

## An Introduction to ASPSpellCheck

ASPSpellCheck is the leading spell checker component for Microsoft® ASP 3 VBSCRIPT web applications.

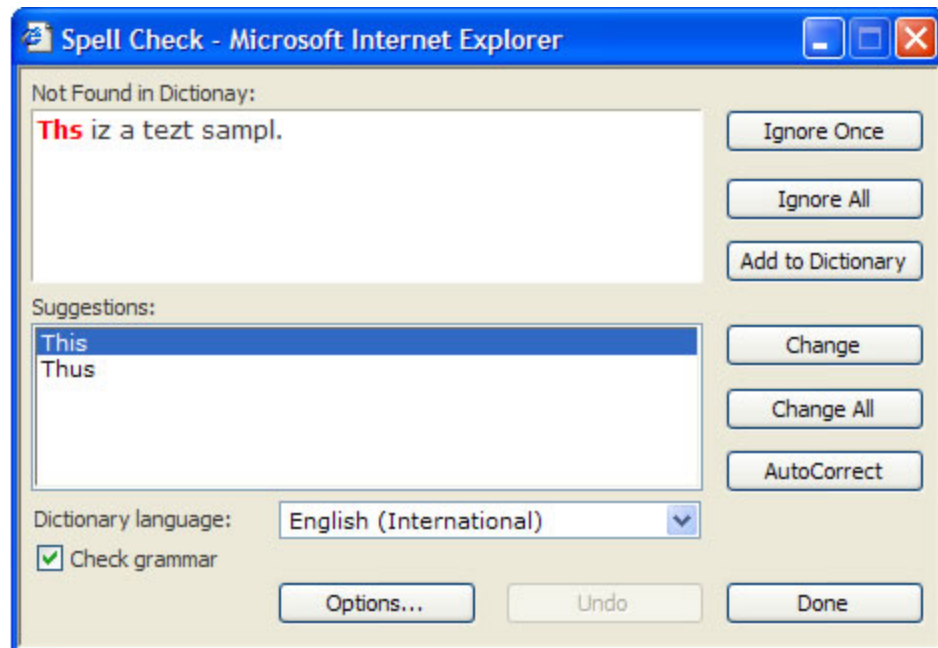
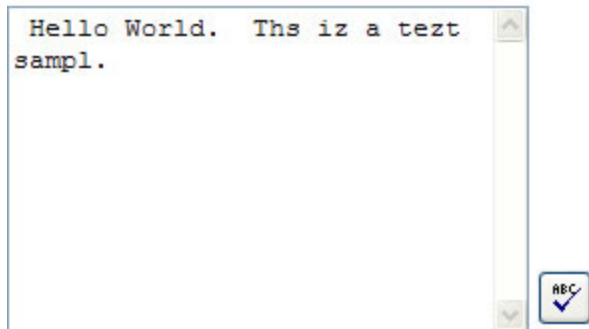
ASPSpellCheck is an advanced script based component; there are no Active-X controls or DLLs to install on the server. In addition no database is required. This means that it can be installed on almost all modern Microsoft® IIS web hosting environments - including Virtual/Shared Hosting.

ASPSpellCheck has dictionaries in many international languages, and an international user interface (8 Languages).

ASPSpellCheck can be used with both HTML and XHTML applications. It is W3C compliant and has a good degree of accessibility.

ASPSpellCheck's uses include:

- Spell Checking for forms on websites
- Spell Checking for content management systems
- Spell Checking for web-mail applications
- Spell Checking for browser based WYSIWYG html editors
- Spell Checking for intranets



## What's New to version 2.4?

### ASPSpellCheck's recent updates:

- Improved spell checker speed & suggestion quality.
- Improved support for WYSIWYG Editors.
- Improved XML and XHTML compatibility.
- Improved support for Scandinavian characters & accents.

# Getting Started with ASPSpellCheck

## Installation Basics

In brief, ASPSpellCheck is installed by copying the downloaded directory "ASPSpellCheck" to the root of your website.

## Usage Basics

You can add a spell-check link or button to almost any Form Field or HTML element within your application using the ASPSpellLink Class.

Its basic usage is shown below:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iz a tezt sampl.
</textarea>

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea" 'Sets the Field id(s) to be spell-checked
response.write myLink.imageButtonHTML("", "", "")
set myLink=nothing
%>
```

## Dictionary Basics

You can install additional language dictionaries such as French, Spanish, German, Dutch, Portuguese and many Specialized English dictionaries. The dictionaries have the file extension .dic.

To install these dictionaries - first download them from [www.aspspellcheck.com](http://www.aspspellcheck.com) and place them in the "Dictionaries" directory within your ASPSpellCheck Directory.

## The ASPSpellCheck Installation Process

ASPSpellCheck installation is simple. There are no DLLS, ActiveX controls or installers. This makes it very easy to use, even on most virtual hosting packages.

### Installing the component into your web application onto your web server

ASPSpellCheck is installed by copying the entire ASPSpellCheck directory (in the ZIP archive you downloaded) into a directory of your website. The ASPSpellCheck folder should be copied into the root level of your website wherever possible. If you do not work with IIS sites, then the root level of your website is likely to be **c:\inetpub\wwwroot**

Normally, the best position for this is in the root level of your website, as this makes for a clean and simple usage. If you decide to place the ASPSpellCheck component elsewhere, please make a note of this.

### Testing your ASPSpellCheck Installation

Using your web browser, view the script **sample.asp** within the ASPSpellCheck directory.  
For example: **<http://www.mysite.com/aspspellcheck/sample.asp>**

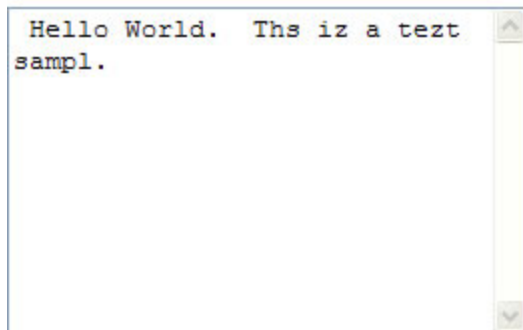
You should be able to see a simple HTML form. By clicking in the button beside it, you will open the spellchecker dialog. The Spellchecker will take a few seconds to set itself up the first time you use it.

### Testing your ASPSpellCheck Installation

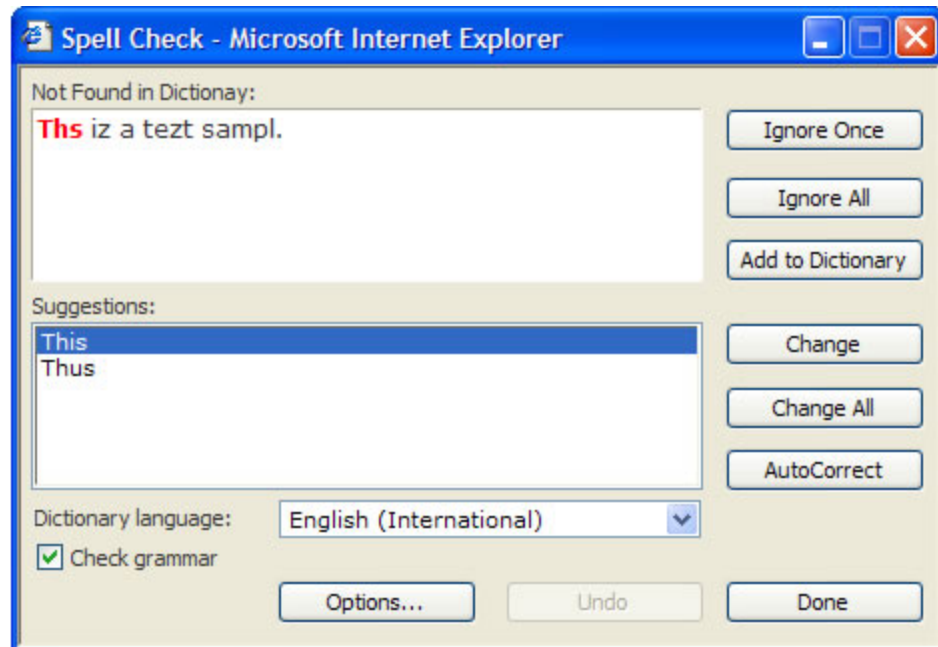
Using your web browser, view the script **sample.asp** within the ASPSpellCheck directory.  
For example: **<http://www.mysite.com/aspspellcheck/sample.asp>**

You should be able to see a simple HTML form.

By clicking in the button beside it, you will open the spellchecker dialog. The Spellchecker will take a few seconds to set itself up the first time you use it.



Hello World. Ths iz a text sampl.



**Note:**

If you installed the ASPSpellCheck to a directory other than **/ASPSpellCheck** then you will have to amend **sample.asp** to test the installation.

The code should read as below, where **myFolderPath** is the root level of your ASPSpellCheck directory **e.g.** `"/components/ASP/ASPSpellChecker/"`

Sample.asp:

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%option explicit%>

<!--#include file="ASPSpellInclude.inc"-->

<html>
<head>
<title>ASPSpellCheck Sample</title>
</head>

<body>

  <textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.  Ths iz a tezt sampl.
</textarea>

  <%
dim myLink
set myLink = new AspSpellLink

myLink.aspSpellPath="myFolderPath"

myLink.fields="MyTextArea"
response.write myLink.imageButtonXHTML("", "", "Spell-Check")
set myLink=nothing
%>

</body>
</html>
```

