Sentry Spelling-Checker Engine Demo

Welcome to the Sentry Spelling-Checker Engine demonstration program for Microsoft Windows!

The Sentry Spelling-Checker Engine (SSCE) is professional-quality spelling checker software which can be added to applications you develop.

To learn more about the SSCE, select one of the following topics:

Thank you for taking the time to try the Sentry Spelling-Checker Engine Demo program. For information not covered here, or to place an order, please contact Wintertree Software:

Sales: 1-800-340-8803
Support: (613) 825-6271
FAX: (613) 825-5521
E-mail: wsi@fox.nstn.ca
Web: http://fox.nstn.ca/~wsi

This demo program may be freely distributed, provided it is distributed in its entirety. The SSCE DLLs and dictionaries included herein may not be used for any purpose other than to support the SSCE demo program. In particular, they may not be called, read, reverse engineered, or invoked by an application program other than the Sentry Spelling-Checker Engine demo program.

About the Sentry Spelling-Checker Engine

The Sentry Spelling-Checker Engine -- SSCE -- is your first choice for application-callable spelling-checker software. The engine is fast, reliable, and field-proven on thousands of installations running on every major computer platform: MS-DOS, Microsoft Windows, Windows NT, OS/2, Macintosh, and UNIX. SSCE's application-program interface has been carefully designed and tested for ease of integration with your software. The English-language dictionaries provided with the SSCE SDKs (Software Development Kits) will meet or exceed the most demanding user's expectations of accuracy and comprehensiveness. Whether you need spelling checker software for a small, in-house project running on a few workstations or for a major, multi-platform application installed in thousands of sites world wide, SSCE is right for you.

SSCE is available in two forms. The Windows SDK is what you need if your application runs only on Microsoft Windows. The Source SDK is suitable if your application runs on multiple or non-Windows platforms, if you need to make special-purpose customizations, or if you just like the security of source-code availability. Each SDK includes everything you need to integrate SSCE with your application, and the software and dictionaries you need to distribute with your application. The Programmer's Guide provided with the SDKs describes not only the calling protocol for each function in the API, but is also packed with examples, tips, advice, and behind-the-scenes information that enables you to harness the power of the engine in your application with a minimum of fuss. Source code for working example applications is also provided.

The dictionary used by a spelling checker has the largest influence on its users' perception of quality. The dictionary must be large enough to cover most general-purpose text, yet small enough that its disk and memory requirements are manageable. And it goes without saying that the words in the dictionary must be scrupulously free of spelling errors. SSCE comes with superb American English and British English dictionaries you can distribute with your applications. The dictionaries contain about 100,000 words each, and are suitable for general-purpose spell checking. You can also create your own compressed dictionaries using the included compression utility.

Also included with the SDKs are redistributable auto-change and conditional-change dictionaries, containing hundreds of frequently misspelled words and their correct replacements. Common misspellings such as typing "recieve" instead of "receive" or "teh" instead of "the" can be corrected either automatically or after user confirmation.

Going international? Wintertree Software also has SSCE-compatible dictionaries available in Dutch, Finnish, French, Italian, German, Spanish, and Swedish. All dictionaries use accented characters as required by native language rules.

Spelling checkers and related text-processing tools are our business. SSCE isn't an add-on component we threw together quickly to round out our product line. We researched, engineered, built, and tested the SSCE to be the best spelling-checker engine available. From its release in 1992, we've continued to improve and support SSCE. If your users demand excellence, select SSCE.

Feature Summary:

* Can check the spelling of words **individually** or in **blocks** of text
* Windows SDKs include **built-in spelling-checker dialogs** that can check **edit controls**, **rich-edit controls**, and **text blocks**
* Can check **over 100,000 words per minute** (486-66, Win32)
* Can detect **capitalization errors** (e.g., "america," "thursday," etc.)
* Offers **suggested replacements** for misspelled words
* Suggestions can include **words defined in user dictionaries**
* Suggestions are returned with **scores** ranging from 0 to 100 which describe the degree

