 65,000 characters. Note that you can use the clipboard to paste in any amount of text - the limit of 65,000 characters is only applied when you type in the characters, or set the Text property from inside of your code.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# MouseIcon Property

## **MouseIcon As Picture**

*No Default Value*

This property controls the icon the mouse should display when it is over the SpellText control. The value must be loaded using the Visual Basic LoadPicture function.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# MousePointer Property

## **MousePointer As Integer**

*Default Is 0*

This property sets the mouse icon to one of a standard set when the mouse moves over the SpellText control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLEDragMode Property

## **OLEDragMode As Integer**

*Default Is 0*

If this property is set to 0, then the OLE drag mode is manual, meaning that the program must handle all OLE drag and drop operations. If the property is 1, the OLE drag mode is automatic and handled by the system.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# OLEDropMode Property

## **OLEDropMode As Integer**

*Default Is 0*

This property returns or sets how the SpellText control handles drop operations. If it is set to 0, then the control doesn't accept OLE drops. If it is set to 1, then the drop mode is manual, meaning that the OLE drop events will fire. If the property is set to 2, then the drop mode is automatic, and if the data is in a valid format, it will be placed into the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# PasswordChar Property

## **PasswordChar As String**

*No Default Value*

This property controls whether the SpellText control should display the characters that the user is typing, or a special password mask character. The usual value for password boxes is \*.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# RightToLeft Property

## **RightToLeft As Boolean**

*Default Is False*

This property controls whether the SpellText control will display letters from right to left. It is designed primarily for countries where the text is entered in this manner.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# SelLength Property

**SelLength As Long**

*No Default Value*

This property returns or sets the number of characters selected in the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# SelStart Property

**SelStart As Long**

*No Default Value*

This property returns or sets the starting point of selected text. If no text is selected, it indicates the position of the insertion point.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# SelText Property

## **SelText As String**

*No Default Value*

This property controls returns or sets the currently selected text. If no text is selected, then it is a zero-length string.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Text Property (EnteredText Property)

## **Text As String**

*Default Is "Text1"*

This property returns or sets the text displayed in the control.

The EnteredText property is provided for those environments that prevent the use of the word "Text" as a property. It is exactly the same as the Text property.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# ToolTipText Property

## **ToolTipText As String**

*No Default Value*

This property returns or sets the tool tip for the control.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# WhatsThisHelpID Property

**WhatsThisHelpID As Long**

*No Default Value*

This property returns or sets the context ID for a help file.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Technical Support

Free support for SpellText is provided through our web page. Check there for the latest versions, frequently asked questions, and our e-mail address.

**[www.chado-software.com](http://www.chado-software.com)**

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# ClearIgnoreList Method

## **ClearIgnoreList()**

This method clears out the list that SpellText keeps of all of the words that the user has said "Ignore All" to.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# AllowCancel Property

**AllowCancel As Boolean**

*Default Is True*

This property controls whether the SpellText control will display the "Cancel" button on the corrections form that is displayed whenever a spelling error occurs, and the AutoHandleErrors property is set to true.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# DebugOption

## DebugOption As eDebugOptions

*Default Is NO\_DEBUGGING*

This property controls the type of debug events the SpellText control fires.

If the property is NO\_DEBUGGING, then no DebugEvent events will be fired.

If the property is DEBUG\_ACTIONS, the DebugEvents relating to the actions taken by the control will be fired. These include messages about each word being checked, as well as other processing being done by the control.

If the property is DEBUG\_PERFORMANCE, then a DebugEvent will be fired at the end of each spell check, with the total amount of time taken by the checking. Note that this time includes any time taken by a user correcting words through the corrections dialog box.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# MisspellingForeColor

**MisspellingForeColor As OLE\_COLOR**

*Default Is Red*

This property controls the color of any misspelled text that the SpellText control finds. It is used in both the SpellText control and the Corrections dialog box.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# MisspellingBackColor

**MisspellingBackColor As OLE\_COLOR**

*Default Is White*

This property controls the background color of any misspelled text that the SpellText control finds. It is used in both the SpellText control and the Corrections dialog box.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# DebugEvent

## **Event DebugEvent (ByVal DebugInfo As String)**

This event is raised for each debug event. See the DebugOption property for information about controlling which debugging information is given through the event. The DebugEvent is provided as a tool for an application programmer. It gives you the ability to find out what the control is doing, as well as the speed of the spell checking.

**DebugInfo** is a string with the debug information.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Validate Event

## **SpellText1\_Validate(KeepFocus As Boolean)**

This event is raised when focus leaves the control. It is typically used to check the validity of the data in the control at the time that a user tabs away or mouse-clicks away from it. Note that you must set the CausesValidation property to TRUE to allow this event to be fired.

**KeepFocus** allows you to control whether the user is allowed to change focus. Set KeepFocus to TRUE until the user has entered valid data.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# Style Property

## **Style as eStyles**

*Default is 0 (MultiLine).*

This property controls the appearance of the SpellText control.

If the value is 0 (MultiLine), then the control will handle carriage returns, activating a vertical scroll bar when necessary.

If the value is 1 (SingleLine), then the control only displays a single line, scrolling to the right as needed.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# PassiveChecking

## **PassiveChecking as Boolean**

*Default is TRUE*

This property controls whether passive (i.e., during typing) checking is done. When this property is TRUE, then a squiggly line will be drawn under each word that is not found. The line's color is controlled by the PassiveCheckColor property.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



# PassiveCheckColor

**PassiveCheckColor as OLE\_COLOR**

*Default is Red*

This property controls the color of the squiggly line underneath misspelled words when the PassiveChecking property is TRUE.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).

# CausesValidation

## **CausesValidation as Boolean**

*Default is TRUE*

This property controls whether a Validation event is fired on the control when focus is lost.

Contents Copyright © 1999 by Chado, Inc. See our web page at [www.chado-software.com](http://www.chado-software.com).