# Compatibility

## Browser Compatibility:

ASPSpellCheck adheres strictly to HTML, CSS and JavaScript standards, to ensure good ongoing browser compatibility for your spellchecker. We believe that ASPSpellCheck has amongst the highest compatibility of any commercial web spell checker.

ASPSpellCheck has been designed to work with modern versions of the following browsers:

- PC AOL Explorer
- PC Internet Explorer 4+ (including 7)
- PC Netscape Navigator 6+
- PC Mozilla FireFox
- PC Safari
- Macintosh Safari
- Macintosh Mozilla FireFox
- Macintosh Internet Explorer
- Macintosh Netscape Navigator
- Macintosh Camino

## Server Compatibility:

ASPSpellCheck has been designed to work with modern Microsoft® web servers. The minimum requirement is for NT4 Server - although Windows Server 2000 or newer is advised.

No special DLLS or EXEs or ACTIVEX controls need to be installed. In addition no database is required. This makes ASPSpellCheck ideal for many "Virtual Hosting" (shared hosting) environments.

To run the server must have IIS installed. The Application will need access to the following standard objects:

- The Application object
- Scripting.FileSystemObject
- Scripting.Dictionary
- Msxml2.ServerXMLHTTP

These environmental factors are available in the majority of modern Microsoft hosting environments, without any special installations being performed.

## Installing Dictionaries for ASPSpellCheck

You can install **additional language dictionaries** to ASPSpellCheck such as:

- English (Australia)
- English (Canada)
- English (International)
- English (UK)
- English (USA)
- French
- German
- Italian
- Dutch
- Portuguese
- Spanish
- Swedish
- Danish

You can download them from [www.aspspellcheck.com](http://www.aspspellcheck.com).

To install these dictionaries simply place them in the "ASPSpellCheck/Dictionaries" directory within your site.

### Custom Dictionaries

- You can add a [custom dictionary](#) of words specific to your application/website.
- In addition, each website user has their own personal dictionary (held as a cookie), to which they can add personal words.
- You can encode your own dictionary using the dictionary encoder utility ([www.aspspellcheck.com/downloads.asp](http://www.aspspellcheck.com/downloads.asp))

## Custom Dictionary

The Custom dictionary allows you to add a list of custom words to your spell checker's vocabulary. This is useful for adding words specific to a business, application or website.

The Spellchecker will look for updates to the custom dictionary every time a user opens the ASPSpellCheck dialog.

### Modifying the Custom Dictionary

The custom dictionary file is by default "**custom.txt**" within the ASPSpellCheck/Dictionaries Directory of your website. This can be changed in the [settings file](#).

To add words to your custom dictionary, simply place a list of up to 100,000 words, each on separate lines into this file. Each word should only consist of ANSI characters.

For significantly improved performance, please sort these words in ANSI code order. If you do not have a tool to do this, then [EditPlus](#) is an excellent choice. For performance issues, it is best to remove any spaces from the file.

### Example contents of custom.txt:

```
Aaron
Jacob
James
Julie
Steve
cybercom
intcom
marcoms
```

### Setting the custom dictionary from a database

You can also build your custom dictionary from an ASP script. The custom dictionary will have to be changed to in the [settings file](#) to specify a URL (e.g. "custom.asp").

Your ASP script should follow the same rules as per editing custom.txt by hand : up to 100,000 words, each on separate lines, no spaces, in ANSI code order.

### Sharing words that have been 'Added to Dictionary'

You can make the 'Add to Dictionary' function send words directly into this shared custom dictionary. To do so, you must set the variable **setting\_CentralDictionary** to **True** in the [settings file](#).

### Common Typographic Errors

The Spell Checker has an awareness of common typos. This is located in a file called "**CommonTypos.txt**" within the International directory of your installation. This file can be amended by adding / removing entries (per language) in the format:

```
typo--->suggestion
```

The entries must be put in in exact alphabetical ASCII character code order.



## The Settings File

The settings file can be found in the ASPSpellCheck directory. It is called **settings.inc**.

The settings file can be used to change the default settings for your spell checker. This can save a lot of coding later, using the [ASPSpellLink](#) Class. In addition, some global settings can be changed in this file which cannot be set in your own scripts.

**The settings file will look something like this:**

```
<%
setting_Session = true
setting_MultiLang = false
setting_DictPath = "Dictionaries/"
setting_CustomDict = "custom.txt"
setting_CSS=""
setting_UILang = "EN"
setting_Langs = "English (International)"
setting_Grammar = True
setting_CaseSensitive = True
setting_AllCaps = True
setting_Web = True
setting_Numeric = True
setting_NewLine = True
setting_NewLine = True
setting_CentralDictionary = False
'settings_BannedWords=array("profanity","profanity")
'settings_customTextArray
%>
```

### Modifying the settings file:

You are advised to be careful whilst editing this file, as errors can cause the ASPSpellCheck dialog not to run.

**setting\_Session** sets the default use of session acceleration for the ASPSpellCheck Dialog. Possible values are true or false.

This is analogous to the [useSession](#) property of the ASPSpellLink class.

**setting\_MultiLang** forces the ASPSpellCheck Dialog to operate in Multilingual mode if set to true. Possible values are true or false.

This is analogous to the [setMultiLingual](#) method of the ASPSpellLink class.

**setting\_DictPath** sets the path where the application looks for [installed dictionaries](#). This path is relative to the ASPSpellCheck Directory.

**setting\_CustomDict** sets the name of the Custom Dictionary file within the **setting\_DictPath**. This is normally a text file, but can also be an ASP script to retrieve words from a dictionary.

