

# WebSpellChecker White Paper

---

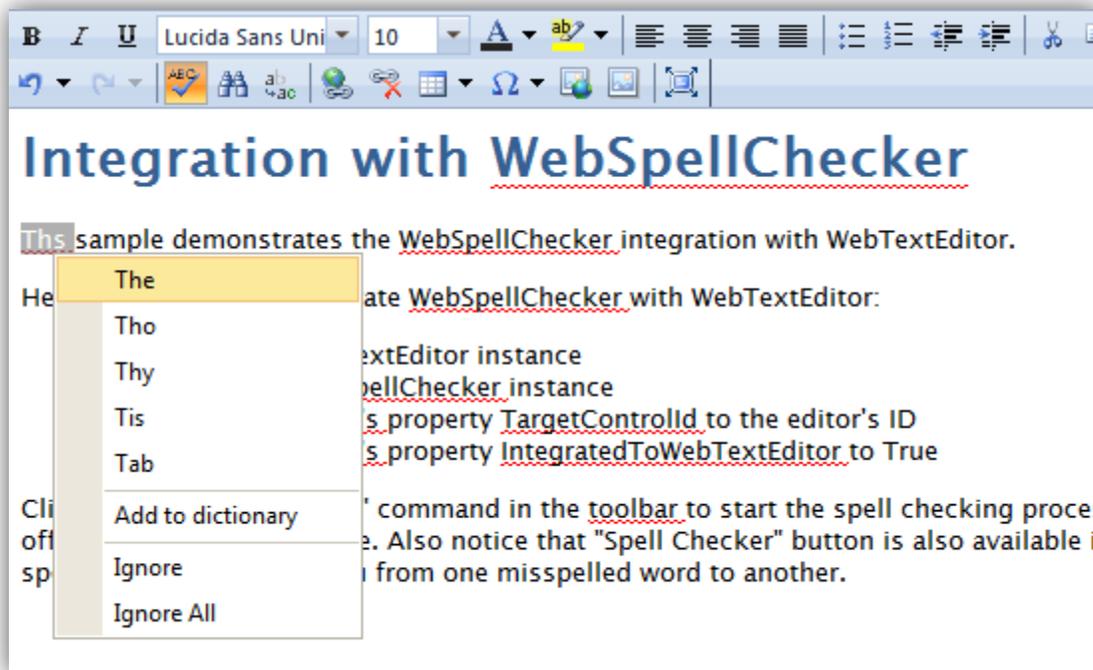
This whitepaper describes the concept and features introduced in WebSpellChecker.

## Table of Contents

Overview.....	2
User Interface.....	2
Maximum text suggestion .....	3
WebSpellChecker options.....	4
Change .....	4
Add to dictionary.....	4
Ignore.....	5
Ignore All.....	5
Use WebSpellChecker with other controls.....	6
TargetControlId .....	6
TargetControlIdValue .....	6
Images settings.....	6
Text settings.....	7
Client side events .....	7
OnInitialize(.....	7
OnBeforeCheckSpell(.....	7
OnAfterCheckSpell( .....	7
OnContextMenu(.....	7
OnClickContextMenuItem( .....	7
OnExitSpellCheck(.....	7
OnReplaceWord(.....	7

## Overview

Intersoft WebSpellChecker is an easy-to-use, yet powerful spell checking component for ASP.NET. In addition to typical spell check features that you expected, WebSpellChecker delivers several innovative features that are not available elsewhere, such as Microsoft Word-style red-wave underline highlight, built-in dialogbox with intuitive design, natural integration with WebTextEditor, wide cross browsers support and more.

