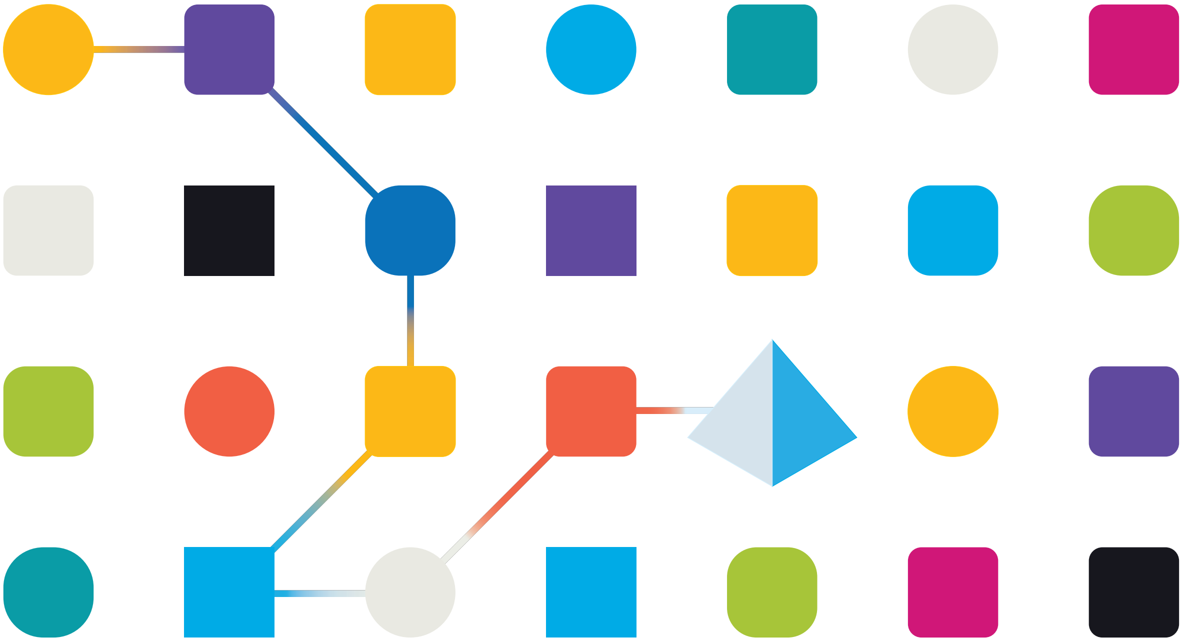
Blue Prism 7.2

Global Send Keys User Guide

Document Revision: 1.0



Trademarks and Copyright

Blue Prism Limited, 2 Cinnamon Park, Crab Lane, Warrington, WA2 0XP, United Kingdom.
Registered in England: Reg. No. 4260035. Tel: +44 370 879 3000. Web: www.blueprism.com

Contents

Global Send Keys and Global Send Key Events 1
- When to use Global Send Keys and Global Send Key Events 1
- Using Global Send Keys 2
- Using Global Send Key Events 3
- Using Global Send Keys with SAP 4

Global Send Keys and Global Send Key Events syntax 6
- Global Send Keys 6
- Global Send Key Events 8

Global Send Keys and Global Send Key Events

Global Send Keys and Global Send Key Events are text input methods for applications or elements that do not support other input methods, such as Write stages. The methods are:

- **Global Send Keys** - Global Send Keys will work for most applications and should be tried before Global Send Key Events. This is a higher-level interface that sends keystrokes to the active application. Global Send Keys uses the .NET Framework SendKeys.SendWait(String) method, either with the whole string of keys parsed from the query, or the individual keys one at a time, separated by an interval if one has been specified. It will also check for control characters if a wait interval has been specified, and will throw an exception if found.
- **Global Send Key Events** - Global Send Key Events work for all applications and is the method recommended for Citrix thin client applications. It is a lower-level interface that mimics keyboard keystrokes in the operating system. Global Send Key Events retrieve a set of virtual key codes obtained from the parameter specified in the action. The keys are then sent individually using the lower-level non-managed method keybd_event. After each keyboard event instruction is sent by Global Send Key Events, the target application thread which SS&C | Blue Prism® Enterprise uses to connect to the application will wait for the specified interval.