**setting\_CSS** sets the path of an external CSS used to style the Dialog. Leaving the value as an empty string uses the default (embedded) style sheet. This is analogous to the [externalCSS](#) property of the ASPSpellLink class.

**setting\_UILang** sets the default user interface language for the spell checking dialog. This is analogous to the [dialogLanguage](#) property of the ASPSpellLink class.

Possible values are:

- "EN" - For English
- "ES" - For Spanish
- "DE" - For German

- "NL" - For Dutch
- "FR" - For French
- "IT" - For Italian
- "PT" - For Portuguese
- "NO" - For Norwegian
- "SV" - For Swedish
- "DK" - For Danish
- "CUSTOM" - To set the dialog texts yourself. This requires **settings\_customTextArray** to be used (see below).

**setting\_Langs** sets the default dictionary language for the spell checking dialog. This relates to the [dictionary](#) property of the ASPSpellLink class.

You may stipulate more than one language in a comma separated list.

Possible values are:

- "Deutsch" - for German
- "English (Australia)" - for Australian English
- "English (Canada)" - for Canadian English
- "English (International)" - for International English
- "English (UK)" - for British English
- "English (USA)" - for USA American English
- "Espagnol" - for Spanish
- "Francais" - for French
- "Italiano" - for Italian
- "Nederlands" - for Dutch
- "Portugues" - for Portuguese
- "Svenska" - for Swedish
- "Dansk" - for Danish

**setting\_Grammar** sets whether the ASPSpellCheck dialog will check grammar by default. Possible values are true or false.

This is analogous to the [checkGrammar](#) property of the ASPSpellLink class.

**setting\_CaseSensitive** sets whether the ASPSpellCheck dialog will be case sensitive by default. Possible values are true or false.

This is analogous to the [caseSensitive](#) property of the ASPSpellLink class.

**setting\_AllCaps** sets whether the ASPSpellCheck dialog will ignore words in all capital letters (such as acronyms) by default. Possible values are true or false.

This is analogous to the [ignoreAllCaps](#) property of the ASPSpellLink class.

**setting\_Web** sets whether the ASPSpellCheck dialog will ignore domains, email addresses and URLs by default. Possible values are true or false.

This is analogous to the [ignoreWebAddresses](#) property of the ASPSpellLink class.

**setting\_Numeric** sets whether the ASPSpellCheck dialog will ignore words with numbers in them by default. Possible values are true or false.

This is analogous to the [ignoreNumbers](#) property of the ASPSpellLink class.

**setting\_NewLine** sets whether the ASPSpellCheck dialog will expect a new sentence after each carriage return (new line) by default. Possible values are true or false.

This is analogous to the [newSentenceOnEachNewLine](#) property of the ASPSpellLink class.

**setting\_CentralDictionary** - if set True, words 'added' to the dictionary are written into the custom dictionary - "custom.txt" . All users will then have access to these words.

This is useful in intranet applications, but less so for public websites. If you wish to use the custom dictionary, you must assign IIS (IUSR\_MACHINENAME) write permissions to the /ASPSpellCheck/Dictionaries directory.

**settings\_BannedWords** allows you disalow words from being used or suggested. This is useful to block profane words.

To do so, replace this line:

'settings\_BannedWords

with the following code:

---

```
settings_BannedWords = array("rude","words","in","an","array")
```

---

**settings\_CustomTextArray** allows you change the text used on the ASPSpellCheck Dialog.

To do so, replace this line:

'settings\_CustomTextArray=array("profanity","profanity")

with the following code:

---

```
settings_CustomTextArray = Array("Spell Check", "Cancel", "OK", "Reset AutoCorrect", "Reset  
Personal Dictionary", "Dictionaries:", "Case Sensitive.", "Ignore ALL CAPS.", "Ignore Email  
and Internet Addresses.", "Ignore Words Containing Numbers.", "New Sentence for each  
New-Line.", "Are you sure you wish to clear your personal dictionary?", "Are you sure you  
wish to clear your personal Auto Corrections?", "Not Found in Dictionary:", "Grammatical  
Error:", "Repeated Word:", "Suggestions:", "Dictionary language:", "Change", "Done", "Ignore  
Once", "Ignore All", "Add to Dictionary", "Change All", "AutoCorrect", "Undo", "Options...",  
"Undo Edit", "Check grammar", "( No suggestions found )", "( Delete repeated word )",  
"Spell-check Summary:", "Words:", "Documents:", "Edits:")
```

---

You can change any of the pieces of text within the array. Be sure to close all of your string properly.

**N.B.** To use this custom text, you must either set setting\_UILang="CUSTOM" in the settings file, or set the ASPSpellLink.dialogLanguage to "CUSTOM" in your scripts.

## Usage Basics

The most fundamental step to using the ASPSpellLink Class is to include the file:  
**/ASPSpellCheck/ASPSpellInclude.inc** to your ASP File.

**I.E:**

```
<!--#include file= "ASPSpellCheck/ASPSpellInclude.inc"-->
```

This uses a relative path to the file.

**Or**

```
<!--#include virtual= "/ASPSpellCheck/ASPSpellInclude.inc"-->
```

This uses an absolute path to the file, relative to the root directory of your website.

If you are unfamiliar with ASP includes, there is a good reference online at [w3schools](http://w3schools.com).

## The ASPSpellLink Class

The ASPSpellLink Class allows you to add a link or button to an ASP page which will initiate the spell checker. It also allows you to modify the spell checker - for example: setting the dialog language to French, or choosing a German dictionary.

Using this class, you can spell-check almost any HTML form field, or HTML element. You can also spell-check multiple fields at once.

**The file /ASPSpellCheck/ASPSpellInclude.inc must be included into your ASP script to use the ASPSpellLink Class.**

### Using ASPSpellLink

The code below shows an example of opening and closing the ASPSpellLink Class .

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World
</textarea>

<%
dim myLink
set myLink = new AspSpellLink 'opens the class
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked
response.write myLink.imageButtonHTML("", "", "") 'renders a button
set myLink= nothing 'closes the class

%>
```

## ASPSpellLink.caseSensitive

This Boolean property is used to decide if the spell-check dialog will act in a case sensitive manner. The default value is defined in the [settings file](#).

Also see the [checkGrammar](#) property.

If checkGrammar is set to true then the spellchecker can pickup irregularities in sentences even if caseSensitive is set to false.

### Example

If you wanted to your spellchecker to ignore case mistakes in words:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.
</textarea>

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked
myLink.caseSensitive=false 'not case sensitive

response.write myLink.imageButtonHTML ("","") 'renders a button
set myLink = nothing 'closes the class
%>
```

## ASPSpellLink.checkGrammar

This Boolean property is used to decide if the spell-check dialog will find basic grammar mistakes such as repeated words, poor sentence casing etc.

The default value is defined in the [settings file](#).

