

# Send Keys and Send Key Events

## Introduction

Where a technical interface is unavailable for an application or an element within an application a Write stage might not work as a method of inputting text into screen elements. Where that is the case Blue Prism provide two text input methods that will work with any application:

- Global Send keys

Send Keys will work for most applications and should be tried before Send Key Events. It is a higher-level interface that sends keystrokes to the active application.

- Global Send Key events

Send Key Events will work for all applications and is the text sending method recommended for Citrix applications. It is a lower-level interface that mimics keyboard keystrokes in the operating system.

Send Keys and Send Key Events are part of the interface techniques that make up the Blue Prism Surface Automation interface. It is recommended that the full Surface Automation training course is completed if these Send Key methods need to be used.

Unlike other interfaces where a write stage will populate an element with text even when that element is not visible, there are some factors that need to be in place for Global Send Key interfaces to work robustly:

- The desktop screen must exist and be persistent. Send Key interfaces will not work if the desktop screen is locked or a screensaver is displayed
- The window that you want to send text into must be activated to be the topmost window of all running applications
- The element within the window you want to send text into must be focused so that the keyboard cursor is within it ready to enter text
- To ensure the application has time to react to any window or element focus navigation tiny delays are required between window activates, element clicks, and using Send Keys
- To ensure text is entered reliably a tiny delay should be placed between each keystroke. Experience in the use of Send Keys interfaces has shown that for some applications entering text too quickly can result in some characters not being correctly entered.

Where your use of Send Keys does not implement techniques to cater for the above factors it is likely that your solution will not work reliably.

This guide will detail how to use Send Keys robustly in different versions of Blue Prism. For a more in-depth understanding of this and other Surface Automation techniques please refer to the Surface Automation training course.

## When to use Send Keys or Send Key Events

Global keystrokes should be used when other methods of inputting text do not work, either because a thin client technology such as Citrix is being used, or because the element within an application either cannot be identified by application modeller or does not accept the use of a write stage.

If one interface method of sending text to an application does not work a developer should move onto trying a different method until a technique that works is found.

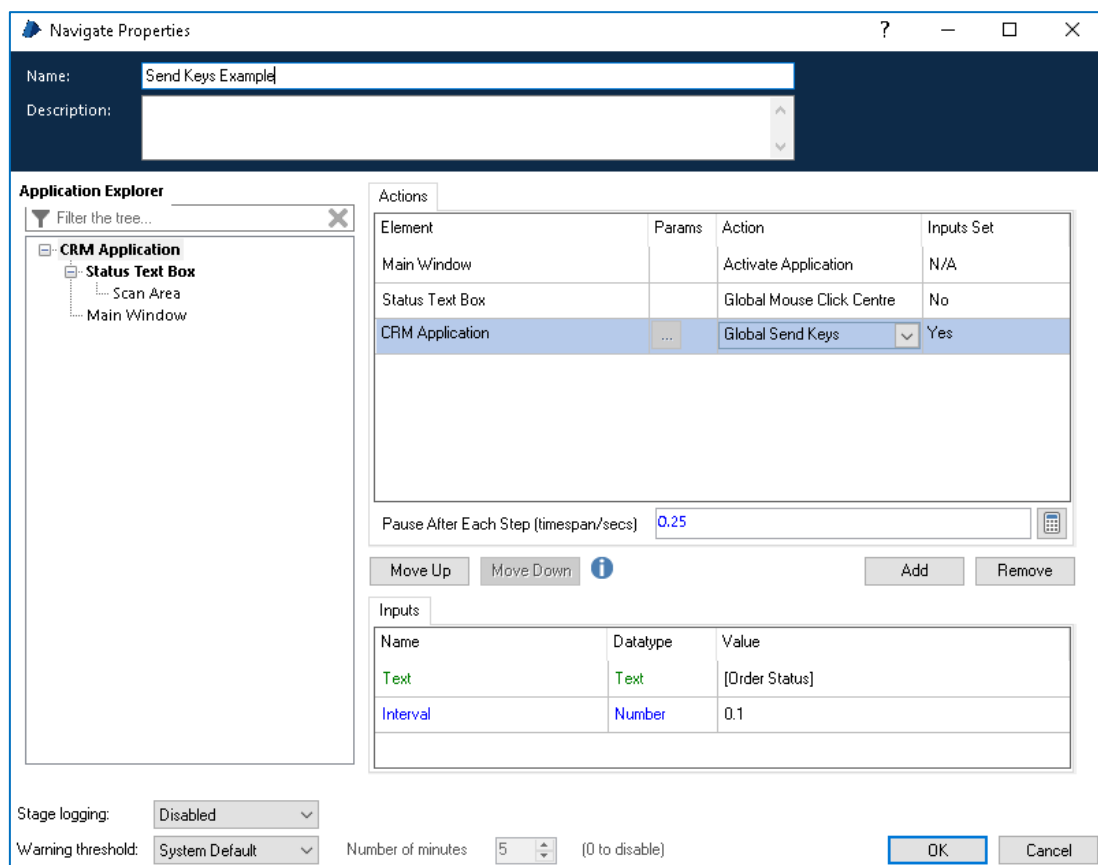
Different methods of entering text into an application should be attempted in the following order:

1. Write stage. Sending text using a write stage uses technological interfaces specific for the application type.
2. Windows Press Keys. For some applications this action in a navigate stage will work to send text. None of the window activation or element focus methods mentioned in this document are required.
3. Global Send Keys. This option should be tried before Global Send Key Events because it is a higher level interface and easier to use control keys.
4. Global Send Key events. This is the final option and should always work. It is the only option that will work when interfacing with Citrix applications. Even where Global Send Keys works this option can be useful for some use cases where a key needs to be held down on it's own whilst other actions are performed.

## Using Global Send Keys with Blue Prism version 6 or later

Blue Prism version 6 included some major improvements in the Surface Automation interfaces to make it far easier to implement the techniques required to robustly use Global Send Keys.

This image is of a version 6 navigate stage using the Global Send Keys action:

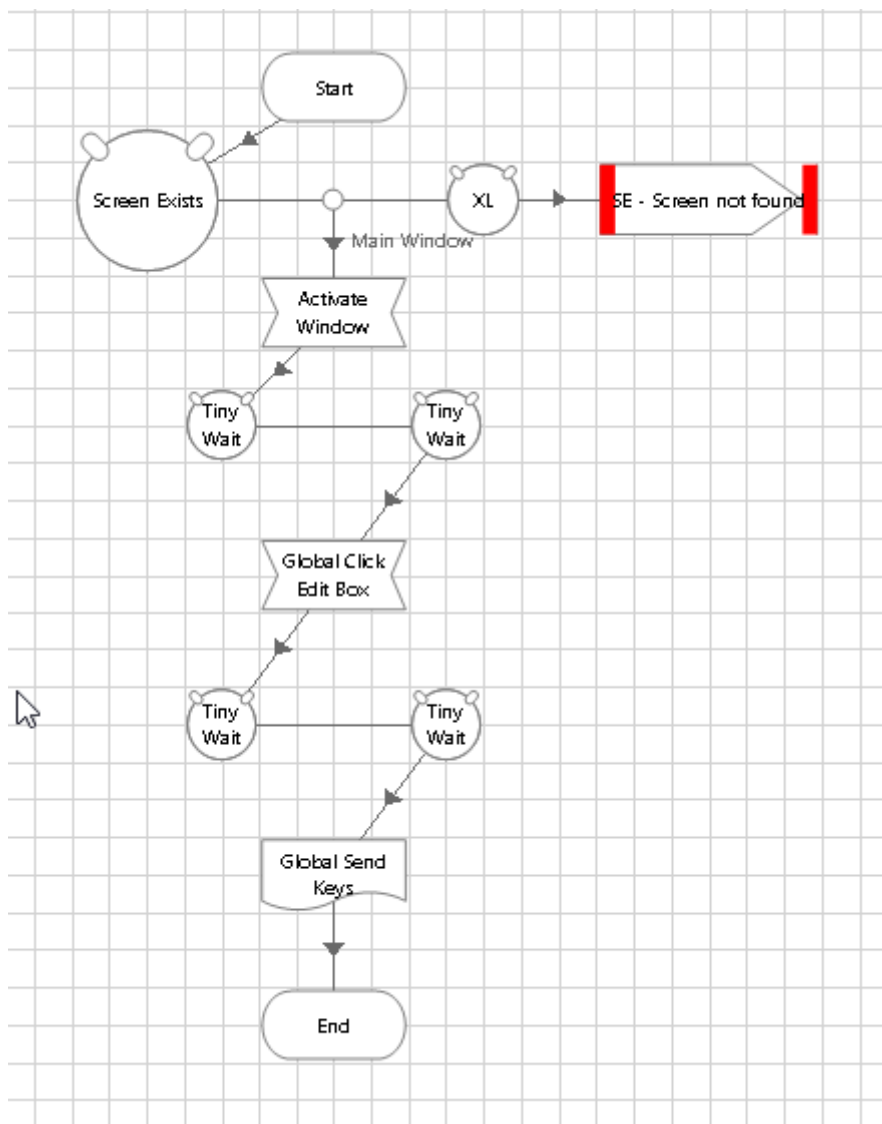


