

WordScale Text and Word Capture SDK Document

WordScale™ Text and Word Capture SDK Document
Copyright © 2008 TMA Software Inc.

Overview	3
WordScale Requirements.....	4
WordScale Installation Location	5
WordScale Initialization APIs	6
WSAppInit	6
WSAppTerm.....	6
WordScale Text Retrieval APIs.....	7
WSGetWindowText	7
WSGetWindowTextFromRect.....	8
WSGetWordFromPoint	9
WSGetLineFromPoint	10
WSGetLineTextFromPointSel	11
WSGetSelectionText	12
WordScale Coordinate & Window Retrieval APIs.....	13
WSChooseWindowFinder	13
WSChoosePointFinder	14
WSSelectRect.....	15
WSGetActiveWindow	16
WordScale License and Information APIs	17
WSGetVersionInfo	17
WSSetRegistrationInfo	18
WSGetLastError.....	19
Text Type Description.....	20
Using WordScale APIs from Visual C++	21
Using WordScale APIs from .Net C#.....	23
Using WordScale APIs from Visual Basic.....	24
Using WordScale APIs from JavaScript	25
WordScale Installer.....	26
WordScaleInstall	27
WordScaleUninstall.....	28
Installing WordScale from InstallShield X	29
Installing WordScale using a C++ Application	30

Overview

The WordScale Text Capture Engine API provides an easy to use set of functions to extract text from any window.

The captured text can be returned in rich text or plain text format.

WordScale captures: The entire text of a window using **WSGetWindowText**, a rectangle area of a window using **WSGetWindowTextFromRect**, a word from a point on the window using **WSGetWordFromPoint** or an entire line of text from a point on the window using **WSGetLineFromPoint**. The current selection can be obtained using **WSGetLineTextFromPointSel**.

It is also possible to retrieve a word with **context** using the **WSGetLineFromPoint** and **WSWordFromIndex** functions. See the function **GetWordFromPointwithContext** function in the C++ example program.

WordScale API provides functionality so text can be extracted from the screen easily using the mouse or other input device. Text can be extracted from a user selected rectangle using the **WSSelectRect** API or text can be extracted from a window using the Window finder APIs **WSChooseWindowFinder** or **WSChoosePointFinder**.

The WordScale APIs can be used from a variety of applications and languages such as Visual C++ applications, Visual Studio .Net applications using C#, VB# etc. Visual Basic and JavaScript applications.

WordScale Requirements

WordScale works with the following Windows versions:

- Windows 2000
- Windows XP
- Windows XP SP2
- Windows Server 2003
- Windows Vista
- Windows Vista with User Account Control Enabled
- Windows Vista SP1
- Windows 7

WordScale on Windows 2000 requires Internet Explorer 6 or higher.

WordScale Installation Location

The WordScale.dll is installed to the following location and registered there:

Windows 2000 and Windows XP

C:\Documents and Settings\All Users\Application Data\TMA Software\WordScale

Windows Vista and Windows 7

C:\Users\All Users\TMA Software\WordScale

WordScale Initialization APIs

WSAppInit

Syntax

```
HRESULT WSAppInit();
```

Parameters

Return Value

Description

Call during application startup to initialize WordScale. If this method is not called WordScale does not work. Method only needs to be called once.

WSAppTerm

Syntax

```
HRESULT WSAppTerm();
```

Parameters

Return Value

Description

Call when application is shutting down.

WordScale Text Retrieval APIs

WSGetWindowText

Syntax

```
HRESULT WSGetWindowText([in] long hwnd, [in] eTextType textType, [out, retval] BSTR *pwindowText);
```

Parameters

Hwnd [in] - Window to retrieve text from.

textType [in] - Returned text format and layout.

eTextType_RichText	=	0
eTextType_PlainText	=	1
eTextType_RichText_NoLayout	=	2
eTextType_PlainText_NoLayout	=	3
eTextType_UniText	=	5
eTextType_UniTextNoLayout	=	6

Return Value

A string which contains the captured text in the specified format or NULL if no text is captured.

If NULL is returned WSGetLastError can be called to determine if an error occurred.