### Example

If you wanted to your spellchecker to ignore grammatical mistakes :

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.
</textarea>

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked
myLink.checkGrammar= false 'grammar check turned off

response.write myLink.imageButtonHTML("", "", "") 'renders a button
set myLink = nothing 'closes the class
%>
```

## ASPSpellLink.dictionary

The dictionary property sets the default dictionary which spellchecker will use.

The value should be the same as the name of any dictionary you have [installed](#) in the Dictionaries directory (do not use the ".dic" at the end though).

### Examples of possible values of dictionary are:

- "Deutsch" - for German
- "English (Australia)" - for Australian English
- "English (Canada)" - for Canadian English
- "English (International)" - for International English
- "English (UK)" - for British English
- "English (USA)" - for US American English
- "Espanol" - for Spanish
- "Francais" - for French
- "Italiano" - for Italian
- "Nederlands" - for Dutch
- "Portugues" - for Portuguese
- "Svenska" - for Swedish
- "Dansk" - For Danish

You may use up to multiple dictionaries at the same time using a comma separated list (see below).

If this method is not used, the Dictionary used will be defined by the [settings file](#).

### Examples

#### If you wanted to your spellchecker to use a German Dictionary:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Guttten Tag.
</textarea >

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked
myLink.dictionary = "Deutsch" 'German Dictionary
response.write myLink.imageButtonHTML("", "", "") 'renders a button
set myLink = nothing 'closes the class
%>
```

#### If you wanted to your spellchecker to use French and English together :

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->
```



```
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Bonjour
</textarea >
```

```
<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked
myLink.dictionary = "English (Canada), Francais" 'English and French Dictionaries used
together
response.write myLink.imageButtonHTML("", "", "") 'renders a button
set myLink = nothing 'closes the class
%>
```

**If you wanted to your spellchecker to use a French Dictionary and use a [French language spellchecker dialog](#):**

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->
```

```
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.  Bonjour.
</textarea >
```

```
<%
dim myLink
set myLink = new AspSpellLink
myLink.fields= "MyTextArea"
myLink.dictionary = "Francais" 'French Dictionary
myLink.dialogLanguage= "FR" 'French Language Dialog
response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing
%>
```

## ASPSpellLink.dialogLanguage

This property is used to decide the language of the spell checking dialog.  
The default value is defined in the [settings file](#).

The possible values of ASPSpellLink.dialogLanguage are:

- "EN" - For English
- "ES" - For Spanish
- "DE" - For German
- "NL" - For Dutch
- "FR" - For French
- "IT" - For Italian
- "PT" - For Portuguese
- "NO" - For Norwegian
- "SV" - For Swedish
- "DK" - For Danish
- "CUSTOM" - To set the dialog texts yourself. This requires modifying the settings\_customTextArray property in the settings file.

Note that changing the Dialog language does not automatically select the Dictionary Language used for spellchecking.  
To choose a dictionary language(s), use the [dictionary](#) property.

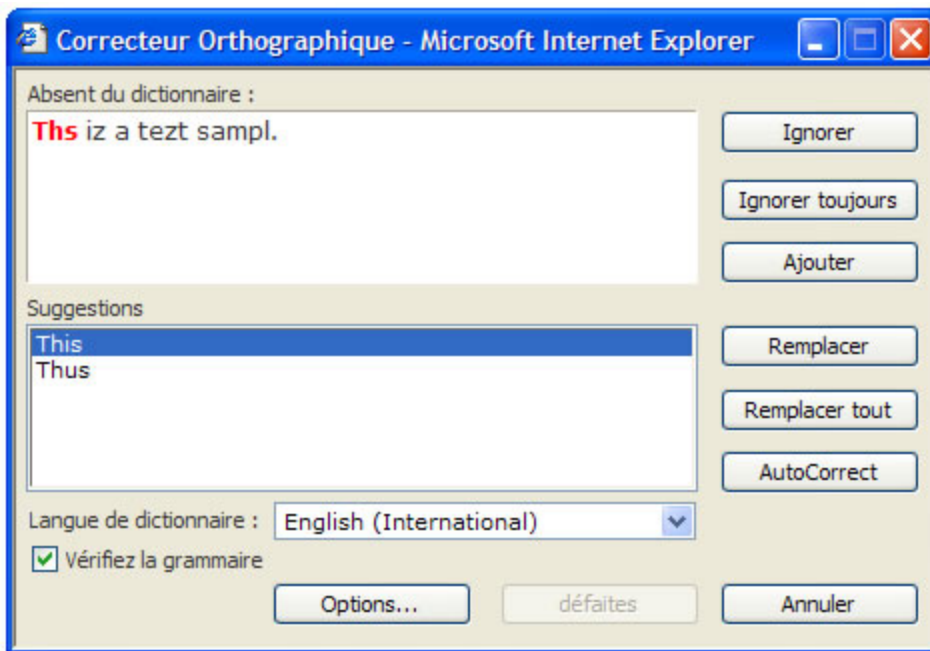
### Example

#### If you wanted to your spellchecker to display in French:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iza tezt sampl.
</textarea>

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"
myLink.dialogLanguage= "FR"
response.write myLink.imageButtonHTML("", "", "") 'renders a button
set myLink = nothing 'closes the class
%>
```



## ASPSpellLink.doneMessage

The doneMessage property is used to decide give an optional Javascript "Alert" at the end of spell checking. The doneMessage property is of type string.

You can customise this message as desired.

### Example

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.  Ths iz a tezt sampl.
</textarea>

<%
dim myLink
set myLink = new AspSpellLink
myLink.doneMessage="Spell Checking Completed!"
myLink.fields="MyTextArea"
response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing
%>
```

## ASPSpellLink.externalCSS

This property is to attach an optional CSS stylesheet to the spellchecking Dialog.  
If this property is not set, the default CSS will be resemble Windows® XP.

There is a sample stylesheet included in the Assets folder within ASPSpellCheck. To use this effectively, it is advisable to study the output HTML from the dialog.

### Example

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.
</textarea >

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea"
myLink.externalCSS="Assets/sample.css" 'Dialog Uses External CSS
response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing
%>
```

## ASPSpellLink.fields

ASPSpellLink.fields defines the Form fields or HTML elements to be spellchecked. ASPSpellLink.fields is a string.

This is possibly the most important property of ASPSpellLink. Fields has no default value, and must be set for the spell-check to work.

Fields may identify one field or HTML element by its ID property.

You may also identify more than one field or HTML element to spell-check using a comma separated list.

### Example

#### To spell-check the textArea with Id: "MyTextArea":

```
<!--#include virtual= "/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.
</textarea >

<%
dim myLink
set myLink = new AspSpellLink

myLink.fields="MyTextArea"
' sets an object ID to be spellchecked

response.write myLink.imageButtonHTML("", "", "") 'renders a button
set myLink = nothing
%>
```

### Spell-Checking more than one Field

If there were more than one field to spell-check:

```
myLink.fields="MyTextArea,MyTextInput,MyHTMLElement"
```

Where MyTextArea, MyTextInput and MyHTMLElement are example ID values of elements on your webpage. Note that field ID names are always Case Sensitive.

### Spell-Checking Iframes

ASPSpellCheck can also spell-check the entire contents of an Iframe. This is useful for integration with WYSIWYG HTML Editors. To spell-check an Iframe prefix its **id** with the string **"iframe:"**

e.g.

```
myLink.fields="iframe:myIFrameId"
```

To spell check in iframe within another iframe (as is common for WYSIWYG Editors) use dot notation to identify its full path.

e.g.

```
myLink.fields="iframe:myMainFrame.mySubFrame"
```

## ASPSpellLink.hideSummary

This Boolean property is used to decide if the spell-check dialog will hide the summary screen.

This is particularly useful in form validation if:

- ASPSpellLink.hideSummary is set **True**
- the JavaScript Callback API is used to submit a form after spellchecking.

This allows you to use the spellchecking button a form submission button - however the form will only be submitted after the spellchecking is complete.

The default value of hideSummary is false.

### Example

If you wanted to have discreet Spelling validation on a form:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<form action="" method="POST" id="form1" >
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
With no spelling errors - the spell check validation is quite discreet.
</textarea>
</form>

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea"
myLink.hideSummary= true
response.write myLink.ButtonHTML("Spellcheck and Submit","")
set myLink = nothing 'closes the class
%>

<script language="javascript">
function ASPSpellDone(strFields){
document.getElementById('form1').submit()
}
</script>
```

## ASPSpellLink.ignoreAllCaps

This Boolean property is used to decide if the spell-check dialog will ignore words which are in ALL CAPITALS, such as acronyms.