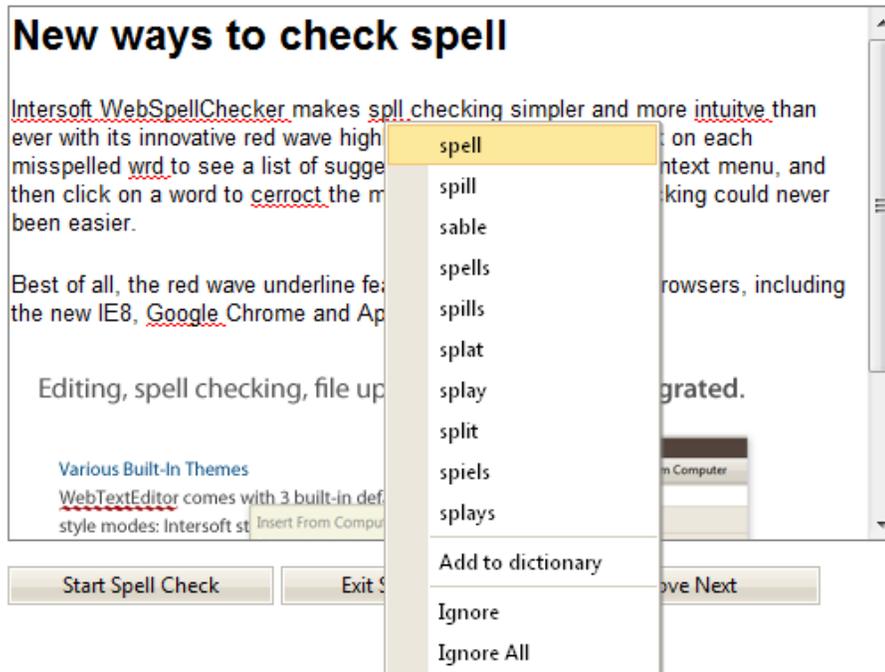


## User Interface

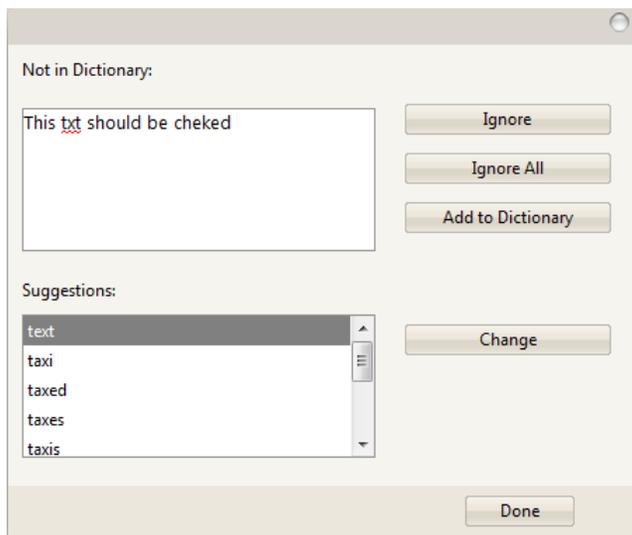
Intersoft WebSpellChecker is the first spell checker component in the industry to provide two modes of spell checking user interface.

1. Red-wave Underline Highlight  
WebSpellChecker introduces more natural, intuitive way to perform spell checking with red wave highlight feature. To correct misspelled words, just right click on each word and choose the correct word from the displayed context menu. It makes spell checking faster and easier than ever.

The red wave highlight feature can only be enabled on editable IFRAME.



## 2. Sleek Dialog Box Interface



This mode will be automatically enabled when the target control is not an editable IFRAME. The dialog box interface includes visual elements that display checked words and the suggestion words list. Just double click on a word in suggestion list or click on the *Change* button to correct the misspelled word with the selected suggested word, and finally click on *Done* button to accept all changes and close the dialog box.

## Maximum text suggestion

You can configure the number of maximum text suggestion for each misspelled word by specifying the *MaximumTextSuggestion* property to an integer value. The default value of maximum text suggestion is 10.

## WebSpellChecker options

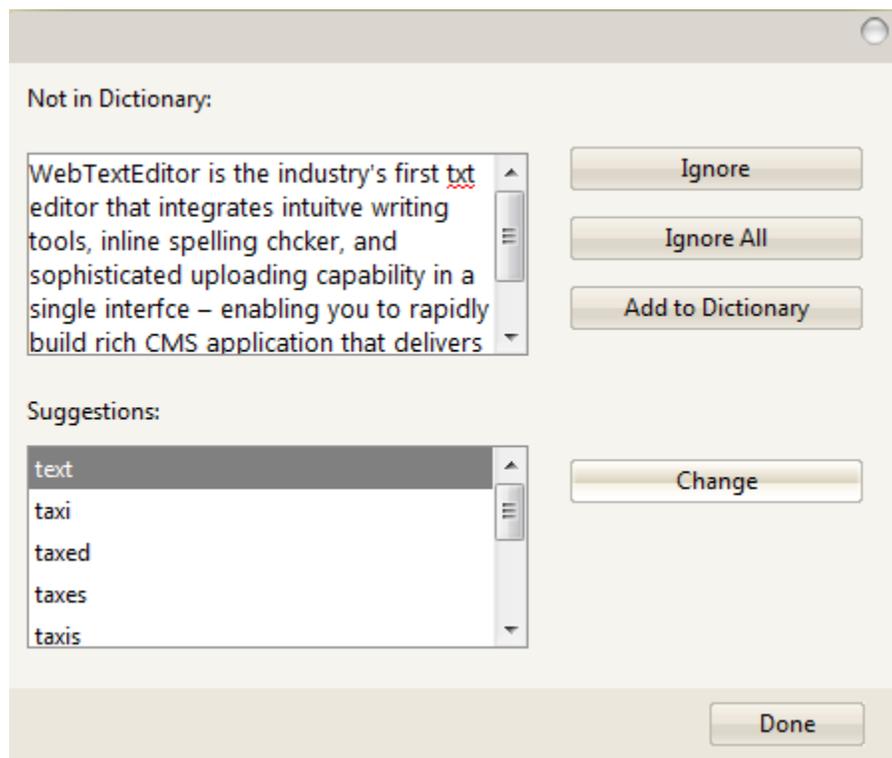
User interaction on WebSpellChecker can be done through several options available in the context menu interface or in WebDialogBox interface.

