## SendKeys Method

Sends one or more keystrokes to the active window (as if typed on the keyboard).

```
object.SendKeys(string)
```

## ⊟ Arguments

*object*
**WshShell** object.

*string*
String value indicating the keystroke(s) you want to send.

## ⊟ Remarks

Use the **SendKeys** method to send keystrokes to applications that have no automation interface. Most keyboard characters a represented by a single keystroke. Some keyboard characters are made up of combinations of keystrokes (CTRL+SHIFT+HON for example). To send a single keyboard character, send the character itself as the *string* argument. For example, to send the letter x, send the *string* argument "x".

> ✎ **Note**
>
> To send a space, send the string " ".

You can use **SendKeys** to send more than one keystroke at a time. To do this, create a compound string argument that represents a sequence of keystrokes by appending each keystroke in the sequence to the one before it. For example, to send keystrokes a, b, and c, you would send the string argument "abc". The **SendKeys** method uses some characters as modifiers characters (instead of using their face-values). This set of special characters consists of parentheses, brackets, braces, and th

- plus sign      "+",
- caret           "^",
- percent sign "%",
- and tilde      "~"

Send these characters by enclosing them within braces "{}". For example, to send the plus sign, send the string argument " {+}". Brackets "[ ]" have no special meaning when used with **SendKeys**, but you must enclose them within braces to accommodate applications that do give them a special meaning (for dynamic data exchange (DDE) for example).

- To send bracket characters, send the string argument "{[}" for the left bracket and "{]}" for the right one.
- To send brace characters, send the string argument "{{}" for the left brace and "{}}" for the right one.

Some keystrokes do not generate characters (such as ENTER and TAB). Some keystrokes represent actions (such as BACKSPACE and BREAK). To send these kinds of keystrokes, send the arguments shown in the following table:

| Key | Argument |
| --- | --- |
| BACKSPACE | {BACKSPACE}, {BS}, or {BKSP} |
| BREAK | {BREAK} |
| CAPS LOCK | {CAPSLOCK} |
| DEL or DELETE | {DELETE} or {DEL} |
| DOWN ARROW | {DOWN} |
| END | {END} |
| ENTER | {ENTER} or ~ |
| ESC | {ESC} |
| HELP | {HELP} |
| HOME | {HOME} |
| INS or INSERT | {INSERT} or {INS} |

| LEFT ARROW | {LEFT} |
|---|---|
| NUM LOCK | {NUMLOCK} |
| PAGE DOWN | {PGDN} |
| PAGE UP | {PGUP} |
| PRINT SCREEN | {PRTSC} |
| RIGHT ARROW | {RIGHT} |
| SCROLL LOCK | {SCROLLLOCK} |
| TAB | {TAB} |
| UP ARROW | {UP} |
| F1 | {F1} |
| F2 | {F2} |
| F3 | {F3} |
| F4 | {F4} |
| F5 | {F5} |
| F6 | {F6} |
| F7 | {F7} |
| F8 | {F8} |
| F9 | {F9} |
| F10 | {F10} |
| F11 | {F11} |
| F12 | {F12} |
| F13 | {F13} |
| F14 | {F14} |
| F15 | {F15} |
| F16 | {F16} |

To send keyboard characters that are comprised of a regular keystroke in combination with a SHIFT, CTRL, or ALT, create a compound string argument that represents the keystroke combination. You do this by preceding the regular keystroke with or or more of the following special characters:

| Key | Special Character |
|---|---|
| SHIFT | + |
| CTRL | ^ |
| ALT | % |

### Note

When used this way, these special characters are not enclosed within a set of braces.

To specify that a combination of SHIFT, CTRL, and ALT should be held down while several other keys are pressed, create a compound string argument with the modified keystrokes enclosed in parentheses. For example, to send the keystroke combination that specifies that the SHIFT key is held down while:

- e and c are pressed, send the string argument "+(ec)".
- e is pressed, followed by a lone c (with no SHIFT), send the string argument "+ec".

You can use the **SendKeys** method to send a pattern of keystrokes that consists of a single keystroke pressed several times i row. To do this, create a compound string argument that specifies the keystroke you want to repeat, followed by the number times you want it repeated. You do this using a compound string argument of the form {*keystroke number*}. For example, to send the letter "x" ten times, you would send the string argument "{x 10}". Be sure to include a space between keystroke an number.

### ✏ Note

The only keystroke pattern you can send is the kind that is comprised of a single keystroke pressed several times. For example, you can send "x" ten times, but you cannot do the same for "Ctrl+x".

### ✏ Note

You cannot send the PRINT SCREEN key {PRTSC} to an application.

## ⊟ Example

## ⊟ Description

The following example demonstrates the use of a single .wsf file for two jobs in different script languages (VBScript and JS Each job runs the Windows calculator and sends it keystrokes to execute a simple calculation.

```
<package>
    <job id="vbs">
        <script language="VBScript">
            set WshShell = WScript.CreateObject("WScript.Shell")
            WshShell.Run "calc"
            WScript.Sleep 100
            WshShell.AppActivate "Calculator"
            WScript.Sleep 100
            WshShell.SendKeys "1{+}"
            WScript.Sleep 500
            WshShell.SendKeys "2"
            WScript.Sleep 500
            WshShell.SendKeys "~"
            WScript.Sleep 500
            WshShell.SendKeys "*3"
            WScript.Sleep 500
            WshShell.SendKeys "~"
            WScript.Sleep 2500
        </script>
    </job>

    <job id="js">
        <script language="JScript">
            var WshShell = WScript.CreateObject("WScript.Shell");
            WshShell.Run("calc");
            WScript.Sleep(100);
            WshShell.AppActivate("Calculator");
            WScript.Sleep(100);
            WshShell.SendKeys ("1{+}");
            WScript.Sleep(500);
            WshShell.SendKeys("2");
            WScript.Sleep(500);
            WshShell.SendKeys("~");
            WScript.Sleep(500);
            WshShell.SendKeys("*3");
            WScript.Sleep(500);
            WshShell.SendKeys("~");
            WScript.Sleep(2500);
        </script>
    </job>
</package>
```

**Applies To:**

[WshShell Object](#)

## ⊟ See Also

**Reference**
[Run Method (Windows Script Host)](#)

---

# In Vbsedit, you only need to press F1 to get Help for the keyword under the cursor!

# [Download Now!](#)

[Download](#)   [Home](#)

Copyright © 2001-2024 adersoft