Global Send Keys and Global Send Key Events each use specific text formats. For more information, see Global Send Keys and Global Send Key Events syntax on page 6.

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 Send Keys to work:

- The desktop screen must exist and be persistent. Send Key interfaces will not work if the desktop screen is locked or a screen saver is displayed.
- The window that you want to send text to 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 in focus so that the keyboard cursor is in it and ready to enter text.
- To ensure the application has time to react to any window or element focus navigation, short delays are required between window activates, element clicks, and using Send Keys.
- To ensure text is entered correctly, a short delay should be configured between each keystroke. For some applications, entering text too quickly can result in some characters not being correctly entered.

You may need to implement additional logic in your processes to cater for the above factors. This will ensure that your solution will work reliably.

When to use Global Send Keys and Global Send Key Events

This functionality should be used when other methods of inputting text do not work, such as:

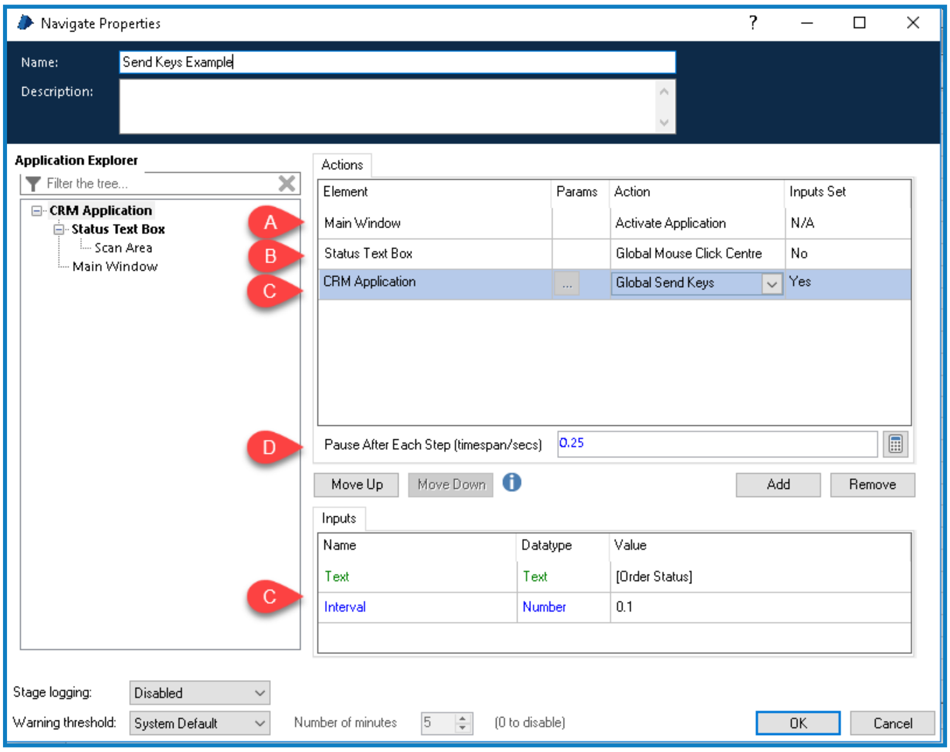
- If a thin client technology, such as Citrix, is being used.
- The element in an application cannot be identified by the Application Modeller.
- The element in an application does not accept the use of a Write stage.

Unless using Citrix thin client applications, if one interface method of sending text to an application does not work, the alternative methods 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 Windows based applications this action in a Navigate stage will work to send text. None of the window activation or element focus methods mentioned in this guide are required.
3. **Global Send Keys** - This option should be tried before Global Send Key Events as it is a higher-level interface and easier to use special keys, such as Control.
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 thin client applications. Even where Global Send Keys work, this option can be useful for some use cases where a key needs to be held down whilst other actions are performed.

Using Global Send Keys

The Global Send Keys function relies on sending keystrokes to the application area that is currently in focus. Not all parts of an application can be focused on, although Blue Prism Enterprise may be able to spy them. Global Send Keys are configured using the Navigate stage.



Navigate Properties

Name: Send Keys Example

Description:

Application Explorer

Filter the tree...