This Blue Prism version 6 navigate stage contains all the actions required for Global Send Keys to work robustly:

1. The Main Window of the application is brought to focus using the Activate Application action
2. The edit or text box we want to send keystrokes to is made active by using a Global Mouse Click Centre action
3. Global Send Keys to send the data to the application. The Input parameters to this action is the text we want to send and the interval between keystrokes which has been set to be 0.1 seconds
4. The 'Pause After Each Step' setting to 0.25 seconds. The correct pause number to for your own interface will depend upon the responsiveness of applications within your environment. This will usually be a number between 0.25 and 1 seconds.

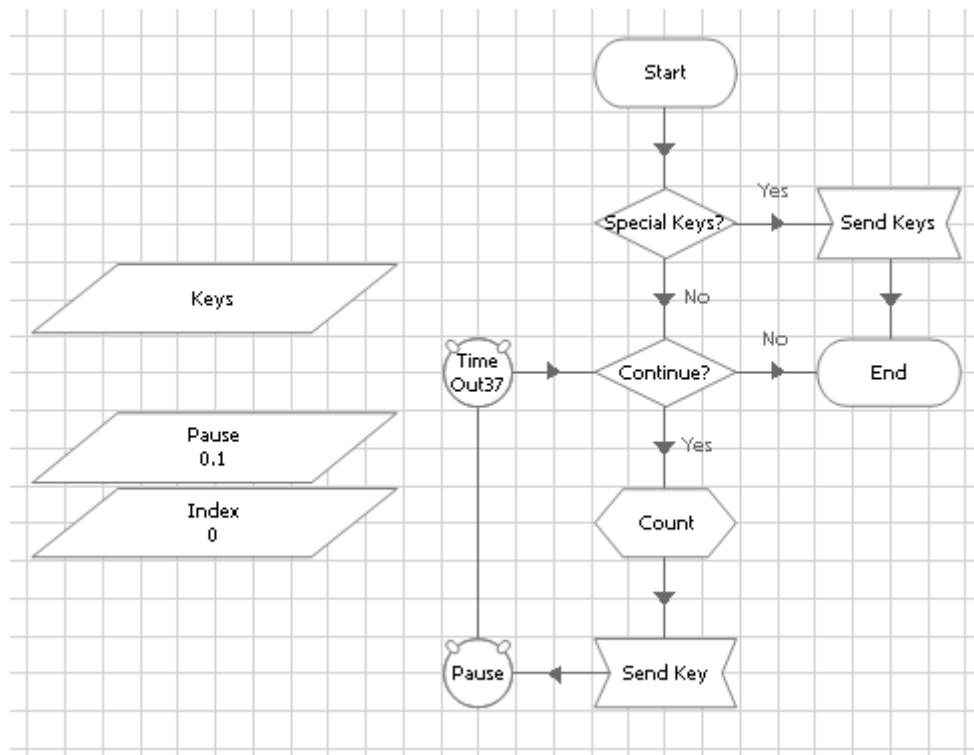
### Using Global Send Keys with Blue Prism version 5 or earlier

In earlier versions of Blue Prism the actions required to robustly use Global Send Keys could not be implemented in a single navigate stage. Instead the object flow to use send keys would look like the following images:



This Blue Prism version 5 flow shows all the same steps as the version 6 example, but they needed to be performed in separate flow diagram stages.