Description

Returns all text from the specified window.

WSGetWindowTextFromRect

Syntax

```
HRESULT WSGetWindowTextFromRect([in] long hwnd, [in] eTextType  
textType, [in] long left, [in] long top, [in] long right, [in] long  
bottom, [out, retval] BSTR *pwindowText);
```

Parameters

Hwnd [in] - Window to retrieve text from.

textType [in] - Returned text format and layout.

eTextType_RichText	=	0
eTextType_PlainText	=	1
eTextType_RichText_NoLayout	=	2
eTextType_PlainText_NoLayout	=	3
eTextType_UniText	=	5
eTextType_UniTextNoLayout	=	6

left [in] - Left screen coordinate of rectangle.

top [in] - Top screen coordinate of rectangle.

right [in] - Right screen coordinate of rectangle.

bottom [in] - Bottom screen coordinate of rectangle.

Return Value

A string which contains the captured text in the specified format or NULL if no text is captured.

If NULL is returned WSGetLastError can be called to determine if an error occurred.

Description

Returns text from rectangle in specified window.

WSGetWordFromPoint

Syntax

```
HRESULT WSGetWordFromPoint([in] long hwnd, [in] long x, [in] long y,  
[out, retval] BSTR * pwordText);
```

Parameters

Hwnd [in] - Window to retrieve text from.
x [in] - X screen coordinate of point.
y [in] - Y screen coordinate of point.

Return Value

A string which contains the word at the specified point.

If NULL is returned WSGetLastError can be called to determine if an error occurred.

Description

Returns word at specified coordinates in specified window.

WSGetLineFromPoint

Syntax

```
HRESULT WSGetLineFromPoint([in] long hwnd, [in] long x, [in] long y,  
[OUT] long *pcharIndexPt, [out, retval] BSTR *plineText);
```

Parameters

Hwnd [in] - Window to retrieve text from.
x [in] - X screen coordinate of point.
y [in] - Y screen coordinate of point.

pcharIndexPt[out] - Index of character at specified point.

Return Value

A string which contains the entire line of captured text.

If NULL is returned WSGetLastError can be called to determine if an error occurred.

Description

Returns line of text from point in specified window.

On return the variable pcharIndexPt contains the index of the character at the specified point position. So it is possible to determine the word at the specified point. The word at the point is retrieved using WSGetWordFromIndex function.

WSGetLineTextFromPointSel

Syntax

```
HRESULT WSGetLineTextFromPointSel([in] long hwnd, [in] eTextType  
textType, [in] long x, [in] long y, [in] long wantSel, [OUT] long  
*pcharIndexPt, [out, retval] BSTR *pwindowText);
```

Parameters

Hwnd [in] - Window to retrieve text from.

textType [in] - Returned text format and layout.

eTextType_UniTextNoLayout = 6

x [in] - X screen coordinate of point.

y [in] - Y screen coordinate of point.

WantSel [in] - Set to one if selected text should be returned.

pcharIndexPt[out] - Index of character at specified point.

Return Value

A string which contains the entire line of captured text plain text format or NULL if no text is captured.

If NULL is returned WSGetLastError can be called to determine if an error occurred.

Description

Returns line of text from point in specified window. Only returns text in Plain Text format.

If wantSel parameter is set then WordScale attempts to retrieve the selected text. The selection can be obtained from applications like Outlook, Office Applications, Notepad, WordPad and Internet Explorer.

The WSGetSelectionText must be used after to retrieve the selected text.

Retrieving text selection can add overhead to the amount of time this function takes to execute.

On return the variable pcharIndexPt contains the index of the character point position. So it is possible to determine the word at the specified point.

WSGetSelectionText

Syntax

```
HRESULT WSGetSelectionText ([out, retval] BSTR * pselection);
```

Parameters

Return Value

A string which contains the selected text if any was detected.

Description

Returns the current selection determined from a previous call to `WSGetLineTextFromPointSel`.

WordScale Coordinate & Window Retrieval APIs

WSChooseWindowFinder

Syntax

```
HRESULT WSChooseWindowFinder( [out, retval] long *phwnd);
```

Parameters

None.

