

Super Spell Checker

.NET Edition

Version 2.0

Spellchecking and Thesaurus Control for .NET applications

© 2009 Software Siglo XXI
www.softwaresigloXXI.com

REFERENCE GUIDE

TABLE OF CONTENTS

Introduction..... 3

Version history..... 4

Installation and Deployment..... 5

Quick Start..... 6

User Dictionary 8

Method Reference..... 9

Using the low-level API..... 10

Introduction

Super SpellChecker .NET is a native 100% C# Spell Checking and Thesaurus control.

It has been designed to be very easy to use. You only need to drag the control from the toolbox palette to your forms and your text controls will gain runtime spell-checking feature.

Features:

- ❖ 100% .NET native component (C# written)
- ❖ Includes a Spell Checker supporting 6 languages: English, Spanish, Italian, French, German and Japanese
- ❖ Also includes a Thesaurus (synonyms) supporting 2 languages: English and Spanish
- ❖ Dialogs localized in 6 languages
- ❖ User dictionary
- ❖ Works with TextBox, RichTextBox (and derived) and TxTextControl controls
- ❖ Easy to distribute: only 1 DLL (component + compressed dictionaries)
- ❖ Can specify spellchecking/thesaurus language in design-time or run-time
- ❖ Invoke the spellchecker by pressing action key (F5) in runtime or call `doSpellChecking(yourTextControl)`.
- ❖ Accurate matching, including Levenshtein and SoundEx algorithms
- ❖ Invoke the thesaurus by pressing action key (F4) in runtime or call `doThesaurus(yourTextControl)`.
- ❖ Can specify action key to launch spellchecking or thesaurus (default keys are F4/F5)
- ❖ Includes a low-level API so you can get spell suggestions and synonyms for any word directly from code, without linking to a specific form or control

If you need further help about anything, don't hesitate to contact us at:
contact@softwaresigloxxi.com or visit our website at www.softwareSigloXXI.com.

Version history

Version 2.0

- Added the English and Spanish Thesaurus functionality
- Added low-level API support.
- Some bugfixes

Version 1.3

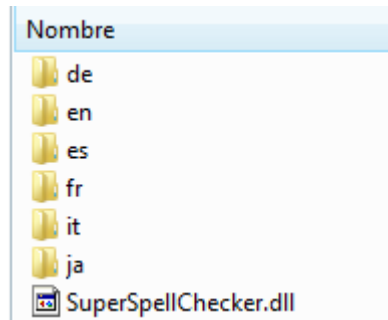
- Added `setSpellingCorrectMessage` property, so you can change or suppress the OK message shown if no errors are found (set it to "").
- Added `setUserDictionaryFile` property, so you can change the location of user dictionary (defaults to User Document folder).
- Some bugfixes

Version 1.0

- First release

Installation and Deployment

The spell checker control is provided as a single DLL (SuperSpellChecker.DLL) and several satellite assemblies for localization of the user dialog when a spell error is found. These satellite assemblies are optional – you can delete them if your application will run only in English environments.