1. The Main Window of the application is brought to focus using the Activate Application action in a navigate stage
2. A tiny wait is added to give the application time to react to the activate. These tiny waits will be between 0.25 seconds and 1 second in duration depending upon your environment.
3. The edit or text box we want to send keystrokes to is made active by using a Global Mouse Click Centre action in a navigate stage
4. Another tiny wait is added to give the application element time to react to the mouse click and become focused
5. Global Send Keys to send the data to the application. To include logic with an interval between keystrokes a separate sub page has been used. The Global Send Keys sub page is shown in the following image:



The Sub Page adds each key one character at a time with a 0.1 second pause between each key. The count calculation increases the Index number by one and the text parameter in the Send Key navigate stage is `Mid([Keys], [Index], 1)`. If there are any special characters in the text to send (such as a shift or ctrl) then the 0.1 second delay logic is not used.

## Send Key and Send Key Events Syntax

When sending text using Global Send Keys and Global Send Key events there is a specific text format that must be used when sending control keys such as function keys, Shift, Alt, or Ctrl.

The different syntaxes are fully described in the sections below. However, the following table is a quick cheat sheet is offered which explains both quickly.

### Cheat Sheet

	Send Keys	Send Key Events
Summary	<ul style="list-style-type: none"> <li>Higher level</li> <li>Sends keys to the application which has focus</li> </ul>	<ul style="list-style-type: none"> <li>Lower Level</li> <li>Sends simulated keystrokes to the Operating System</li> <li>Any application which has focus will receive and process these keystrokes</li> </ul>
Highlights	<ul style="list-style-type: none"> <li>SHIFT: +</li> <li>CTRL: ^</li> <li>ALT: %</li> <li>Special keys must be enclosed in braces. i.e. {HOME}</li> <li>To repeat a special key n times, indicate after key name: i.e. {HOME 10}</li> <li>To apply Shift, Alt or Control to more than one key, enclose in parenthesis. i.e. +(EC)</li> </ul>	<ul style="list-style-type: none"> <li>SHIFT: {SHIFT}</li> <li>CTRL: {CTRL}</li> <li>ALT: {ALT}</li> <li>Special keys must be enclosed in braces. i.e. {HOME}</li> <li>Press key(s) down: &lt;</li> <li>Release key(s): &gt;</li> </ul>
Example: Type "Blue Prism", then press Ctrl+A to select text and then Ctrl+C copy to clipboard	Blue Prism^A^C	<{SHIFT}b>{SHIFT}lue <{SHIFT}p>{SHIFT}rism<{CTRL}AC>{CTRL}

### Send Keys

Send Keys talks directly to the application that an object is connected to. This method uses the following mnemonics to indicate for the Shift, Control and Alt keys:

- Shift: +
- Control: ^
- Alt: %

Special keys like for example Home must always be enclosed in braces. i.e. {HOME}.

To specify that any combination of Shift, Control and Alt should be held down while several other keys are pressed, enclose the code for those keys in parentheses. I.e. to specify to hold down Shift while E and C are pressed, use "+(EC)". On the other hand, to tell the application to hold down Shift while E is pressed followed by C without Shift, use "+EC".

Send Keys offers the option to repeat a key press a given number of times. For such functionality the number of times must appear after the name of the key. I.e. {LEFT 10} tells Send Keys to press the Left arrow key 10 times.

### Table of Send Keys codes:

Key	Code
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} (reserved for future use)
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}
Keypad add	{ADD}
Keypad subtract	{SUBTRACT}
Keypad multiply	{MULTIPLY}
Keypad divide	{DIVIDE}

Further technical details on how Windows treats Send Keys can be found in the following link:  
<https://msdn.microsoft.com/en-us/library/system.windows.forms.sendkeys.send.aspx>

## Send Key Events

Send Key Events communicates with the operating system and simulates keystrokes coming from the keyboard. Just like a person working on a computer, the keystrokes will be processed by the application which has the active focus. This method is usually, but not exclusively, useful when working with virtual desktops like Citrix, where standard application spying is not available and automation has to rely on techniques such as Surface Automation. This method uses the following mnemonics to indicate for the Shift, Control and Alt keys:

- Shift: {SHIFT}
- Control: {CTRL}
- Alt: {ALT}

Special keys like for example Home must always be enclosed in braces. i.e. {HOME}.

To tell Send Key Events that either Shift, Control or Alt must be pressed, they must be preceded with the “>” sign. Similarly, “releasing” a key must be preceded with the sign “<”. I.e. To send the Ctrl+c combination, the following sequence must be indicated: “<{CTRL}c>{CTRL}”. This tells the system to press the Control key plus the C key and then release Control.