- CRM Application
  - Status Text Box
    - Scan Area
  - Main Window

Actions

Element	Params	Action	Inputs Set
Main Window		Activate Application	N/A
Status Text Box		Global Mouse Click Centre	No
CRM Application	...	Global Send Keys	Yes

Pause After Each Step (timespan/secs) 0.25

Inputs

Name	Datatype	Value
Text	Text	[Order Status]
Interval	Number	0.1

Stage logging: Disabled

Warning threshold: System Default   Number of minutes 5   (0 to disable)

The Navigate stage contains all the required actions to set up Global Send Keys:

A. The Main Window of the application is brought into focus using the **Activate Application** action.

B. The text box to send keystrokes to is made active by using a **Global Mouse Click Centre** action.

C. The **Global Send Keys** action sends the data to the application. This uses the following input parameters:

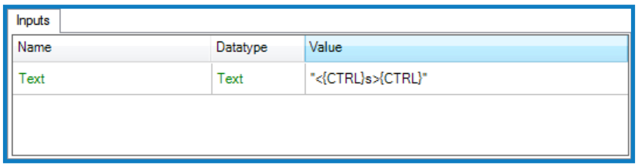
  - **Text** - The text to be input. For detail on the supported values, see Global Send Keys and Global Send Key Events syntax on page 6.
  - **Interval** - The interval between keystrokes.

D. The **Pause After Each Step** is the configured delay between each step. The appropriate value for an interface will depend upon the responsiveness of applications in your environment. This is typically between 0.25 and 1 seconds.

Using Global Send Key Events

The Global Send Key Events function sends keystroke events to the active application. This does not need to be the target application - keys are processed by whichever application is in focus. This differs from the Global Send Keys function, which sends keystrokes to the application that application modeller is currently attached to. Global Send Key Events uses a lower-level method of sending keys, and is therefore more likely to work with some applications.

Global Send Key Events are configured using the Navigate stage, with the text to be input configured using the Text parameter.



Name	Datatype	Value
Text	Text	"<{CTRL}s>{CTRL}"

The above image demonstrates pressing a CTRL+S key combination. The order of key presses is:

1. The key down signal (hold the key down)
2. The code for the Control key
3. The 's' key
4. The key up signal (release the key)
5. The Control key code as the key to be released

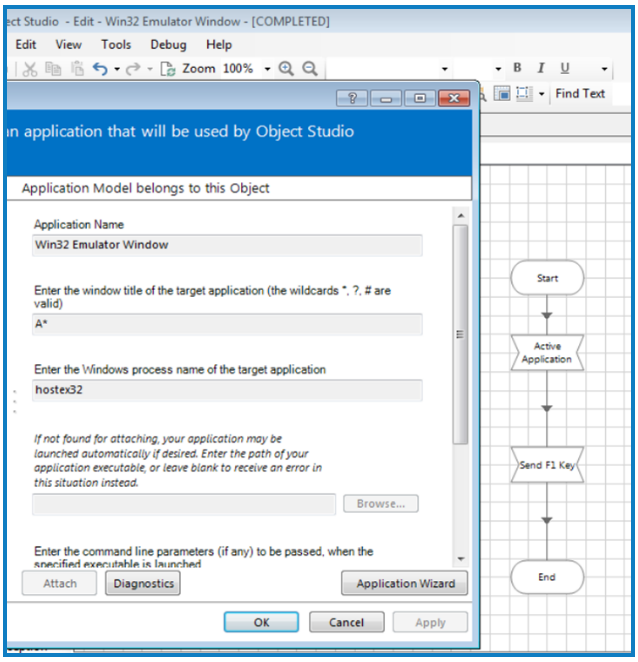
For details on the supported values and examples, see Global Send Keys and Global Send Key Events syntax on page 6.

A down keystroke for special keys must always be followed by a corresponding up keystroke.

Focusing applications

The application must have the current focus in order for Global Send Key Events to work, otherwise nothing is sent. To retain the focus when keys are being sent, it may be necessary to launch the window and then immediately send the keys as the next step.