of closeness with the misspelled word

* Suggested words are returned in **order of decreasing score**, so the first word is the most likely correct replacement
* Suggestions can be offered based on **typographical** ("looks like"), **phonetic** ("sounds like") similarity, or both
* Supports **automatic and conditional word replacements** (e.g., replace "recieve" with "receive")
* Up to **16 dictionaries** can be open at once, including user dictionaries
* Can read and update **other applications' user dictionaries**, including Microsoft Word.
* Your application can **add and remove words** to/from user dictionaries
* Built-in support for "**Ignore All**" and "**Replace All**" processing
* Dictionaries and text being checked can contain **accented characters** (using ISO-8859 character sets)
* **Highly compressed dictionaries** hold over 100,000 words in just 331K
* **Dictionary compression utility provided** so you can create your own main dictionaries
* Can detect or ignore **capitalized words**, **words containing mixed case**, and **words containing embedded digits** or symbols
* **Doubled words** in text blocks (e.g., "Paris in the the spring") can be detected and removed
* Function interface is **straightforward and intuitive**, using only simple data types (integers and character strings).

See Also

API (Application Program Interface) Overview

About the Sentry Spelling-Checker Engine Demo Program

This demo program is a stand-alone Windows spelling checker application written to use the Sentry Spelling-Checker Engine.

The demo program can check and correct the spelling of text files and text entered into the edit area. In addition, you can set spell-checking options and edit the contents of the user dictionaries.

The main window of the demo program contains a large edit control. When you select the Check Spelling item in the File menu, the demo program passes the handle of the edit control to the SSCE DLL, which checks the spelling of words contained within the control. From this point on, the entire spell-checking operation is handled by the SSCE DLL. When a misspelled word is detected, the SSCE DLL highlights the word and displays a dialog to report the misspelling. If you replace the misspelled word with the correct spelling, the change is automatically reflected in the edit control.

SSCE can also check the spelling of individual words and text strings, using our dialogs or your own. Although a standard edit control was used in the demo, the SSCE Dialogs DLL works just as well with rich-edit controls and blocks of text.

The demo program is normally distributed with an American English dictionary. If you would prefer to try the demo with a British English dictionary, please contact Wintertree Software.

To see the SSCE DLL in action, try the following:

* Click on the Suggest button in the check-spelling dialog. If no suggestions appear, or the correct spelling isn't in the list, try clicking on the Suggest button again. Each time you click on Suggest, SSCE searches its dictionaries a little deeper. The button is disabled once all dictionaries have been completely searched.
* Try entering some words you frequently misspell into the demo program's edit area. Use the Suggest button to locate the correct spellings. Most often, the first word suggested is the correct replacement.
* Try joining two words in the edit area to create a misspelling (e.g., enter "thewords" instead of "the words"). Click on the Suggest button. Note how SSCE recognizes the two joined words and suggests separating them. This feature can be enabled and disabled via the "Suggest split words" item in the Options dialog.
* Try entering some common misspellings, such as "Recieve teh comittee's schedual." Note how SSCE automatically corrects some of the words, and conditionally corrects others.
* Try entering the same word twice in a row (e.g., "Paris in the the spring.") Note how SSCE detects the repeated word and lets you delete the second occurrence. This feature can be enabled and disabled via the "Report doubled words" item in the Options dialog.
* Load a large text file (up to 30K) into the edit area using the Open item in the File menu. Check the spelling, and replace any misspelled words, acronyms, etc. with correctly spelled words. Check the spelling again, and notice how quickly SSCE works.
* Using the Edit Dictionaries menu item or the Dictionaries button, add a personal word to the "uignore.tlx" user dictionary (e.g., your family name, street name, etc.) Enter this word in the edit area, but misspell it. Click on the Suggest button, and note that the correct spelling -- obtained from your user dictionary -- is suggested.
* Using the Dictionaries dialog again, select the uchange.tlx dictionary file. In the Words box, enter one of your favorite misspellings and its correct spelling, separated by a colon like this:

  resluts:results

  Click on the Add Word button. In the edit area, enter your misspelled word. Select the Check Spelling menu item and note that the misspelling is corrected automatically. If you had added the misspelling and its correct replacement to the "usuggest.tlx" user dictionary, you would have been prompted for confirmation before the replacement was made.