Return Value

Window handle of the window found using the finder or NULL if no window selected.

If NULL is returned WSGetLastError can be called to determine if an error occurred or the action was cancelled.

Description

Activates the WordScale Finder tool. This allows the user to select a window to capture text from. As the user moves the cursor around the screen a highlighting rectangle is drawn around the selected window.

A user can cancel this action by pressing the escape key.

WSChoosePointFinder

Syntax

```
HRESULT WSChoosePointFinder([out] long *pX, [out] long *pY, [out, retval] long *phwnd);
```

Parameters

pX [out] - X screen coordinate of point.
pY [out] - Y screen coordinate of point.

Return Value

Window handle of the window found using the finder or NULL if no window selected.

If NULL is returned WSGetLastError can be called to determine if an error occurred or the action was cancelled.

Description

Activates the WordScale Finder tool. This allows the user to select a window to capture text from. As the user moves the cursor around the screen a highlighting rectangle is drawn around the selected window.

A user can cancel this action by pressing the escape key.

WSSelectRect

Syntax

```
HRESULT WSelectRect([out]long *pcancelled, [out] long *pleft, [out]
long *ptop, [out] long *pright, [out] long *pbottom, [out, retval] long
*phwnd);
```

Parameters

pcancelled [out] - Non zero if the action is cancelled.
pleft [out] - Left screen coordinate of rectangle.
ptop [out] - Top screen coordinate of rectangle.
pright [out] - Right screen coordinate of rectangle.
pbottom [out] - Bottom screen coordinate of rectangle.

Return Value

Window handle of the window or NULL if no window selected.

If NULL is returned WGetLastError can be called to determine if an error occurred or the action was cancelled.

Description

Activates the WordScale rectangle selection tool. The user can select a rectangle area of the window to select text from.

A user can cancel this action by pressing the escape key.

WSGetActiveWindow

Syntax

```
WSGetActiveWindow([out, retval] long *phwnd);
```

Parameters

phwnd [out] - Handle to the active window.

Return Value

Window handle to the active window.

Description

Returns the WordScale identified active window. This is usually the window that has the focus.

WordScale License and Information APIs

WSGetVersionInfo

Syntax

```
HRESULT WSGetVersionInfo([out, retval] VARIANT *pvariant);
```

Parameters

Return Value

Variant list containing WordScale version in different format.

Description

Returns WordScale version in string and number format. For an example how to use this API see the CWordscaleAppDlg::WordScaleGetVersion in the WordScale demo application.

WSSetRegistrationInfo

Syntax

```
HRESULT WSSetRegistrationInfo([in] BSTR bstrRegName, [in] BSTR  
bstrProductName, [out, retval] BSTR *pbstrErrorMsg);
```

Parameters

bstr bstrRegName [in] - Registration name used when installing WordScale provided by TMA Software.

bstr bstrProductName [in] - Your product name. Same name used with WordScaleInstall API.

BSTR pbstrErrorMsg [out] - If failed or demo expired then contains error message.

Return Value

NULL if successful registration anything else is failure.

Description

Easy to use API that verifies that the client of WordScale is valid.

Normally this is done once when your application starts if you have one WordScale interface object.

However, if you instantiate multiple WordScale interface objects then this call must be made for each instance.

During WordScale Installation the client specifies the Serial Number, Registration Name and Application Name to the WordScale installer. At runtime WordScale verifies that the Registration Name and Application Name are linked to a valid serial number.

When a WordScale license is purchased your company receives the Serial Number and Registration Name. You provide the Application Name to the WordScale installer during installation.

WSGetLastError

Syntax

```
WSGetLastError([out, retval] eWordScaleError * pLastError);
```

Parameters

None.

Return Value

Last WordScale error.

Description

Returns the last WordScale error.

eWSError_NoError: No error.
eWSError_Cancelled: Selection cancelled.
eWSError_SelectActive: Selection already active.
eWSError_TextCaptureActive: Capture engine in use.
eWSError_InvalidTargetWnd: Invalid target window specified.
eWSError_WSNotRegistered: WordScale is not properly registered.
eWSError_InvalidTargetProcess: Process is not a WordScale process.