A new Business Object may need to be created for connecting to the target application's main window, particularly if working with applications such as mainframe emulators. If you are trying to send keys to a mainframe session hosted within a mainframe emulator product, you will need a Win32 object to activate the emulator window and then send keys to that window, rather than to the underlying mainframe session.



in application that will be used by Object Studio

Application Model belongs to this Object

Application Name: Win32 Emulator Window

Enter the window title of the target application (the wildcards *, ?, # are valid): A*

Enter the Windows process name of the target application: hostex32

If not found for attaching, your application may be launched automatically if desired. Enter the path of your application executable, or leave blank to receive an error in this situation instead.

Enter the command line parameters (if any) to be passed, when the specified executable is launched

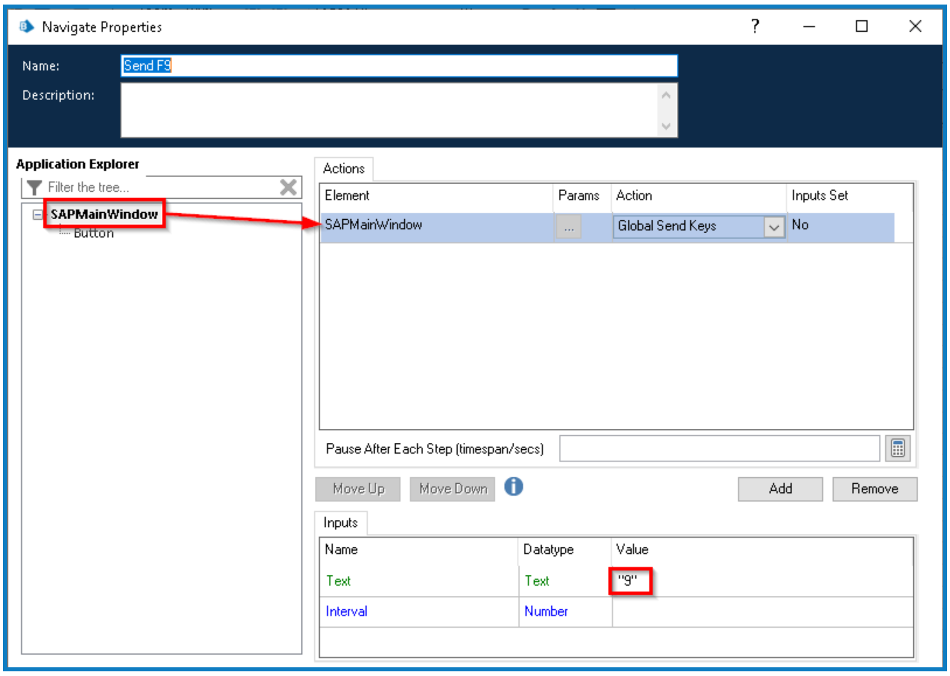
Using Global Send Keys with SAP

SAP must be configured correctly in order for Blue Prism to automate it. Please refer to our SAP Integration Guide on the Blue Prism University for more information about this configuration.

Assuming SAP has been configured correctly, when using Global Send Keys for a SAPMainWindow in SAP version 7 or above, the text has to be an integer only value.

For a complete list of virtual keys that can be used with SAP, see the SAP website.

The list of keyboard shortcuts listed on the SAP website are only called when you send a SAPMainWindow element the corresponding integer value. An example of this is when sending F9 (which has the virtual key V9 as per the SAP documentation), the integer value that needs to be input into the Value field of the Navigate Stage is "9".



Name: Send F9

Description:

Element	Params	Action	Inputs Set
SAPMainWindow	...	Global Send Keys	No

Pause After Each Step (timespan/secs)

Name	Datatype	Value
Text	Text	"9"
Interval	Number	

Global Send Keys and Global 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 keys.

Global Send Keys

Global Send Keys communicate directly to the application that an object is connected to. Key points include:

- This method uses the following characters to indicate the Shift, Control and Alt keys:
  - Shift - +
  - Control - ^
  - Alt - %
- Special keys, such as the Home key, must always be enclosed in braces. For example, {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. For example, to specify to hold down Shift while E and C are pressed, use "+(EC)". Additionally, to instruct the application to hold down Shift while E is pressed followed by C without Shift, use "+EC".
- Global 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. For example, "{LEFT 10}" means the Left arrow key will be pressed 10 times.
- A Global Send Keys action to send the Control and 'F' key combination (for example, to open a Find window) could be specified as: '^+f'.

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}

Key	Code
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 Microsoft documentation.