See Also

Using the Demo Program

Using the Demo Program

Select one of the following topics for instructions:

How to check a file

How to set spelling-checking options

How to edit user dictionaries

How to check a file

Use the following procedure to check the spelling of words in a text file:

1. Select the Open item in the File menu.
2. Use the Open File dialog to locate the file you wish to check. The file's contents are displayed in the demo program's edit area.
3. Select the Check Spelling item in the File menu.
4. If any spelling errors are detected, a dialog appears which you can use to correct or dispose of the problem. See Using the Check-Spelling Dialog for instructions.
5. If the file contains any words defined in the auto-change dictionary, the corrections will be substituted automatically.

How to set spell-checking options

You can set various options which control the way the SSCE checks the spelling of words. You can set the options in one of two ways. First, you can select the Spelling Preferences item in the Options menu. Second, you can select the Options button in the Check-Spelling Dialog. In either case, the Options dialog appears.

How to edit user dictionaries

The SSCE demo program uses several user dictionaries which you can edit in one of two ways: by selecting the Edit Dictionaries item in the Dictionaries menu, or by selecting the Dictionaries button on the Check-Spelling Dialog. In either case, the Dictionaries dialog appears.

Note that only the user dictionaries can be edited; the main dictionary cannot be edited.

About the SSCE Windows SDK

The SSCE Windows SDK includes a user interface for Microsoft Windows. Using the SSCE DLL, you can add a spelling-checker capability to your application by making a single function call. This SSCE demo program uses the SSCE DLL.

The SSCE DLL can check the spelling of text strings, standard edit controls, and rich-edit controls. If a spelling error is detected, the DLL displays a dialog with which the user corrects or disposes of the problem. If the user replaces a misspelled word with a correct one, the substitution is made directly into the text string or control.

The SSCE DLL also contains dialogs for setting spell-checking options and editing user dictionaries. These dialogs can be accessed by the user via buttons on the check-spelling dialog or via function calls made by your application.

The SSCE DLL makes it very easy to add a spelling checker to your applications. In fact, you can invoke a full featured spelling checker by adding just **one statement** to your code!

The SSCE Windows SDK includes both 16- and 32-bit DLLs, and comes with interfaces for C/C++, Delphi (VCL), and Visual Basic. Example applications written in C, Delphi, and Visual Basic are also provided.

SSCE Packages Available

The Sentry Spelling-Checker Engine is available in the following packages:

Windows SDK

The SSCE Windows SDK provides everything you need to add a full spelling-checker capability to your Microsoft Windows applications. The Windows SDK is compatible with most DLL-capable environments, including Visual C++, Borland C++, Visual Basic, Delphi, Microsoft Access, Paradox, PowerBuilder, FoxPro, etc.

A key feature of the Windows SDK is the built-in spelling checker dialog, which makes the full power of the SSCE -- including a complete user interface -- just one function call away. In less than 10 minutes, your application can have a powerful, professional-quality spelling checker.

Using the SSCE Windows SDK, your application can check the contents of text strings, standard edit controls, and Rich Edit controls. Misspelled words are highlighted in the controls as they're found, and replacements made by the user are updated instantly.

Your users will appreciate the flexibility afforded by having several user dictionaries open at once, or opening sets of user dictionaries for different purposes. SSCE can even read and update user dictionaries created by other applications, including Microsoft Word. All user-dictionary management is handled through dialogs contained within the SSCE DLL, so no additional programming is needed.

What about network-based applications? No problem. Network users can share a common main dictionary and site-wide custom dictionaries stored on a common server while updating user-specific dictionaries on their local disk drives.

Developers with special-purpose needs can bypass the built-in dialogs and access the SSCE API directly.

The Windows SDK includes 16- and 32-bit DLLs for the spelling-checker engine and dialogs; 100,000-word American- and British-English dictionaries; interfaces to the DLLs for C/C++, Delphi, and Visual Basic; Delphi VCL; programmer's guide, source code for example C, Delphi, and Visual Basic applications; dictionary compression utility; license to distribute an unlimited number of copies.