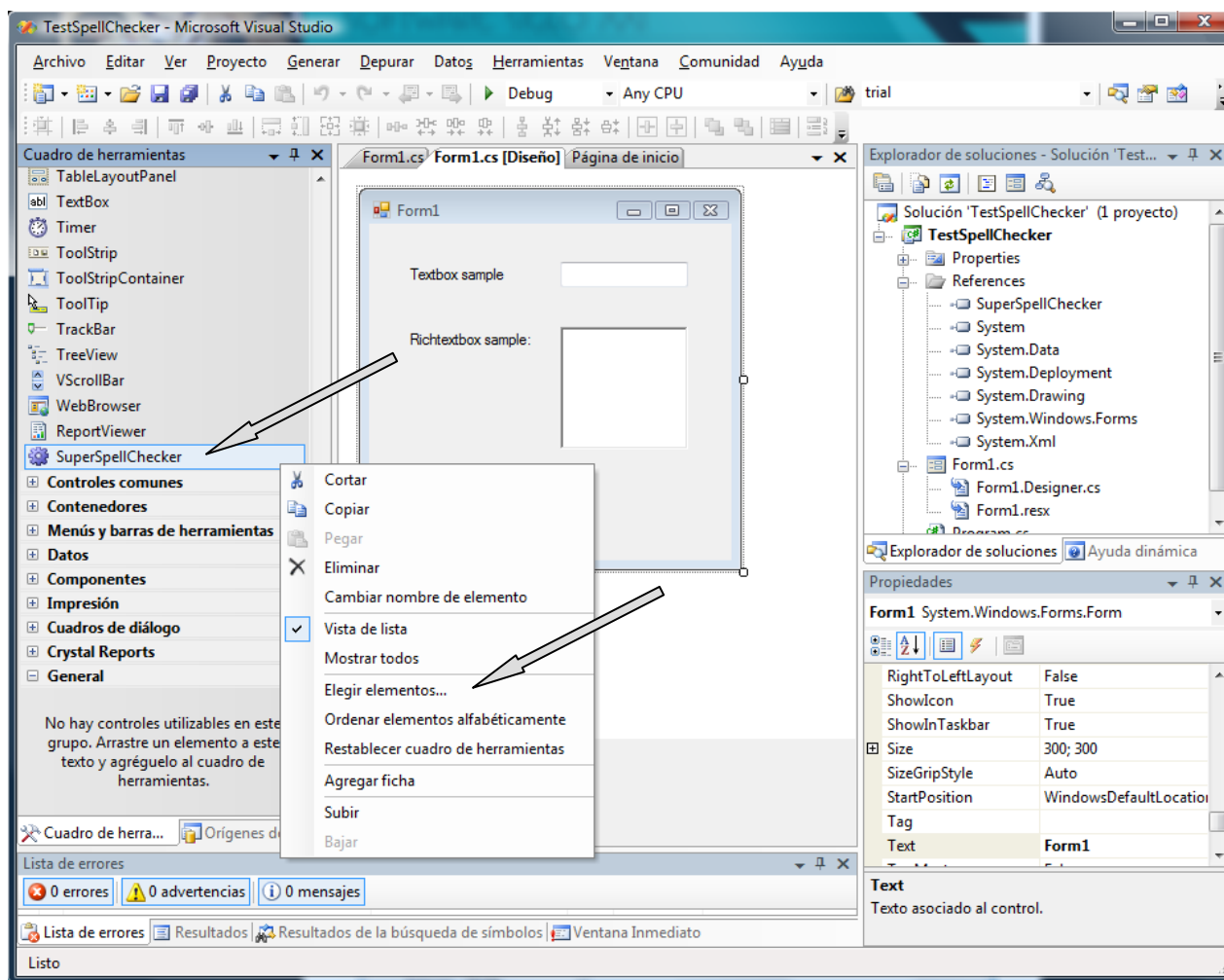


To deploy the control, include SuperSpellChecker.dll and above folders along your application main directory.

Quick Start

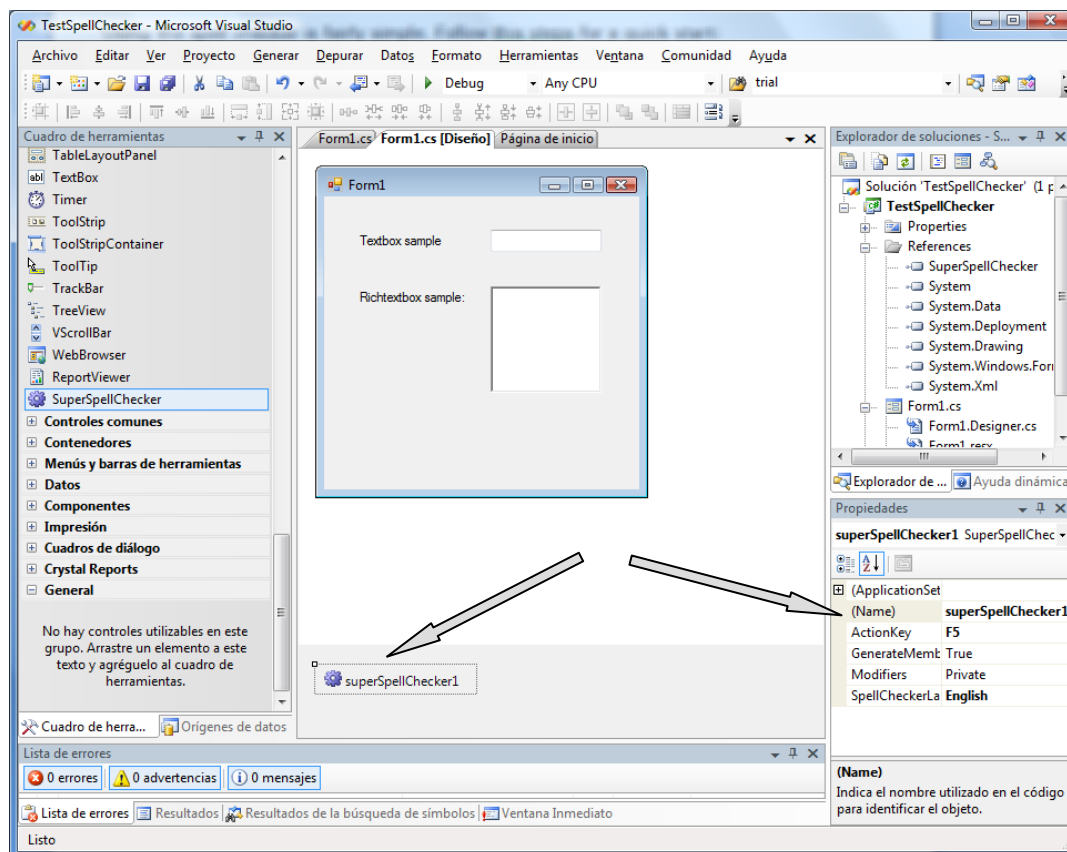
Using the spell checker is fairly simple. Follow these steps for a quick start:

Step 1.- Locate SuperSpellChecker.dll and add the library to the control palette:

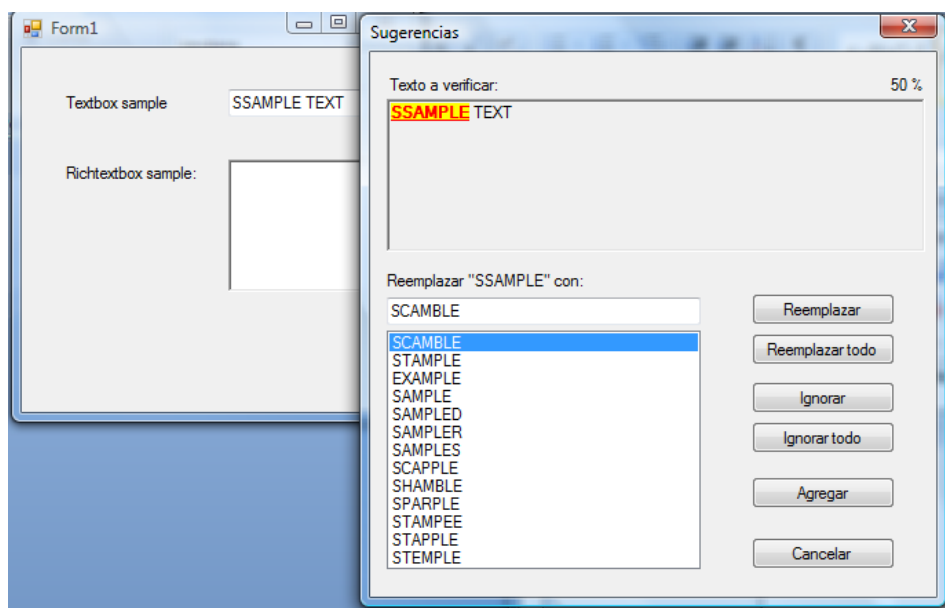


Super SpellChecker .NET Reference Guide

Step 2.- Drag the control to your form and set language and action key in property editor:



Step 3.- Run the application, enter some text and press action key (default is F5):



User Dictionary

The Spell Checker maintains a local database of user defined words. This dictionary is called "User dictionary" and is located in user documents root folder (default location – you can change in runtime using setUserDictionaryFile property).

This dictionary contains a list of words that are not included in the provided dictionaries, embedded in the DLL of the control.

You can manually edit this dictionary using Notepad for example. It's a plain-text file. Note that the separator between words must be either the space or the new line character.

You may also add words with the "-" prefix. This will mean that the word isn't accepted by your application, although it may be included in the standard dictionaries. For example, adding "-EUR" will raise a spell error if the word "EUR" is present in the text.

Method Reference

The Spell Checker control currently has the following method/properties which you can use in your code:

For Spellchecking:

- `SpellCheckerActionKey`: specifies the key used for invoking the spell checker (default: F5).
- `doSpellCheck(control)`: invoke the spell checker only for the specified control.
- `setSpellingCorrectMessage`: sets or suppress the message shown when no error found (set it to "" to suppress the message).
- `setUserDictionaryFile`: sets the full path for the user dictionary file.
- `SpellCheckerLanguage`: choose one of the six provided dictionaries (default: English).
- `getSpellCheckerSuggestions(word)`: returns a string array containing spell checking suggestions for "word".

For Thesaurus:

- `ThesaurusActionKey`: specifies the key used for invoking the thesaurus (default: F4).
- `doThesaurus(control)`: invoke the thesaurus only for the specified control.
- `ThesaurusLanguage`: choose one of the two provided synonyms dictionaries (default: English).
- `getThesaurusSynonyms(word)`: returns a string array containing synonyms for "word".

Using the low-level API

1. GETTING SPELL CHECKER SUGGESTIONS

This sample code will return spell checking suggestions for word "treex" (misspelled).

```
SuperSpellChecker ssc = new SuperSpellChecker();
ssc.SpellCheckerLanguage = SPELLCHECKER_AVAILABLE_LANG.English;
if (!ssc.IsWordSpellingCorrect("treex"))
{
    string[] words = ssc.GetSpellCheckerSuggestions("treex");
    StringBuilder sb = new StringBuilder();
    foreach (string w in words)
    {
        sb.Append(w + " ");
    }
    MessageBox.Show(sb.ToString());
}
```

2. GETTING SYNONYMS

This sample code will return synonyms for word "dark".

```
SuperSpellChecker ssc = new SuperSpellChecker();
ssc.ThesaurusLanguage = THESAURUS_AVAILABLE_LANG.English;
string[] words = ssc.GetThesaurusSynonyms("dark");
StringBuilder sb = new StringBuilder();
foreach (string w in words)
{
    sb.Append(w + "\n");
}
MessageBox.Show(sb.ToString());
```