The default value is defined in the [settings file](#).

### Example

If you wanted to your spellchecker to ignore ALL CAPITAL words:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
HELLO World.
</textarea >

<%
dim myLink
set myLink = new ASPSpellLink
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked

myLink.ignoreAllCaps= true 'ALL CAPS Ignored

response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing
%>
```



## ASPSpellLink.ignoreNumbers

This Boolean property is used to decide if the spell-check dialog will ignore words containing numbers such as "\$100mill" or "8Ball".

The default value is defined in the [settings file](#).

### Example

If you wanted to your spellchecker to ignore words with numbers in them:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello 123World.
</textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"

myLink.ignoreNumbers= true
'Numeric Words Ignored

response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing
%>
```

## ASPSpellLink.ignoreWebAddresses

This Boolean property is used to decide if the spell check dialog will ignore words which resemble web addresses such as "**http://www.aspspellcheck.com**" or "**Ebay.com**" and email addresses.

The default value is defined in the [settings file](#).

### Sub Header

If you wanted to your spellchecker to ignore web addresses and email addresses:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. http://www.aspnetspell.com
</textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"

myLink.ignoreWebAddresses= true
'URLS and Emails Ignored

response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing
%>
```

## ASPSpellLink.javaScriptCallback

When the spell checking is complete - a JavaScript function can be called in your web-page. The name of this function is set by javaScriptCallback .

This is ideal for [submitting forms](#) after spellchecking.

You may use a different function for each spell-check button.

### Example:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iz a tezt sampl.
</textarea >

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea"

myLink.javaScriptCallback = "allDone"

response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing
%>

<script language= "javascript">
function allDone (){
alert ('Spellchecking Completed')
}
</script>
```

## ASPSpellLink.newSentenceOnEachNewLine

This Boolean property is used to decide if the spell-check dialog will expect a new sentence after each carriage return (new line).

This is useful to ensure proper grammar as per a word-processor, but may not be appropriate for more 'casual' web forms.

The default value is defined in the [settings file](#).

### Example

If you wanted to your spellchecker to not expect a new sentence on each new line of the document:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id= "MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.
</textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked

myLink.newSentenceOnEachNewLine = false
'less formal grammar

response.write myLink.imageButtonHTML("", "", "") 'renders a button
set myLink = nothing 'closes the class
%>
```

## ASPSpellLink.useSession

This Boolean property is used to decide if the spell-check dialog will use the IIS Session object to accelerate the dialog for each user.

Using the session will reduce the bandwidth used by the ASPSpellCheck dialog, but will use more server memory.

The default value is defined in the [settings file](#).

### Example

If you wanted to your spellchecker to not use the session to accelerate the ASPSpellCheck dialog:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id= "MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths izateztsampl.
</textarea >

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea"

myLink.useSession= false 'no session acceleration

response.write myLink.imageButtonHTML("", "", "")
set myLink = nothing 'closes the class
%>
```

### Note:

Session keys used by the ASPSpellCheck dialog are always prefixed with "ASPSPELLCHECK.FORMVARIABLES."

## ASPSpellLink.ASPSpellPath

This property is used to set the full path to the ASPSpellCheck directory within your site. It should be used as an absolute path with respect to the root of your website.

ASPSpellLink.ASPSpellPath is a string.

The default value of ASPSpellLink.ASPSpellPath is `"/ASPSpellCheck/"`.

This property need only be set if this is not the full path to the folder in which you installed ASPSpellCheck.

### Example

If you installed ASPSpellCheck in the directory `"/Components/ASP/ASPSpellCheck/"` then an example usage might be:

### Sub Header

If you installed ASPSpellCheck in the directory `"/Components/ASP/ASPSpellCheck/"` then an example usage might be:

```
<!--#include virtual= "/Components/ASP/ASPSpellCheck/ASPSpellInclude.inc"-->
```

```
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >  
Hello World.  
</textarea>
```

```
<%
```

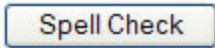
```
dim myLink  
set myLink = new AspSpellLink  
myLink.fields="MyTextArea" ' sets an object ID to be spellchecked  
myLink.ASPSpellPath="/Components/ASP/ASPSpellCheck/"  
  
response.write myLink.imageButtonHTML("", "", "") 'renders a button  
set myLink = nothing 'closes the class
```

```
%>
```

## ASPSpellLink.buttonHTML(buttonText,CSSclass)

The method buttonHTML returns a W3C compliant HTML string which can be used to add an HTML button to your ASP page. When this button is clicked, the ASPSpellCheck dialog will be opened.

### Example button:



The parameter buttonText represents the ALT text for the button. If set to an empty string (e.g. "") the default value will be "Spell Check"

The parameter CSSclass allows you to nominate a CSS class to be applied to the button. If set to an empty string "" then no CSS class will be applied.

### Example:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

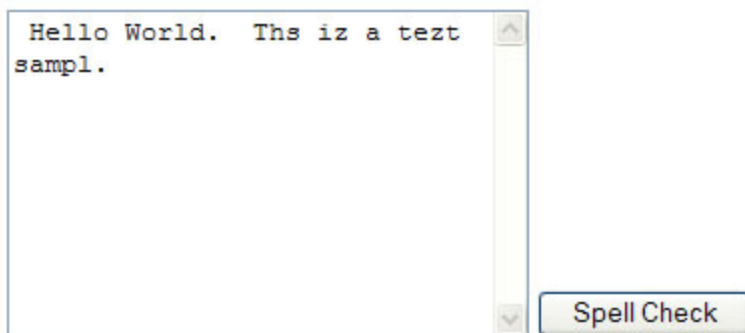
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >Hello World. Ths
izateztsampl. </textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"

response.write myLink.ButtonHTML("Spell Check","myButtonClass") ' renders an HTML button

set myLink= nothing
%>
```

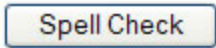
### Result:



## ASPSpellLink.buttonXHTML(buttonText,CSSclass)

The method buttonXHTML returns a W3C compliant XHTML string which can be used to add an XHTML button to your ASP page. When this button is clicked, the ASPSpellCheck dialog will be opened.

### Example button:



The parameter buttonText represents the ALT text for the button. If set empty string (e.g. "") the default value will be "Spell Check"

The parameter CSSclass allows you to nominate a CSS class to be applied to the button. If set to an empty string ("") then no CSS class will be applied.

### Example:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

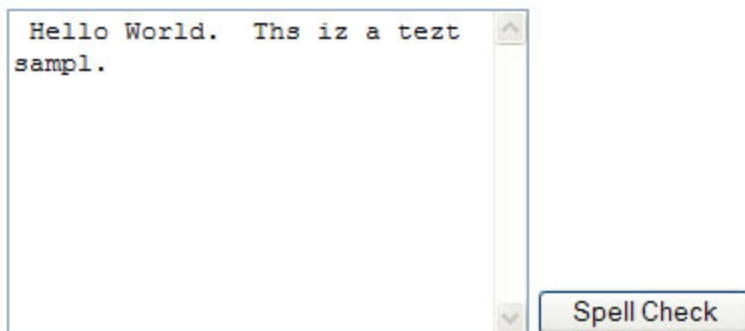
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >Hello World. Ths
izateztsampl. </textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"

response.write myLink.ButtonXHTML("Spell Check","myButtonClass")
' renders an XHTML button

set myLink= nothing
%>
```

### Result:





## ASPSpellLink.imageButtonHTML(buttonImage,rolloverImage, buttonText)

The method `imageButtonHTML` returns a W3C compliant HTML string which can be used to add a visually pleasing button (with a roll-over state) to your ASP page. When this button is clicked, the `ASPSpellCheck` dialog will be opened.