Source SDK

The SSCE Source SDK provides everything needed to add a powerful yet easy-to-integrate spelling checker to your C/C++ applications on virtually any platform.

The SSCE Source SDK includes the complete ANSI-C source code for the SSCE and related software. The code was designed and built for maximum portability. The SSCE is running right now in MS-DOS, Windows (16 and 32 bit), OS/2, Macintosh, and UNIX applications. The code makes no assumptions about the underlying hardware or operating system. All I/O is accomplished using ANSI-compatible STDIO calls. Getting SSCE up and running on a new platform typically takes less than one hour.

The SSCE source code was designed, coded, and tested with strict attention to quality, maintainability, and portability. No GOTOs. No cheesy coding hacks. No dependence on platform-specific features. Just consistent, well-commented, and modular ANSI C that's ready to be integrated with your applications today.

Dictionaries in Other Languages

Dictionaries in Dutch, Finnish, French, Italian, German, Spanish, and Swedish are available individually or as a package.

Other Products from Wintertree Software

Wintertree Software also products The ThesDB Thesaurus Engine. ThesDB includes a 50,000-synonym thesaurus database and software libraries which access the database from your applications.

For more information on ThesDB, please contact Wintertree Software.

Auto-Change Dictionary

The auto-change dictionary contains words which should be replaced by other words. Each entry in the auto-change dictionary contains a word (usually a misspelling) and its correction. The SSCE will automatically substitute the correction wherever the word appears.

SSCE API Overview

The Sentry Spelling-Checker Engine contains two API "layers": a basic API for the core spelling-checker engine, and an extended Windows API.

The basic API contains the following functions:

CheckBlock: Checks the spelling of words in a block of text -- a single word, a phrase, sentence, paragraph, or an entire document. CheckBlock can also optionally detect uncapitalized words, repeated words, words with embedded digits, etc.

CheckWord: Like CheckBlock, but checks one word at a time.

Suggest: Locates suggested alternative spellings for a misspelled word. Also produces a score, expressed as a percentage, showing the degree of correlation for each alternative word found.

CreateLex and **OpenLex**: Creates a new lexicon (dictionary) or opens an existing one. Up to 16 dictionaries can be open at once. Once opened, dictionaries are searched automatically by CheckBlock, CheckWord, and Suggest.

AddToLex and **DelFromLex**: Adds or removes words to/from lexicons.

GetLexInfo and **GetLex**: Obtains information about a dictionary, or obtains a copy of its contents.

The SSCE Windows API contains the following functions:

CheckBlockDlg: Interacts with the user to check the spelling of a text string.

CheckCtrlDlg: Interacts with the user to check the contents of a standard edit control or rich-edit control.

OptionsDlg: Interacts with the user to set spell-checking options.

EditLexDlg: Interacts with the user to manage and edit the user dictionaries.

SSCE Technical White Paper

Integrating the Sentry Spelling Checker Engine with an Application

White Paper

1. Introduction

This White Paper discusses some of the technical issues and requirements for integrating the Sentry Spelling-Checker Engine (SSCE) with an application. It is intended to aid application developers in planning and estimating.

2. Requirements

The SSCE software requires about 50K - 100K of memory at run time before any lexicons are opened. Compressed lexicons can be allocated a memory budget (with a minimum budget of about 40K). The SSCE uses a virtual memory scheme to ensure the memory needed for a compressed lexicon never exceeds its budget. An unlimited budget can also be specified for maximum performance; even with an unlimited budget, the compressed lexicon's memory use will never exceed the size of its disk image. Ignore- and exclude type text lexicons require an overhead of 4 bytes of memory per word at run time (in addition to the space required to hold each words characters). Change- and suggest-type text lexicons require an overhead of 8 bytes of memory per word at run time.

Compressed lexicons typically occupy 300K - 350K of disk space. Text lexicons are stored on disk as text files with one word per line.

3. Integrating SSCE with an Application

SSCEs API exists at two levels. The basic API allows access to the engine without providing a user interface. This API is available with all SSCE SDKs. At this level, your application can open and close dictionaries, check the spelling of words or text blocks, look up suggestions for misspelled words, etc.