Double quotes

If the text contains double quotation marks, you will need to specify the ASCII code for double quotation marks within a Chr() function. For example, Chr(34). If not, the text will be interpreted as multiple strings of text separated by functions. As this is an incorrect expression, it will cause errors when the process is run.

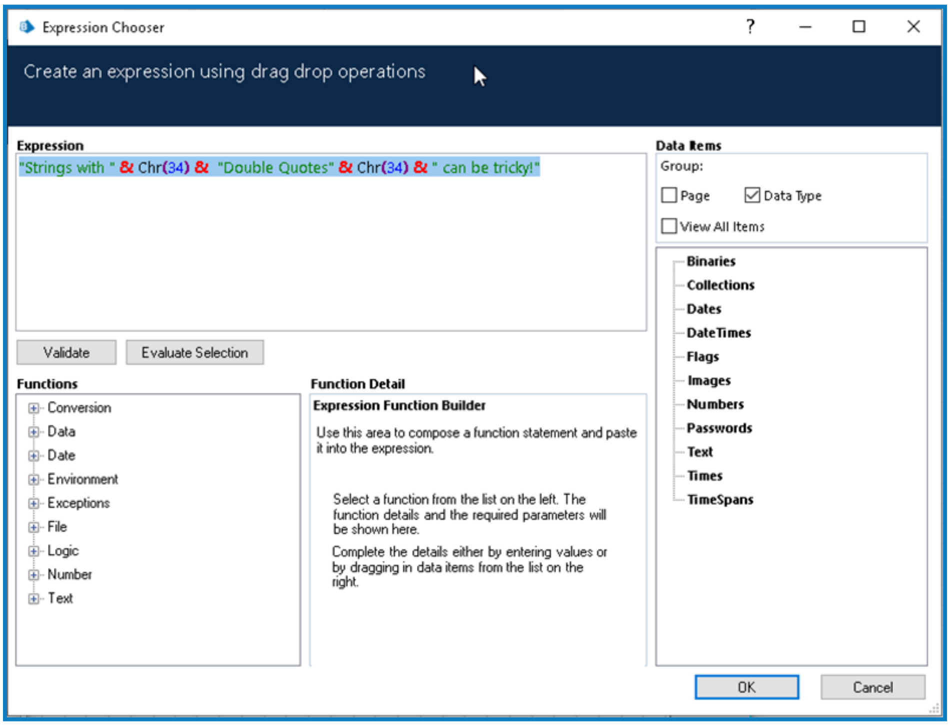
Where you have used the Chr(34) value, you must enter an ampersand (&) to join the text together so the string is read as one value. The table below shows how double quotation marks, the Chr(34) value, and ampersands function:

Expression	Output
Chr(34)	"
"hello"	hello
"hello" & "world"	hello world
"hello" & Chr(34)	hello "
“hello” & Chr(34) & “world”	hello " world

Example

Original text: “Strings with "Double Quotes" can be tricky!”

Formatted text: "Strings with "&Chr(34) & "Double Quotes"& Chr(34) &" can be tricky!"



"Strings with " & Chr(34) & "Double Quotes" & Chr(34) & " can be tricky!"

Use this area to compose a function statement and paste it into the expression.

Select a function from the list on the left. The function details and the required parameters will be shown here.

Complete the details either by entering values or by dragging in data items from the list on the right.

Global Send Key Events

Global 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 that has the active focus. This method is often useful when working with virtual desktops like Citrix, where standard application spying is not available. Key points:

- This method uses the following codes for the Shift, Control and Alt keys:
  - Shift - {SHIFT}
  - Control - {CTRL}
  - Alt - {ALT}

- To tell Global Send Key Events to press either the Shift, Control, or Alt keys, they must be preceded with the > character. Similarly, releasing a key must be preceded with the character <. For example, to send Ctrl+c , the following sequence would be used: “<{CTRL}c>{CTRL}”. This tells the system to press the Control key plus the C key, and then release Control.
- The "<" and ">" characters are used to modify the NEXT key to be just a key down < or key up > event respectively.
- Global Send Keys Events 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. For example, {LEFT 10} means the Left arrow key will be pressed 10 times.
- Since "{", "}", "<" and ">" are special characters, they must be enclosed in curly brackets. For example: {{} {}, {<} and {>}.