Text Type Description

eTextType_RichText	=	0	Text is returned in rich text format.
eTextType_PlainText	=	1	Text is returned in plain text format.
eTextType_RichText_NoLayout	=	2	Text is returned in rich text format but no layout of the text is done all words are separated by spaces.
eTextType_PlainText_NoLayout	=	3	Text is returned in plain text format but no layout of the text is done all words are separated by spaces.
eTextType_UniText	=	5	Text is returned in unicode format.
eTextType_UniTextNoLayout	=	6	Text is returned in unicode format but no layout of the text is done all words are separated by spaces.

Using WordScale APIs from Visual C++

To use the WordScale COM object in your C++ application do the following:

#Import

add the following import statement to your C++ source file, like this:

```
#import "WordScale.tlb"
```

To use WordScale #import the type library wordscale.tlb located in the Include directory.

Name Space Reference

```
using namespace WordScaleLib;
```

CoInitialize

Initialize COM.

Instantiation

To instantiate a WordScale object using smart pointers do the following:

```
IWordScalePtr m_pWordScale;

HRESULT hr = m_pWordScale.CreateInstance(__uuidof(WordScale));

if (S_OK != hr)
{
    AfxMessageBox(_T("Failed to create WordScale Object.\nBe sure
                    Wordscale.Dll is registered with Regsvr32"),
                  MB_OK);
}
}
```

Registration Information

Call WSSetRegistrationInfo to set registration and product information be sure and check return value.

```
BSTR errMsg = m_pWordScale->WSSetRegistrationInfo("WordScale",
"WordScaleSdk");

if (errMsg)
{
    _bstr_t bstrerrMsg = errMsg;
    AfxMessageBox(bstrerrMsg, MB_OK | MB_ICONERROR);
}
}
```

Using WordScale API

```
HWND hwndActive = (HWND)m_pWordScale->WSGetActiveWindow();
```

Destruction

This example uses smart pointers so the object is automatically released when it goes out of scope. If you don't use smart pointers be sure and call the object's release.

Example

See the CPlus example in the WordScale SDK.

Using WordScale APIs from .Net C#

To use the WordScale COM object in your C# application do the following:

Solution Reference

Add a reference to WordScale by selecting References from the solution tree, right clicking and selecting the Add Reference menu item. Then select the COM tab and scroll down to WordScale type library.

Add Name Space to Source file

```
using WordScaleLib;
```

Instantiation

```
WordScaleLib.WordScaleClass m_WordScale = null;
```

```
m_WordScale = new WordScaleLib.WordScaleClass();
```

Registration Information

Call WSSetRegistrationInfo to set registration and product information be sure and check return value.

```
m_WordScale.WSSetRegistrationInfo("WordScale", "WordScaleSdk");
```

Using WordScale API

```
int hwndActive = m_WordScale.WSGetActiveWindow();
```

Example

See the CSharp example in the WordScale SDK.

Using WordScale APIs from Visual Basic

Project Reference

Add WordScaleLib to list of project references.

Instantiation

```
Dim wordScaleObj As New WordScaleLib
```

Using WordScale API

```
Dim hwnd As Long  
hwnd = wordScaleObj.WSGetActiveWindow()
```

Using WordScale APIs from JavaScript

All WordScale APIs can be called from JavaScript. For examples Web\wordscale_web.htm example in WordScale SDK.

Using WordScale API

```
function GetWindowTextFinder()
{
    var errMsg;
    var wsobj = new ActiveXObject("Wordscale.WordScale.1");

    //
    // When a new WordScale object is created call WSSetRegistrationInfo
    // to verify license set during install. This must be done once for each WordScale
    // object.
    //
    errMsg = wsobj.WSSetRegistrationInfo(REGNAME, YOURPRODUCTNAME);

    var varVariants = wsobj.WSChooseWindowFinderScript();
    var vbararray = new VBArray(varVariants);

    var error = vbararray.getItem(0);
    var hwnd = vbararray.getItem(1);

    if (!error &&
        hwnd)
    {
        var text = wsobj.WSGetWindowText(hwnd, 1);

        alert(text);
    }
}
```

WordScale Installer

WordScale comes with its own easy to use self contained installer. The WordScale installer, WordScaleInstaller.dll, installs WordScale on the target system.