Send Keys offers the option to repeat a key press a given number of times. For such functionality the number of times must appear after the name of the key. I.e. {LEFT 10} tells Send Keys to press the Left arrow key 10 times.

### Table of Send Key Events codes:

Key	Description
A	The A key.
Add	The add key.
Alt	The ALT modifier key.
Apps	The application key (Microsoft Natural Keyboard).
Attn	The ATTN key.
B	The B key.
Back	The BACKSPACE key.
BrowserBack	The browser back key (Windows 2000 or later).
BrowserFavorites	The browser favorites key (Windows 2000 or later).
BrowserForward	The browser forward key (Windows 2000 or later).
BrowserHome	The browser home key (Windows 2000 or later).
BrowserRefresh	The browser refresh key (Windows 2000 or later).
BrowserSearch	The browser search key (Windows 2000 or later).
BrowserStop	The browser stop key (Windows 2000 or later).
C	The C key.
Cancel	The CANCEL key.
Capital	The CAPS LOCK key.
CapsLock	The CAPS LOCK key.
Clear	The CLEAR key.
Control	The CTRL modifier key.
ControlKey	The CTRL key.
CrSel	The CRSEL key.
D	The D key.
D0	The 0 key.
D1	The 1 key.
D2	The 2 key.
D3	The 3 key.
D4	The 4 key.
D5	The 5 key.



D6	The 6 key.
D7	The 7 key.
D8	The 8 key.
D9	The 9 key.
Decimal	The decimal key.
Delete	The DEL key.
Divide	The divide key.
Down	The DOWN ARROW key.
E	The E key.
End	The END key.
Enter	The ENTER key.
EraseEof	The ERASE EOF key.
Escape	The ESC key.
Execute	The EXECUTE key.
Exsel	The EXSEL key.
F	The F key.
F1	The F1 key.
F10	The F10 key.
F11	The F11 key.
F12	The F12 key.
F13	The F13 key.
F14	The F14 key.
F15	The F15 key.
F16	The F16 key.
F17	The F17 key.
F18	The F18 key.
F19	The F19 key.
F2	The F2 key.
F20	The F20 key.
F21	The F21 key.
F22	The F22 key.
F23	The F23 key.
F24	The F24 key.

F3	The F3 key.
F4	The F4 key.
F5	The F5 key.
F6	The F6 key.
F7	The F7 key.
F8	The F8 key.
F9	The F9 key.
FinalMode	The IME final mode key.
G	The G key.
H	The H key.
HanguelMode	The IME Hanguel mode key. (maintained for compatibility; use <b>HangulMode</b> )
HangulMode	The IME Hangul mode key.
HanjaMode	The IME Hanja mode key.
Help	The HELP key.
Home	The HOME key.
I	The I key.
IMEAccept	The IME accept key, replaces IMEAccept.
IMEAcept	The IME accept key. Obsolete, use IMEAccept instead.
IMEConvert	The IME convert key.
IMEModeChange	The IME mode change key.
IMENonconvert	The IME nonconvert key.
Insert	The INS key.
J	The J key.
JunjaMode	The IME Junja mode key.
K	The K key.
KanaMode	The IME Kana mode key.
KanjiMode	The IME Kanji mode key.
KeyCode	The bitmask to extract a key code from a key value.
L	The L key.
LaunchApplication1	The start application one key (Windows 2000 or later).
LaunchApplication2	The start application two key (Windows 2000 or later).
LaunchMail	The launch mail key (Windows 2000 or later).
LButton	The left mouse button.

