UNIX Platform-Specific Information

This section discusses platform-specific information necessary for using the Writing Tools API (WTAPI) on UNIX platforms. It includes information on installing a writing tool, the UNIX Transport Protocol, and WTAPI platform-specific structures and types. It also explains how to invoke a writing tool from within WordPerfect for UNIX.

Installing a UNIX Writing Tool

When a writing tool is installed on a computer system, the installation program needs to place information about the tool in a writing tools information file. For UNIX, this file is named wtapi.init, and is currently located in the shlib10 directory for installations of WordPerfect for UNIX. This location allows applications to access information about the writing tool so that the application can invoke the writing tool. In the future, as non-WordPerfect clients begin writing to this API, the wtapi.init file should be moved to a more company-independent location. There is no UNIX standard that dictates the location for this file. The location will eventually need to be negotiated by the company's that provide WTAPI-compliant clients.

The installation program is required to create a new item in the [Writing Tools] section of the wtapi.init file. This item's keyname cannot conflict with any existing keynames in the [Writing Tools] section of the wtapi.init file. After creating the new item, the installation program must create a new section in the wtapi.init file. The name of the new section must be the same as the keyname inserted in the [Writing Tools] section. Required entries in this section include the PathExe and the MenuName. There are several other optional entries that can be included to enhance or customize the operation of the writing tool. The section below provides an example of the wtapi.init file modifications related to installing a writing tool. It also gives the explanations and syntax for each related wtapi.init file entry.

Example wtapi.init File

In the following example, the installation program for WordPerfect's Thesaurus would be required to place the bolded information in the wtapi.init file. The italicized information is optional.

```
[Writing Tools]
Tool1=
Tool2=
WPThesaurus=WordPerfect Thesaurus
...

[WPThesaurus]
PathExe=/wp60/shbin10/xwpthes or shbin10/xwpthes
MenuName=Thesaurus
DescriptiveName=WordPerfect Thesaurus
Parameters=
LongPrompt=Display synonyms and antonyms for a word
ButtonBarText=Thesaurus
BitmapPath=
HelpFilePath=/wp60/shlib10/wpsh20us.hlp
HelpFileKey=#474
```

PathExe

This is the path and filename of the executable file for the writing tool being installed. It can be a full path or a relative path. If a full path is not provided, the client should look for the executable relative to a certain directory or mount point defined by the client. This allows multiple users to remotely mount file systems containing writing tools without being required to use the same paths. In WordPerfect, if the full path is not provided then WordPerfect looks for the executable in the wpbin and shbin10 directories relative to where WordPerfect was invoked. For example, if WordPerfect was invoked from the ../wp directory, WordPerfect would look for the writing tool executable in ../wp/shbin10 and ../wp/wpbin. The same is true if the full path is specified and it does not point to a valid executable but it does contain wpbin or shbin10. The entry's syntax is:

PathExe=*PathOfExeFile*

MenuName

This is the text that will appear on the menu of WordPerfect or any other client which chooses to have the writing tool installed on one if its menus. A single ampersand (&) may be inserted before any letter to indicate that it should be used as the mnemonic for that menu item. The entry's syntax is:

MenuName=*NameOfMenuItem*

DescriptiveName

A descriptive name for a writing tool can be used by a client during tools setup to help users quickly distinguish exactly what writing tools available. In WordPerfect, this entry is currently ignored. The entry's syntax is:

DescriptiveName=*DescriptiveNameForTool*

Parameters

Any parameters specified here will be added to the command line parameters that are passed to the writing tool at startup. The entry's syntax is:

Parameters=*ParameterList*

LongPrompt

This is a string that is used as the long prompt help for the writing tool. The entry's syntax is:

LongPrompt=*TextForLongPrompt*

ButtonBarText

If the user chooses to place a button on one of WordPerfect's Button Bars for this writing tool, this string will be used as the text of that button. The entry's syntax is:

ButtonBarText=*TextForButtonBar*

BitmapPath

If the user chooses to place a button on one of WordPerfect's Button Bars for this writing tool, this bitmap file will be used on the button. In WordPerfect, this entry is currently ignored and default bitmaps are used. The entry is included here for future compatibility. The entry's syntax is:

BitmapPath=*PathOfBitmapFile*

HelpFilePath

This is the path and filename for the writing tool's help file. In WordPerfect, the help file for a writing tool may be used if the user asks for help on that writing tool. Help is requested by highlighting the menu item and pressing F1, or by typing Shift-F1 and clicking the button with the mouse. The entry's syntax is:

HelpFilePath=*PathOfHelpFile*

HelpFileKey

If provided, the help file key may be used by WordPerfect when it uses the help file for a writing tool. The entry's syntax is:

HelpFileKey=#*KeyForHelpFile* (# is optional)

UNIX Transport Protocol

On UNIX platforms, the Writing Tools API (WTAPI) uses parent and child pipes as its method of communication between the writing tool and the client. The client is the parent application and invokes the writing tool as a child application. The following sections describe the protocol defined for this implementation of the WTAPI.