### Example image button:



The `buttonImage` parameter sets the path to an image you wish to use for the button. If this is set to an empty string (e.g. `""`) then the button above is rendered.

The `rolloverImage` parameter sets the path to an image you wish to use for the button. If this is set to an empty string (e.g. `""`) then there will be no customized rollover state.

The `buttonText` parameter represents the ALT text for the button. If set to an empty string (e.g. `""`) then the default value will be "Spell Check"

### Example:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

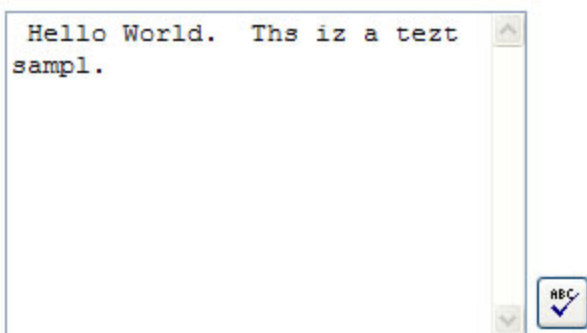
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iz a tezt sampl.
</textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"

response.write myLink.imageButtonHTML("", "", "")

set myLink=nothing
%>
```

### Result:



## ASPSpellLink.imageButtonXHTML(buttonImage,rolloverImage, buttonText)

The method `imageButtonXHTML` returns a W3C compliant XHTML string which can be used to add a visually pleasing button (with a roll-over state) to your ASP page. When this button is clicked, the `ASPSpellCheck` dialog will be opened.

### Example image button:



The `buttonImage` parameter sets the path to an image you wish to use for the button. If this is set to an empty string (e.g. `""`) then the button above is rendered.

The `rolloverImage` parameter sets the path to an image you wish to use for the button. If this is set to an empty string (e.g. `""`) then there will be no customized rollover state.

The `buttonText` parameter represents the ALT text for the button. If set to an empty string (e.g. `""`) the default value will be "Spell Check"

### Example:

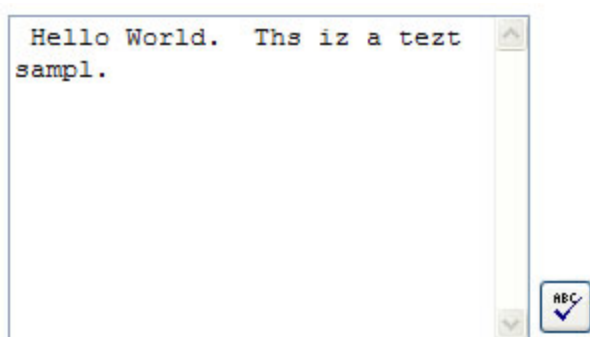
```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->
<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iz a tezt sampl.
</textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"

response.write myLink.imageButtonXHTML("", "", "")

set myLink=nothing
%>
```

### Result:



## ASPSpellLink.linkHTML(linkText,CSSclass)

The method linkHTML returns a W3C compliant HTML/XHTML string which can be used to add a spell-checking hyperlink to your ASP web pages.

### Example link:

[SpellCheck](#)

The linkText parameter represents the text of the link. If set to an empty string (e.g. "") then the default value will be "Spell Check"

The parameter CSSclass allows you to nominate a CSS class to be applied to the link (anchor tag). If set to an empty string (e.g. "") , no CSS class will be applied.

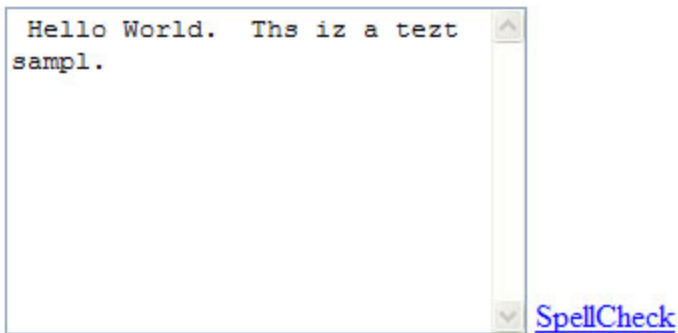
### Example:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iz a tezt sampl.
</textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"
response.write myLink.linkHTML("SpellCheck","")
set myLink=nothing
%>
```

### Result:



Hello World. Ths iz a tezt  
sampl.

[SpellCheck](#)

## ASPSpellLink.spellingCheck(strWord)

The method spellingCheck spell checks a string in ASP. It returns either true or false.

### Example:

---

```
<%  
dim myLink, myResult  
set myLink = new ASPSpellLink  
myResult = myLink.spellingCheck("Hello World")  
set myLink=nothing  
%>
```

---

**Outcome:** myResult is true.

[Learn more](#)

## ASPSpellLink.spellingSuggestions(strWord)

The method spellingSuggestions returns spelling suggestions for a string in ASP. It returns an array of suggestions.

### Example:

```
<%  
dim myLink, myResult  
set myLink = new ASPSpellLink  
  
myResult = myLink.spellingSuggestions("misspellt")  
  
set myLink=nothing  
response.write join(myResult, ",") ' write the outcome to the screen  
%>
```

**Outcome:** misspell,misspelled,misplay,misspells,misplace

[Learn more](#)

## ASPSpellLink.url()

The method returns a url which will call open the spellchecker the the settings as set by your choices in the ASPSpellLink class.

This is useful for custom spellchecking applications such as [run-time dictionary selection](#).

Note: when opening a dialog window using this url we suggest minimum dimensions of 460x290 pixels, with no scrollbars.

### Example:

```
<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<textarea id= "MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World.
</textarea >

<%
dim myLink
set myLink = new AspNetSpellLink
myLink.fields="MyTextArea"

response.write myLink.url()

set myLink=nothing
%>
```

**Outcome:** /ASPSpellCheck/ASPSpellCheck.asp?fields=MyTextArea

## Notes for Advanced Developers

Source code is provided for the ASPSpellLink Class. (ASPSpellCheck/ASPSpellInclude.inc)

This Class is written in VBScript, in an uncomplicated OO style. For reference, here is a good [tutorial on OO Classes in VBSCRIPT](#).

## ASP.Net Spell Check Compatibility

**Before reading this article - we suggest that you review our sister project - ASPNetSpell**  
**[www.aspnetspell.com](http://www.aspnetspell.com)**

ASPSpellCheck is easily compatible with ASP.NET

First, install copy ASPSpellCheck by copying it to the root of your website.  
The following code will add spell checking capabilities to a button:

### Source Code in VB.NET

```
Fields = "Textbox1"

Button1.Attributes.Add("onclick", "window.open ('\\ASPSpellCheck/ASPSpellCheck.asp?fields=" +
Fields + "', 'aspSpellWin', 'width=460, height=290, scrollbars=no')")
```

**Button1** is name of the button which will initiate spell checking.

**Fields** is the name of a text box to be spell checked, if you wish to spellcheck more than one - use a comma separated list.

You can use the [settings file](#) to customize the spell checker in detail.

### Advanced Usage

You can also set any property of the spellchecker by adding that property to the URL query string.