Here are the options:

### Change

This option is only available in WebDialogBox mode; it is used to change the selected misspelled word with the selected suggested word.

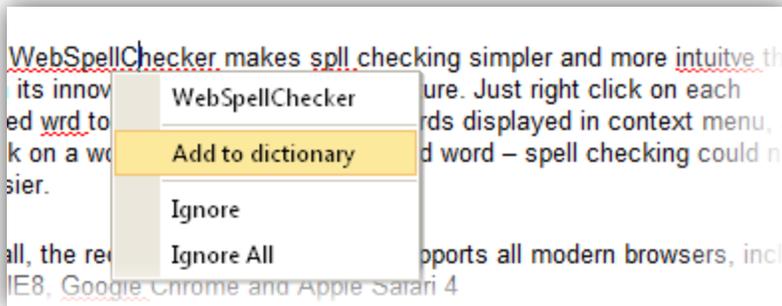
Alternatively, you can correct the misspelled word by simply double clicking on the suggested word.



### Add to dictionary

Often times, WebSpellChecker may show specific terms such as scientific or other valid words as misspelled words. This occurred because the specific words don't exist in dictionary. WebSpellChecker makes it possible for users to add such words into dictionary, so that WebSpellChecker will not mark it as misspelled word again in the future. To add word to a dictionary, simply right click on the specific word and choose "Add to dictionary" option in the context menu.

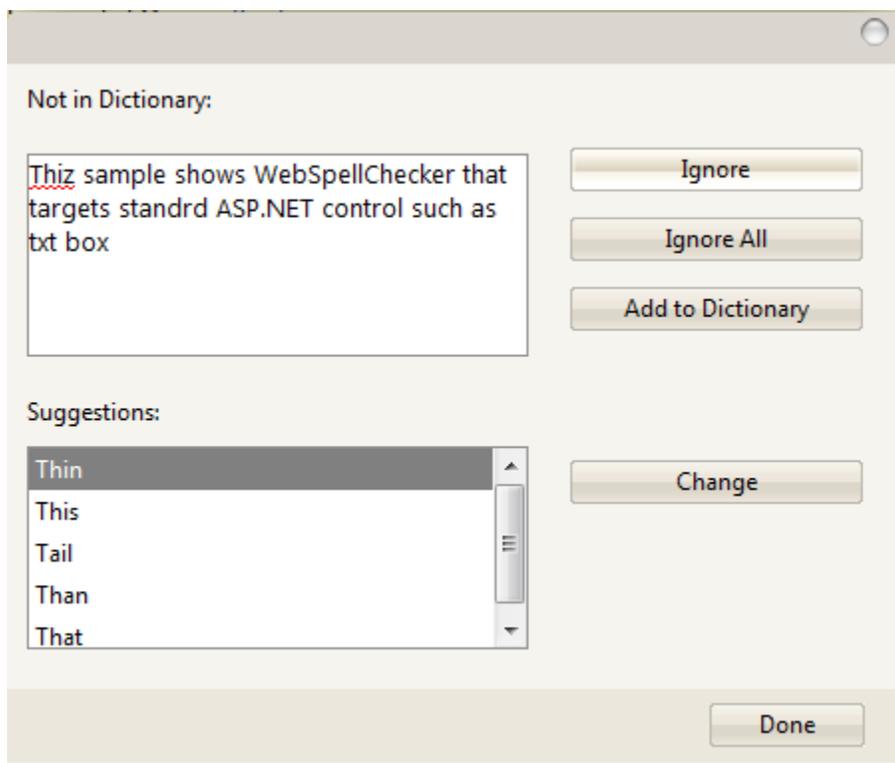
Note that in order to add to dictionary, the dictionary folder's permission should be granted with write access.