Initializing the Writing Tool Client

2 When a writing tool is selected, the client creates two pipes, one to write messages to the writing tool and the other to read messages from the writing tool. Each of these pipes consists of a read file descriptor and a write file descriptor.

4 The client registers WtClMsgReceived as an XtInputCallbackProc to be called whenever there is activity on the read pipe's read file descriptor. It sets the read pipe's read file descriptor and the write pipe's write file descriptor to be closed on the fork and execution of the writing tool.

6 The client forks and executes the writing tool using the path to the executable provided in the wtapi.init file. It must pass along the argument -wtapi followed immediately by the read and write file descriptors, such as /wp60/shbin10/xwpthes -wtapi 4 5. The client must also pass along any parameters specified in the wtapi.init file.

8 After executing the writing tool, the client closes the read pipe's write file descriptor and the write pipe's read file descriptor.

Initializing the Writing Tool

Upon invocation, the writing tool parses the command-line parameters and performs any related initialization. The writing tool registers WtTlMsgReceived as an XtInputCallbackProc to be called whenever there is activity on its read file descriptor.

Sending WTAPI Messages to the Writing Tool from the Client

WTAPI messages to the writing tool from the client are sent by writing to the client’s write file descriptor. The size of the message in bytes is written to the pipe first, immediately followed by the message itself.

Sending WTAPI Messages to the Client from the Writing Tool

WTAPI messages to the client from the writing tool are sent by writing to the writing tool’s write file descriptor. The size of the message in bytes is written to the pipe first, immediately followed by the message itself.

UNIX Platform-Specific Structures

These are the platform-specific structures used by the WTAPI as defined for use on UNIX platforms.

WTPIPE

A structure containing a read file descriptor and a write file descriptor.

Structure

```
typedef struct
{
    int read;
    int write;
} WTPIPE;
```

Types

WTPIPE

The WTPIPE Structure.

Members

read

File descriptor to read from.

write

File descriptor to write to.

WTCOMMDATA, WTCOMM

The WTCOMMDATA type is a structure containing platform-specific communication information. The WTCOMM type is a pointer to this structure.

Structure

```
typedef struct
{
    char        *client;
    char        *tool;
    WTPIPE      cl_pipe;
    WTPIPE      tl_pipe;
    XtInputId   inputid;
    Widget      toplevel;
    Window      window;
} WTCOMMDATA, *WTCOMM;
```

Types

WTCOMMDATA

The WTCOMM structure.

WTCOMM

A pointer to the WTCOMM structure.

Members

client

Client ID string.

tool

Tool ID string.

cl_pipe

Client’s read/write file descriptors.

tl_pipe

Tool’s read/write file descriptors.

inputid

ID of XtInputCallbackProc.

toplevel

Top-level shell of client or tool.

window

Window ID of communication partner.

WTLANG, WTLANGP

The WTLANG type is a structure containing a language designation. The

WTLANGP is a pointer to this structure.

Structure

```
typedef struct
{
    unsigned char    script;
    unsigned char    language;
    unsigned short   region;
} WTLANG, *WTLANGP;
```

Types

WTLANG

The WTLANG structure containing the language designation.

WTLANGP

A pointer to the WTLANG structure.

Members

script

Script of text.

language	Language of text.
region	Region (dialect) of text.

UNIX Platform-Specific Types

The table below describes the writing tools types that are specific to the UNIX platform. See *Symbolic Constants* in *Writing Tools API* for additional information.

Type Name	**Type Definition**	**Description**
WTBUF	unsigned char	Arrays of this type are used to hold WTAPI messages and data.
WTBUFP	unsigned char *	Pointer to WTBUF.
WTSIZE	unsigned	Usually used to indicate the maximum size of a message.
WTSIZEP	unsigned *	Pointer to WTSIZE.
WTCOUNT	int	Used for a variety of numeric data.
WTCOUNTP	int *	Pointer to WTCOUNT.
WTACTION	unsigned short	16-bit field.
WTDAMODE	unsigned short	16-bit field.
WTINFO	unsigned short	16-bit field.
WTTEXT	unsigned long	32-bit field.
WTUNIT	unsigned long	32-bit field.

Invoking a Writing Tool from WordPerfect for UNIX 7

Any writing tool which has placed its tool-specific information in the wtapi.init file may be invoked from WordPerfect. WordPerfect's regular installation includes a spell checker, thesaurus, and grammar checker. A user may install up to 10 additional writing tools on the WordPerfect *Tools* menu using the wtapi.init file. The user is then free to invoke any of these tools by displaying the *Tools* menu and clicking the appropriate menu item. Writing tools may also be placed on Button

Bars.