The Windows API available only with the Windows SDK, adds Microsoft Windows specific features, including automatic initialization and built-in dialogs.

This section discusses use of SSCE at the basic API level. If your application runs under Microsoft Windows, see Using the SSCE Windows API.

Communication between an application and the basic SSCE API typically occurs in three stages: initialization, access, and termination.

3.1 Initialization

During initialization, the application forms a connection with the SSCE via the OpenSession function. OpenSession returns a session id which identifies the session in later function calls. The SSCE is designed to be called by multiple applications, and the session id represents the relationship between the SSCE and a single application. Most SSCE operations occur within the context of a session, so the actions of one application have no influence on another application.

Also during initialization, an application opens one or more lexicons. Opening a lexicon is very much like opening a file: a file name is provided, and a lexicon id is returned. The lexicon id is used to refer to the lexicon in subsequent function calls. The file name includes a directory path in a form used by the local operating system.

Because the path name is provided with each lexicon file, lexicons can exist in different directories, or even on different machines across a network. Updates made to text lexicons are synchronized to prevent unwanted over-writing.

SSCE supports two lexicon formats: compressed and text. Compressed lexicons typically hold in excess of 100,000 words. They are not modifiable at run time. Text lexicons are typically used as supplements to the main lexicons, and are modifiable at run time. Text lexicons which are modified at run time are often called "user dictionaries," because the lexicons contain words personal to the application's user.

SSCE supports four lexicon types:

Change-type lexicons contain words and their replacements. When a word is located in a change-type lexicon, SSCE requests the application to automatically substitute the replacement word. Change-type lexicons are always in text format.

Exclude-type lexicons contain words which should be considered misspelled, even if they are listed in a main lexicon. Words in exclude-type lexicons never appear in suggestion lists. Exclude-type lexicons are always in text format.

Ignore-type lexicons contain words which should be considered correctly spelled, and should therefore be ignored (skipped) by the spelling checker. Ignore-type lexicons can be in either text or compressed format.

Suggest-type lexicons contain words and their replacements. When a word is located in a suggest-type lexicon, SSCE requests the application to conditionally substitute the replacement word. The application is expected to obtain the user's confirmation before making the replacement. Suggest-type lexicons are always in text format.

An application can also create new text-format lexicons at run time. A new lexicon can be temporary, meaning it exists entirely in memory and is never written to disk, or permanent, meaning it is written to disk when it is updated. Temporary lexicons are useful for holding words the user wants to ignore or to replace with another word.

The application is responsible for opening all lexicons, including main lexicons, user lexicons, and temporary lexicons.

3.2 Accessing the Spelling-Checker Engine

Following initialization, the application can access the SSCE to check the spelling of text, locate suggested replacements for misspellings, and update user lexicons.

3.2.1 Checking the Spelling and Form of Text

SSCE has two functions for checking the spelling of text: CheckWord and CheckBlock.

The CheckWord function checks the spelling of an individual word. A flag is returned to the caller indicating whether the word is spelled correctly.

The CheckBlock function checks the spelling of a text block. A text block can be any amount of text, ranging from a few words to an entire book. Before a text block can be checked, it must be opened via the OpenBlock function. The CloseBlock function closes an open block. Words in a text block must be separated by white space or ASCII control characters. Because a block of text may contain many misspelled words, the CheckBlock function is called in a loop which terminates when CheckBlock indicates the end of the block has been reached. The CheckBlock function returns when it encounters a word which needs the application's attention, or when all words have been checked. When CheckBlock reports a misspelled word, the application can call other SSCE functions to obtain the offset of the word from the start of the text block (so the misspelled word can be highlighted, for example) or replace the word with another word. Any replacements are made directly to the contents of the block.

In addition to spelling errors, the SSCE can also detect several word form problems, including uncapitalized words, words containing embedded digits or symbols, and words containing unusual combinations of upper and lower-case letters.

Both the CheckWord and CheckBlock functions may also return indications that the word should be automatically or conditionally replaced with another word.