LControlKey	The left CTRL key.
Left	The LEFT ARROW key.
LineFeed	The LINEFEED key.
LMenu	The left ALT key.
LShiftKey	The left SHIFT key.
LWin	The left Windows logo key (Microsoft Natural Keyboard).
M	The M key.
MButton	The middle mouse button (three-button mouse).
MediaNextTrack	The media next track key (Windows 2000 or later).
MediaPlayPause	The media play pause key (Windows 2000 or later).
MediaPreviousTrack	The media previous track key (Windows 2000 or later).
MediaStop	The media Stop key (Windows 2000 or later).
Menu	The ALT key.
Modifiers	The bitmask to extract modifiers from a key value.
Multiply	The multiply key.
N	The N key.
Next	The PAGE DOWN key.
NoName	A constant reserved for future use.
None	No key pressed.
NumLock	The NUM LOCK key.
NumPad0	The 0 key on the numeric keypad.
NumPad1	The 1 key on the numeric keypad.
NumPad2	The 2 key on the numeric keypad.
NumPad3	The 3 key on the numeric keypad.
NumPad4	The 4 key on the numeric keypad.
NumPad5	The 5 key on the numeric keypad.
NumPad6	The 6 key on the numeric keypad.
NumPad7	The 7 key on the numeric keypad.
NumPad8	The 8 key on the numeric keypad.
NumPad9	The 9 key on the numeric keypad.
O	The O key.
Oem1	The OEM 1 key.
Oem102	The OEM 102 key.

Oem2	The OEM 2 key.
Oem3	The OEM 3 key.
Oem4	The OEM 4 key.
Oem5	The OEM 5 key.
Oem6	The OEM 6 key.
Oem7	The OEM 7 key.
Oem8	The OEM 8 key.
OemBackslash	The OEM angle bracket or backslash key on the RT 102 key keyboard (Windows 2000 or later).
OemClear	The CLEAR key.
OemCloseBrackets	The OEM close bracket key on a US standard keyboard (Windows 2000 or later).
Oemcomma	The OEM comma key on any country/region keyboard (Windows 2000 or later).
OemMinus	The OEM minus key on any country/region keyboard (Windows 2000 or later).
OemOpenBrackets	The OEM open bracket key on a US standard keyboard (Windows 2000 or later).
OemPeriod	The OEM period key on any country/region keyboard (Windows 2000 or later).
OemPipe	The OEM pipe key on a US standard keyboard (Windows 2000 or later).
Oemplus	The OEM plus key on any country/region keyboard (Windows 2000 or later).
OemQuestion	The OEM question mark key on a US standard keyboard (Windows 2000 or later).
OemQuotes	The OEM singled/double quote key on a US standard keyboard (Windows 2000 or later).
OemSemicolon	The OEM Semicolon key on a US standard keyboard (Windows 2000 or later).
Oemtilde	The OEM tilde key on a US standard keyboard (Windows 2000 or later).
P	The P key.
Pa1	The PA1 key.
Packet	Used to pass Unicode characters as if they were keystrokes. The Packet key value is the low word of a 32-bit virtual-key value used for non-keyboard input methods.
PageDown	The PAGE DOWN key.
PageUp	The PAGE UP key.
Pause	The PAUSE key.
Play	The PLAY key.
Print	The PRINT key.
PrintScreen	The PRINT SCREEN key.
Prior	The PAGE UP key.
ProcessKey	The PROCESS KEY key.

Q	The Q key.
R	The R key.
RButton	The right mouse button.
RControlKey	The right CTRL key.
Return	The RETURN key.
Right	The RIGHT ARROW key.
RMenu	The right ALT key.
RShiftKey	The right SHIFT key.
RWin	The right Windows logo key (Microsoft Natural Keyboard).
S	The S key.
Scroll	The SCROLL LOCK key.
Select	The SELECT key.
SelectMedia	The select media key (Windows 2000 or later).
Separator	The separator key.
Shift	The SHIFT modifier key.
ShiftKey	The SHIFT key.
Sleep	The computer sleep key.
Snapshot	The PRINT SCREEN key.
Space	The SPACEBAR key.
Subtract	The subtract key.
T	The T key.
Tab	The TAB key.
U	The U key.
Up	The UP ARROW key.
V	The V key.
VolumeDown	The volume down key (Windows 2000 or later).
VolumeMute	The volume mute key (Windows 2000 or later).
VolumeUp	The volume up key (Windows 2000 or later).
W	The W key.
X	The X key.
XButton1	The first x mouse button (five-button mouse).
XButton2	The second x mouse button (five-button mouse).
Y	The Y key.

Z	The Z key.
Zoom	The ZOOM key.

Further technical details on how Windows treats Send Key Events can be found in the following link:  
[https://msdn.microsoft.com/en-us/library/system.windows.forms.keys\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.forms.keys(v=vs.110).aspx)

Further details on all the Surface Automation techniques available in Blue Prism are available in the Surface Automation training available on the Blue Prism Portal.