Writing Tools API

The Writing Tools API

This document explains the Writing Tools Application Programming Interface (WTAPI). This API is being developed by WordPerfect, Corel Corporation in conjunction with third-party companies interested in the development of integrated writing tools. The WTAPI is a group of functions and definitions in a source code file that you include in your program source code. It is designed to allow the integration of any writing tool with any client application (for example, a word processor) through inter-process communication.

The graphic below represents an overview of the different parts of the interface.

For this document a *writing tool* is defined as an application that provides services such as spell checking, thesaurus lookup, grammar checking, translation, and so forth. On occasion a writing tool will be referred to as a *server*.

A *writing tool user* or *writing tool client* is an application that uses a writing tool.   Typically, WordPerfect will act as the writing tool client.

WordPerfect Corporation has recognized a need in the software industry to allow writing tools to integrate easily with word processors and other client applications. Currently, developers of writing tool software either include their code directly in the code of a word processor (through licensing or other agreements) or they must invent a makeshift method of capturing the text from the word processor, making changes, and returning the text. This forces developers to invest research time and money to develop makeshift solutions. Newer writing tool technologies are often overlooked by users, simply because an older technology has been included in a word processing package. The Writing Tools API is designed to address these problems.

The WTAPI allows tight integration of a writing tool and its client by providing communication between them. Information starts in the tool or client application and is passed to the WTAPI in system-independent function calls. The information flows from the WTAPI to the system-specific communications link using the specific packaging and function calls that are available for a given operating system. The operating system then passes the information to the opposite application.

Once the communication is received by the opposite tool or client application the data flow process is reversed. The data moves from the system-specific communication, is unpackaged by the WTAPI, and is then passed to a callback function of the application.

After some initial setup, the user may invoke the writing tool directly from the client. The two applications determine the protocols that will be used during the writing tool session. The writing tool becomes the foreground application and uses the Writing Tools API to request blocks of text from the client. The writing tool then processes this text, prompting the user for input as needed. The writing tool communicates changes by first telling the client the position in the document where the change should be made. It then tells the client how much text to delete (if any) and gives it a buffer of replacement text (which may be empty). The client follows these instructions, making the changes in its own text buffer.

The Writing Tools API Documentation

This document is divided into the following sections:

Writing Tool Session

This section outlines the different components of the API that comprise a writing tool session. Installation and initiation, message transmission, initialization, tool data areas, text transmission, text correction, user editing, and termination are covered.

Writing Tools API Reference

This section contains complete references for all C functions, enumerated types, symbolic constants, platform-specific types, and API messages.

UNIX Platform-Specific Information

The Writing Tools API was developed to be as platform independent as possible. This section contains information necessary to use it under UNIX.