3.2.2 Locating Suggested Replacements for Misspelled Words

The Suggest function is used to obtain a set of candidate replacements for misspellings. The Suggest function returns a set of suggestions and a score for each word. The score indicates how closely the suggested word matches the misspelled word. The application can use the score values to limit the display of suggested words based on some criteria e.g., only words with scores in the top 20%.

The Suggest function attempts to locate replacements in all open lexicons, including user lexicons. However, words contained in exclude-type lexicons are never offered as suggestions.

Because locating suggestions can be time consuming, the Suggest function accepts a parameter which indicates the depth of the search. On a very fast computer, the application might always request the deepest search. On a slower computer, the application could allow the user to repeatedly request suggestions for the same misspelled word, using a deeper search-depth with each request.

The Suggest function determines its suggestions based on typographical similarity, phonetic similarity, or both, to the misspelled word.

3.2.3 Updating User Lexicons

At run time, the application can update user lexicons. These updates may occur for several reasons, each of which is described below. Words are added via the AddToLex function, and deleted via the DelFromLex function.

Spelling checkers typically have Ignore-All commands, which indicate that any further occurrences of a word flagged as misspelled should be skipped (e.g., a correctly spelled proper name). Such words can be added to a temporary ignore-type lexicon.

Spelling checkers also have Replace-All commands, which indicate that any further occurrences of a word flagged as misspelled should be replaced with another word (e.g., a misspelling the user knows occurs more than once). Such words can be added to a temporary change-type lexicon.

Spelling checkers may support Add to Dictionary commands, which indicate that a word flagged as misspelled is actually correctly spelled and should never be reported as a misspelling again (e.g., specialized terminology not present in the general-purpose main lexicon). Such words can be added to a permanent ignore-type lexicon.

SSCE's text lexicons are stored as text files, with one word per line. The first line is a header which contains information about the lexicon, such as its type and the language of the words it contains.

3.3 Termination

When an application no longer needs to use SSCE, it should close any open blocks (via the CloseBlock function), lexicons (via the CloseLex function), and sessions (via the CloseSession) function.

4. Using the SSCE Windows API

The SSCE DLL provides an easy-to-use spelling-checker component for developers of Microsoft Windows applications. The SSCE DLL performs most of the initialization and termination functions described in section 3.

The SSCE DLL contains the basic SSCE API (described in section 3) and an extended Windows API.

4.1 Configuration

The SSCE DLL was designed to simplify integration with an application. Most of the configuration information is stored in the SSCE.INI file or system registry, and is read and processed by the DLL automatically. Once the SSCE.INI file or system registry has been set up, an application can invoke the spelling checker by calling a single function.

The SSCE.INI file or system registry contains two key profile items: MainLexPath and

UserLexPath. MainLexPath defines the path to the directory containing the main lexicon and other common lexicons. UserLexPath defines the path to the user's personal lexicons.

On a network, MainLexPath can point to a directory on a file server, while UserLexPath points to a directory on the user's local disk drive. This allows the main lexicon to be shared among all users, while the user lexicons can be private to each user.

Applications with special needs can circumvent some of this behavior at run time. For example, the SSCE DLL defines a function which allows the application to specify a different UserLexPath setting.

4.2 Spell Checking

The SSCE DLL has two interface functions to check the spelling of text.

The CheckBlockDlg function checks the spelling of words in a block of text. Any changes made by the user (e.g., word replacements) are made to the text directly, so the corrected text is returned to the application when CheckBlockDlg returns.

The CheckCtrlDlg function checks the spelling of text in a Windows control. CheckCtrlDlg can check the contents of edit controls and Rich Edit controls. It can check any control which returns text in response to a GetWindowText call. In the case of edit and rich-edit controls, CheckCtrlDlg can check all the text or just the portion which is selected. Misspelled words can be highlighted in the control as they are detected. Any replacements made by the user are reflected in the control immediately.

If a misspelled word is detected by either CheckBlockDlg or CheckCtrlDlg, a dialog is displayed with which the user interacts to correct the problem. Once the dialog is displayed, it remains visible until all words have been checked. The dialog contains buttons which permit the user to set spelling-checker options or edit the user lexicons.