**e.g. &dictionary=Francais&ignoreAllCaps=true**



## Notes on Server Caching

To increase performance, ASPSpellCheck caches data at the ISS Application level. This hugely saves on server processor resources, and takes only a few MB of memory per dictionary.

For this reason, the first time you use a dictionary, or after IIS is restated, the spell checker will take a few seconds to set its self up.

The data is stored in the Application using Keys starting with "**ASP\_SPELL**". You are advised to not try to read or modify these application variables

## Basic SpellCheck

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%option explicit%>

<!--#include virtual= "/ASPSpellCheck/ASPSpellInclude.inc"-->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
http://www.w3.org/TR/html4/loose.dtd>

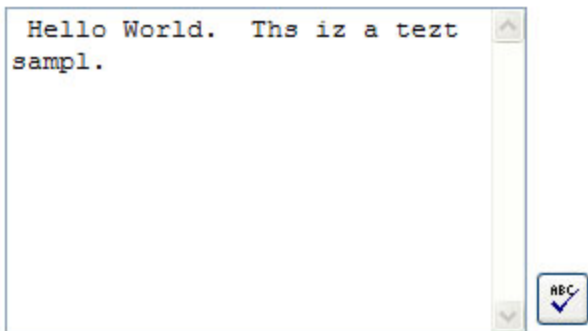
<html>
<head>
<title>ASPSpellCheck Sample</title>
</head>
<body>

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
    Hello World.
</textarea>

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea"
response.write myLink.imageButtonHTML ("","","")
set myLink=nothing
%>

</body>
</html>
```

### Outcome:



### Notes:

Use buttonHTML, buttonXHTML, imageButtonXHTML or linkHTML methods instead of imageButtonHTML for different types of spell-check link.

# Server Side Spellchecking

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%option explicit%>

<!--#include virtual= "/ASPSpellCheck/ASPSpellInclude.inc -->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
http://www.w3.org/TR/html4/loose.dtd>
<html>
<head>
<title>ASPSpellCheck Sample</title>
</head>
<body>

<form id="form1" name="form1" action="" method="post">
<input name="strWord" id="strWord" value="<%=request.form("strWord")%>">
<input type="submit">

<%

dim myLink, myWord, binGood, arrSuggestions, strSuggestions
if request.form("strWord")<>"" then
myWord = request.form("strWord")
%>

<hr><h2><%=myWord%></h2><br>

<%
set myLink = new AspSpellLink
binGood = myLink.spellingCheck ( myWord )
%>
Spelling: <%=binGood%>

<%
if not binGood then
arrSuggestions = myLink.spellingSuggestions ( myWord )
strSuggestions = join(arrSuggestions," ", " ")
%>

Suggestions: <%=strSuggestions%>

<%
end if
set myLink=nothing
end if
%>

</form>
</body>
</html>
```

# Language Variants

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%option explicit%>

<!--#include virtual= "/ASPSpellCheck/ASPSpellInclude.inc"-->

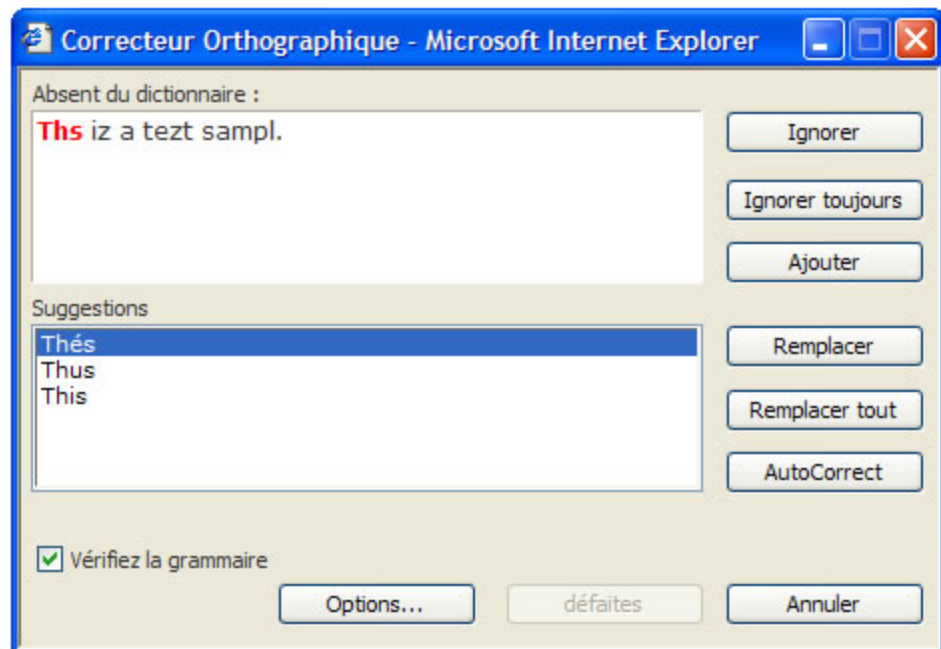
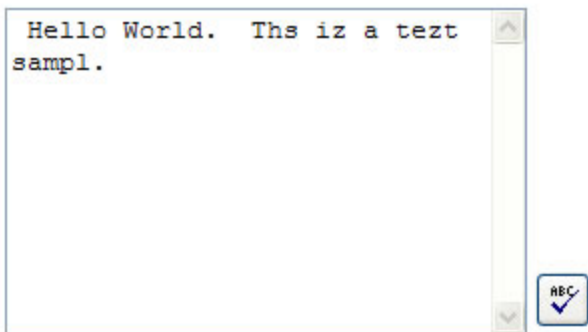
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
http://www.w3.org/TR/html4/loose.dtd>
<html>
<head>
<title>ASPSpellCheck Sample</title>
</head>
<body>

<textarea id="MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iz a tezt sampl.
</textarea>

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea"
myLink.dictionary = "Francais, English (Canada)"
myLink.dialogLanguage= "FR" 'French Language Dialog
response.write myLink.imageButtonHTML ("","","")
set myLink=nothing
%>

</body>
</html>
```

## Outcome:



French Dialog. Spellchecking in Canadian English and French simultaneously.

## Notes:

You must [install the appropriate dictionaries](#) (such as "**Francais**" and "**English (Canada)**") before they can be used.

Learn more about [setting the dialog language](#).

Learn more about [choosing the Dictionary Languages you wish to spell-check with](#).

## Submitting a Form after Spellchecking

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%option explicit%>

<!--#include virtual= "/ASPSpellCheck/ASPSpellInclude.inc"-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
http://www.w3.org/TR/html4/loose.dtd>
<html>
<head>
<title>ASPSpellCheck Sample</title>
</head>
<body>
<form id="form1" name="form1" action="" method="post">

<textarea id= "MyTextArea" name="MyTextArea" cols="30" rows="10" >
Hello World. Ths iz a tezt sampl.
</textarea>

<%
dim myLink
set myLink = new AspSpellLink
myLink.fields="MyTextArea"
myLink.javascriptCallback = "allDone"
response.write myLink.buttonHTML ("Spellcheck then submit","")
set myLink=nothing
%>

<script language= "javascript">
function allDone(strFields){
alert("submitting the form")
document.form1.submit()
}
</script>

</form>
</body>
</html>
```