Two WordScale installation examples are included with the WordscaleSdk.

1. InstallShield X Example.
2. Console Application.

The installer insures that WordScale is correctly installed on the target system. The installer also insures that WordScale is only removed when the last client uninstall is it. This is necessary because WordScale is a shared component used by multiple clients.

The installer is easy to use, runs silently and can be called from Installshield or any installation program.

The Installer DLL has two exported functions:

WordScaleInstall

Syntax

```
int WINAPI WordScaleInstall(PSTR pszregName, PSTR pszserialNumber, PSTR  
pszproductName)
```

```
typedef int (WINAPI *PFN_WordScaleInstall)(PSTR pszregName, PSTR  
pszserialNumber, PSTR pszproductName);
```

Parameters

pszserialNumber [**in**] - WordScale serial number received when product is licensed.

pszregName [**in**] - Registration name received when WordScale license purchased. Note this is the value used with the WSSetRegistrationInfo bstrRegName parameter.

pszproductName[**in**] - Your supplied product name must be at least 5 characters long. Note this is the value used with the WSSetRegistrationInfo bstrProductName parameter.

Return Value

The return value:

WSINSTALLER_SUCCESS: Success.

WSINSTALLER_REBOOT: WordScale install successful reboot required to complete.

WSINSTALLER_INVALID_SN: Invalid serial number.

WSINSTALLER_LICENSE_EXPIRED: License expired.

WSINSTALLER_INVALID_NAME - Invalid registration name specified.

>0: Win32 Error code

Description

WordScale installer installation function.

Header File

wordscale_install_examples\include\WordScaleInstaller.h

Lib

wordscale_install_examples\lib\WordScaleInstaller.lib

WordScaleUninstall

Syntax

```
int WINAPI WordScaleUninstall(PSTR pszregName, PSTR pszproductName)

typedef int (WINAPI *PFN_WordScaleInstall)(PSTR pszregName, PSTR
pszproductName);
```

Parameters

pszregName [**in**] - Registration name received when WordScale license purchased. Note this is the value used with the WSSetRegistrationInfo bstrRegName parameter.

pszproductName[**in**] - Your supplied product name must be at least 5 characters long. Note this is the value used with the WSSetRegistrationInfo bstrProductName parameter.

Return Value

The return value:

WSINSTALLER_SUCCESS: Success.

WSINSTALLER_REBOOT: WordScale removal successful reboot required to complete.

WSINSTALLER_MORE_CLIENTS: WordScale removal successful but there are more WordScale clients so WordScale has not been removed.

WSINSTALLER_INVALID_REGNAME: WordScale removal invalid registration name or product name.

>0: Win32 Error code

Description

WordScale installer removal function. If more than one client has WordScale installed then WordScale is not removed from the system only your reference is removed.

Header File

wordscale_install_examples\include\WordScaleInstaller.h

Lib

wordscale_install_examples\lib\WordScaleInstaller.lib

Installing WordScale from InstallShield X

Example Location

wordscale_install_examples\installshield_x\WordScale Example Installer Project.

WordScale Install

1. Add script "Script Files\WordScale.rul" to your project.
2. Update: SERIAL_NUMBER, REGNAME and YOURPRODUCTNAME constants.
3. From OnFirstUIAfter in Setup.url call InstallWordScale.

WordScale Uninstall

1. From DefaultFeature_UnInstalling in Setup.rul call
UninstallWordScale.

WordScaleInstaller.dll

Add WordScaleInstaller.DLL to the "Support Files \ Language Independent" section of the project.

This insures the WordScaleInstaller DLL is located in the <SUPPORTDIR> during installation/removal. This is where the WordScale.rul expects WordScaleInstaller.dll to be located.

Installing WordScale using a C++ Application

This example demonstrates installing and removing WordScale from C++ console application.

Example Location

wordscale_install_examples\cplusplus