Send Key Events codes

Key	Code
The A key	A
The add key	Add
The ALT modifier key	Alt
The application key (Microsoft Natural Keyboard)	Apps
The ATTN key	Attn
The B key	B
The BACKSPACE key	Back
The browser back key (Windows 2000 or later)	BrowserBack
The browser favorites key (Windows 2000 or later)	BrowserFavorites
The browser forward key (Windows 2000 or later)	BrowserForward
The browser home key (Windows 2000 or later)	BrowserHome
The browser refresh key (Windows 2000 or later)	BrowserRefresh
The browser search key (Windows 2000 or later)	BrowserSearch
The browser stop key (Windows 2000 or later)	BrowserStop
The C key	C
The CANCEL key	Cancel
The CAPS LOCK key	Capital
The CAPS LOCK key	CapsLock
The CLEAR key	Clear
The CTRL modifier key	Control
The CTRL key	ControlKey
The CRSEL key	Crsel
The D key	D
The 0 key	D0

Key	Code
The 1 key	D1
The 2 key	D2
The 3 key	D3
The 4 key	D4
The 5 key	D5
The 6 key	D6
The 7 key	D7
The 8 key	D8
The 9 key	D9
The decimal key	Decimal
The DEL key	Delete
The divide key	Divide
The DOWN ARROW key	Down
The E key	E
The END key	End
The ENTER key	Enter
The ERASE EOF key	EraseEof
The ESC key	Escape
The EXECUTE key	Execute
The EXSEL key	Exsel
The F key	F
The F1 key	F1
The F10 key	F10
The F11 key	F11
The F12 key	F12
The F13 key	F13
The F14 key	F14
The F15 key	F15
The F16 key	F16
The F17 key	F17
The F18 key	F18
The F19 key	F19
The F2 key	F2
The F20 key	F20

Key	Code
The F21 key	F21
The F22 key	F22
The F23 key	F23
The F24 key	F24
The F3 key	F3
The F4 key	F4
The F5 key	F5
The F6 key	F6
The F7 key	F7
The F8 key	F8
The F9 key	F9
The IME final mode key	FinalMode
The G key	G
The H key	H
The IME Hanguel mode key (maintained for compatibility; use HangulMode)	HanguelMode
The IME Hangul mode key	HangulMode
The IME Hanja mode key	HanjaMode
The HELP key	Help
The HOME key	Home
The I key	I
The IME accept key, replaces IMEAceept	IMEAccept
The IME accept key. Obsolete, use IMEAccept instead	IMEAceept
The IME convert key	IMEConvert
The IME mode change key	IMEModeChange
The IME nonconvert key	IMENonconvert
The INS key	Insert
The J key	J
The IME Junja mode key	JunjaMode
The K key	K
The IME Kana mode key	KanaMode
The IME Kanji mode key	KanjiMode
The bitmask to extract a key code from a key value	KeyCode
The L key	L
The start application one key (Windows 2000 or later)	LaunchApplication1

Key	Code
The start application two key (Windows 2000 or later)	LaunchApplication2
The launch mail key (Windows 2000 or later)	LaunchMail
The left mouse button	LButton
The left CTRL key	LControlKey
The LEFT ARROW key	Left
The LINEFEED key	LineFeed
The left ALT key	LMenu
The left SHIFT key	LShiftKey
The left Windows logo key (Microsoft Natural Keyboard)	LWin
The M key	M
The middle mouse button (three-button mouse)	MButton
The media next track key (Windows 2000 or later)	MediaNextTrack
The media play pause key (Windows 2000 or later)	MediaPlayPause
The media previous track key (Windows 2000 or later)	MediaPreviousTrack
The media Stop key (Windows 2000 or later)	MediaStop
The ALT key	Menu
The bitmask to extract modifiers from a key value	Modifiers
The multiply key	Multiply
The N key	N
The PAGE DOWN key	Next
A constant reserved for future use	NoName
No key pressed	None
The NUM LOCK key	NumLock
The 0 key on the numeric keypad	NumPad0
The 1 key on the numeric keypad	NumPad1
The 2 key on the numeric keypad	NumPad2
The 3 key on the numeric keypad	NumPad3
The 4 key on the numeric keypad	NumPad4
The 5 key on the numeric keypad	NumPad5
The 6 key on the numeric keypad	NumPad6
The 7 key on the numeric keypad	NumPad7
The 8 key on the numeric keypad	NumPad8
The 9 key on the numeric keypad	NumPad9
The O key	O