## Ignore

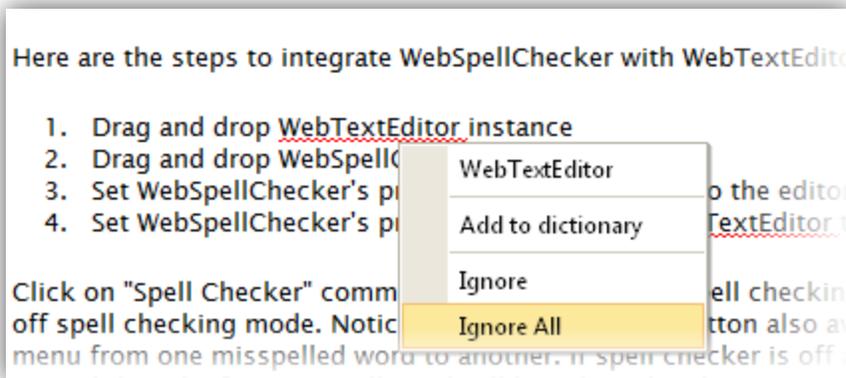
This command is used in a scenario where user would like WebSpellChecker to ignore a misspelled word so it will be marked as correct word instead of misspelled. If there are other similar words, WebSpellChecker will not mark other instances as correct word.

This command is available in both context menu and dialog box interface.



## Ignore All

Similar with Ignore command, “Ignore all” will mark the misspelled word to correct word. The only difference is “Ignore all” will mark all similar words as correct words, instead of just the selected word.



## Use WebSpellChecker with other controls

Integration between WebSpellChecker with other controls is made easy and simple. It is controlled with two properties, *TargetControlId* and *TargetControlIdValue*.

### TargetControlId

This property is required to determine the target control for spell check. For instance, *TargetControlId* should be filled with "TextBox1" value if you would like to spell check on TextBox1 control.

### TargetControlIdValue

This property determines the client-side property of the element from which WebSpellChecker should obtain the value for spell check. It's important to note that the value or content property should be accessible from client-side. For instance, if WebSpellChecker is targeting *TextBox* server side control, then the *TargetControlIdValue* property should be set to "value" instead of "Text".

Integration between WebSpellChecker and WebTextEditor is slightly different as you don't need to specify *TargetControlIdValue* property. Instead, please set the *IntegratedToWebTextEditor* property to true and *TargetControlId* property to the ID of WebTextEditor. For further information about this integration, please refer to Integration with WebSpellChecker topic in WebTextEditor User Interface whitepaper.

## Images settings

WebSpellChecker provides highly customizable settings for images, which can be done in *ImagesSettings* property.

Here is the list of customizable images in WebSpellChecker:

Category	Image Setting Key	Default Image Setting Value
<b>Misspell highlight line</b>	ErrorLine	red_underline.gif
<b>WebDialogBox inprogress image</b>	InProgress	inprogress.gif

## Text settings

Similar to other WebUI Studio® member, all textual settings in WebSpellChecker can be customized via *TextSettings* property.

The following is the list of customizable textual settings in WebSpellChecker:

Category	Text Setting Key	Default Text Setting Value
Context menu	AddToDictionaryText	Add to dictionary
	IgnoreAllText	Ignore all
	IgnoreText	Ignore

## Client side events

### OnInitialize(*controlId*)

Gets or sets the event handler when WebSpellChecker is initialized.

### OnBeforeCheckSpell(*controlId*)

Gets or sets the event handler before check spell is performed.

### OnAfterCheckSpell(*controlId*)

Gets or sets the event handler after check spell is performed.

### OnContextMenu(*controlId, menuObject, element*)

Gets or sets the event handler when WebSpellChecker's context menu is about to be displayed.

### OnClickContextMenu(*controlId, menuObj, action*)

Gets or sets the event handler when context menu item is clicked.

### OnExitSpellCheck(*controlId*)

Gets or sets the event handler when exit spell check.

### OnReplaceWord(*controlId, menuItem, originalWord, suggestionWords*)

Gets or sets the event handler when WebSpellChecker replace word.