## WYSIWYG Usage

Please see the full tutorial on Rich Editor Integration and try the FCKEditor Spellchecker Plug-in:  
<http://www.aspspellcheck.com/demo-wysiwyg-html-editor-spellchecking.asp>

```
<%@LANGUAGE="VBSCRIPT" CODEPAGE="1252"%>
<%option explicit%>

<!--#include virtual="/ASPSpellCheck/ASPSpellInclude.inc"-->

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
http://www.w3.org/TR/html4/loose.dtd>
<html>
<head>
<title>ASPSpellCheck Sample</title>
</head>
<body>

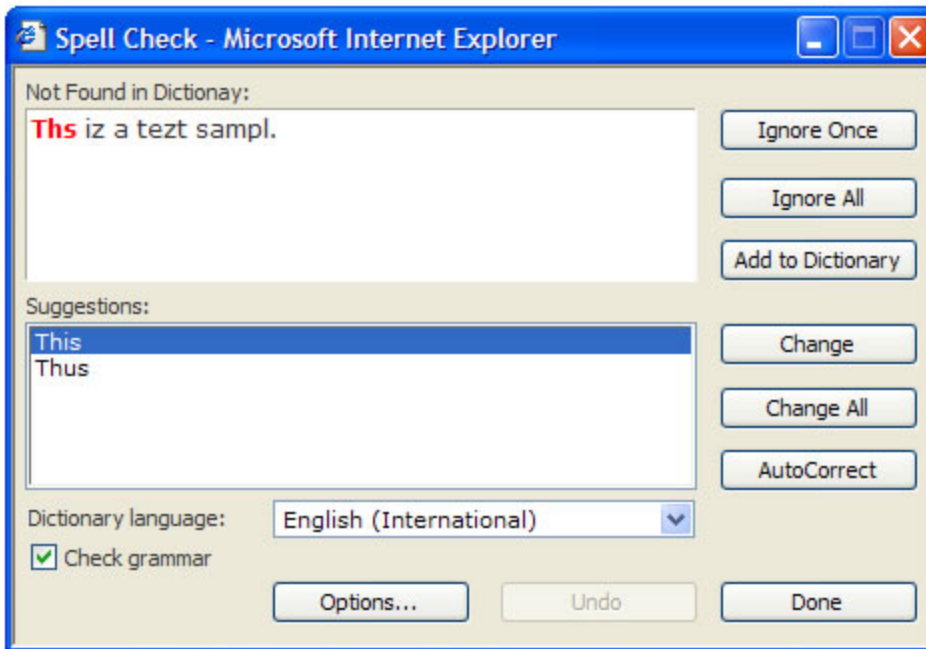
    <textarea name="myWYSIWYG" cols="30" rows="10" id="myWYSIWYG" >
        Hello World.
    </textarea>

<br>

    <%
        dim myLink
        set myLink = new AspSpellLink
        myLink.fields="iframe:0"
        response.write myLink.imageButtonHTML ("","","")
        set myLink=nothing
    %>

</body>
</html>
```

**Outcome:**



Spellchecking for the WYSIWYG HTML Editor

## Notes:

There are HTML numerous WYSIWYG editors available on the internet. A good resource to find one is <http://www.htmlarea.com>

For most WYSIWYG editors, you need to ASPSpellCheck the Iframe which the WYSIWYG uses. The Fields property should be set to **iframe:myWYSIWYGFrameId** .

If you don't know the ID of your editor, **iframe:0** points to the first frame or Iframe in the DOM. **iframe:1** points to the second etc...

Learn More about the [ASPSpellLink.fields](#) property.

## FCK Editor:

```
myLink.fields="iframe:MyTextArea__Frame.0"
```

Where **MyTextArea** is the ID of the Textarea to be spellchecked.



There is now an FCKeditor Plugin for ASPSpellCheck:  
<http://www.aspspellcheck.com/fck-editor-asp-spell-check-plugin.asp>

### **YUSASP Editor (formerly Innova) :**

---

```
myLink.fields="iframe:idContentoEdit1"
```

---

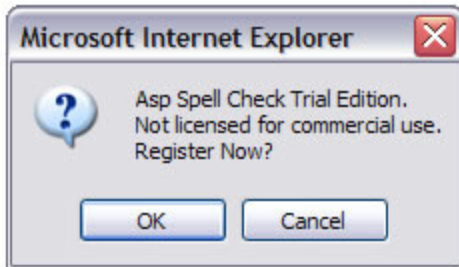
Where **oEdit1** is the JavaScript name for the editor.

## Free Trial

There is a freely downloadable trial edition of ASPSpellCheck which you may use to evaluate the software. It is [licensed](#) for evaluation only, and not for commercial or public use.

The trial edition presents a "Registration Reminder" from time to time. This is removed by installing a full version of [ASP SpellCheck](#).

### The "Registration Reminder":



## Registering

The trial is not [licensed](#) for commercial or public use!

### Buying Licenses

To purchase a suitable license for ASPSpellCheck please visit: <http://www.aspspellcheck.com/purchase.asp>

You will receive a unique license key. Copy this into the file **/aspspellcheck/regkey.asp**

**E.G:**

---

```
<% LicenseKey = "MY-LICENSE-12345-KEY-0001" %>
```

---

### Upgrading Licenses

You can also upgrade your licenses to affordably meet your expanding usage:

<http://www.aspspellcheck.com/upgrade.asp>

# License Agreement

ASPSpellCheck.com hereby grants you a non-exclusive license to the ASPSpellCheck Software Component (the Software). By downloading or using the Software, the Licensee agrees not to utilize the software in a manner which is disparaging to ASPSpellCheck.com, and not to rent, lease or otherwise transfer rights to the Software.

The Licensee agrees that no attempt will be made by the Licensee or associated parties to translate, reverse engineer, modify, decompile, disassemble or distribute the Software.

License terms are offered on the following terms, as purchased. If no purchase or insufficient purchase has been made then the Free Trial License terms apply.

**Free Trial License** - Allows the use of the TRIAL VERSION of the software for private evaluation purposes only. The software should not be used for public or commercial use of any kind.

**Single Website License** - Allows the use of the software on any number of web / intranet pages belonging to a single subdomain, hosted on a single server.

**Single Server License** - This license allows the usage of the software on any number of web / intranet pages for any number of domains located on a single server.

**Enterprise License** - This license allows the usage of the software on any number of web / intranet pages for any number of domains located on any number of servers belonging to one company or organization.

## UPGRADES

If a new release of the software is produced within 12 months from the date of purchase then you will be entitled to a free upgrade. This license does not grant you any right to any enhancement or update beyond the initial 12 month period, commencing from the date of purchase.

## COPYRIGHT

Title, ownership rights, and intellectual property rights in and to the Software shall remain with ASPSpellCheck.com. The Software is protected by the international copyright laws. Title, ownership rights, and intellectual property rights in and to the content accessed through the Software is the property of the applicable content owner and may be protected by applicable copyright or other law. This License gives you no rights to such content.

## LIMITATION OF LIABILITY.

THIS SOFTWARE IS PROVIDED "AS IS," WITHOUT A WARRANTY OF ANY KIND. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. ASPSPELLCHECK.COM AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THE SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL ASPSPELLCHECK.COM OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE SOFTWARE, EVEN IF ASPSPELLCHECK.COM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

## MISCELLANEOUS

This software is not designed or intended for use in on-line control of aircraft, air traffic, aircraft navigation or aircraft communications; or in the design, construction, operation or maintenance of any nuclear facility. Licensee represents and warrants that it will not use or redistribute the Software for such purposes.

New York law and controlling U.S. Federal law will govern any action related to this License. No choice of law rules of any jurisdiction will apply.

If any provision of this License is held to be unenforceable, such provision shall be reformed only to the extent necessary to make it enforceable.

This License represents the complete agreement concerning subject matter hereof.