Key	Code
The OEM 1 key	Oem1
The OEM 102 key	Oem102
The OEM 2 key	Oem2
The OEM 3 key	Oem3
The OEM 4 key	Oem4
The OEM 5 key	Oem5
The OEM 6 key	Oem6
The OEM 7 key	Oem7
The OEM 8 key	Oem8
The OEM angle bracket or backslash key on the RT 102 key keyboard (Windows 2000 or later)	OemBackslash
The CLEAR key	OemClear
The OEM close bracket key on a US standard keyboard (Windows 2000 or later)	OemCloseBrackets
The OEM comma key on any country/region keyboard (Windows 2000 or later)	Oemcomma
The OEM minus key on any country/region keyboard (Windows 2000 or later)	OemMinus
The OEM open bracket key on a US standard keyboard (Windows 2000 or later)	OemOpenBrackets
The OEM period key on any country/region keyboard (Windows 2000 or later)	OemPeriod
The OEM pipe key on a US standard keyboard (Windows 2000 or later)	OemPipe
The OEM plus key on any country/region keyboard (Windows 2000 or later)	Oemplus
The OEM question mark key on a US standard keyboard (Windows 2000 or later)	OemQuestion
The OEM singled/double quote key on a US standard keyboard (Windows 2000 or later)	OemQuotes
The OEM Semicolon key on a US standard keyboard (Windows 2000 or later)	OemSemicolon
The OEM tilde key on a US standard keyboard (Windows 2000 or later)	Oemtilde
The P key	P
The PA1 key	Pa1
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	Packet
The PAGE DOWN key	PageDown
The PAGE UP key	PageUp

Key	Code
The PAUSE key	Pause
The PLAY key	Play
The PRINT key	Print
The PRINT SCREEN key	PrintScreen
The PAGE UP key	Prior
The PROCESS KEY key	ProcessKey
The Q key	Q
The R key	R
The right mouse button	RButton
The right CTRL key	RControlKey
The RETURN key	Return
The RIGHT ARROW key	Right
The right ALT key	RMenu
The right SHIFT key	RShiftKey
The right Windows logo key (Microsoft Natural Keyboard)	RWin
The S key	S
The SCROLL LOCK key	Scroll
The SELECT key	Select
The select media key (Windows 2000 or later)	SelectMedia
The separator key	Separator
The SHIFT modifier key	Shift
The SHIFT key	ShiftKey
The computer sleep key	Sleep
The PRINT SCREEN key	Snapshot
The SPACEBAR key	Space
The subtract key	Subtract
The T key	T
The TAB key	Tab
The U key	U
The UP ARROW key	Up
The V key	V
The volume down key (Windows 2000 or later)	VolumeDown
The volume mute key (Windows 2000 or later)	VolumeMute
The volume up key (Windows 2000 or later)	VolumeUp

Key	Code
The W key	W
The X key	X
The first x mouse button (five-button mouse)	XButton1
The second x mouse button (five-button mouse)	XButton2
The Y key	Y
The Z key	Z
The ZOOM key	Zoom

Further technical details on how Windows treats Send Key Events can be found in the Microsoft documentation.

Examples

- "hello{RETURN}" - Sends HELLO and presses return.
- "<{CTRL}A>{CTRL}" - Presses the Control key down, then presses A, then releases the Control key.
- "<{ALT}AB>{ALT}" - Presses the Alt key down, then presses A, then B, then releases the Alt key - i.e. does Alt-A, Alt-B.
- "<{ALT}A>{ALT}B" - Presses the Alt key down, then presses A, then releases the Alt key and pressed B - i.e. does Alt-A, B.
- "<{CTRL}<{SHIFT}{ESCAPE}>{SHIFT}>{CTRL}" - Presses Control and Shift, then Escape, then lets go of Control and Shift.

Sending capital letters using the Shift key

The format for sending capital letters via Global Send Key Events is shown below:

"<{SHIFT}s>{SHIFT}"

This sends a capital "S".