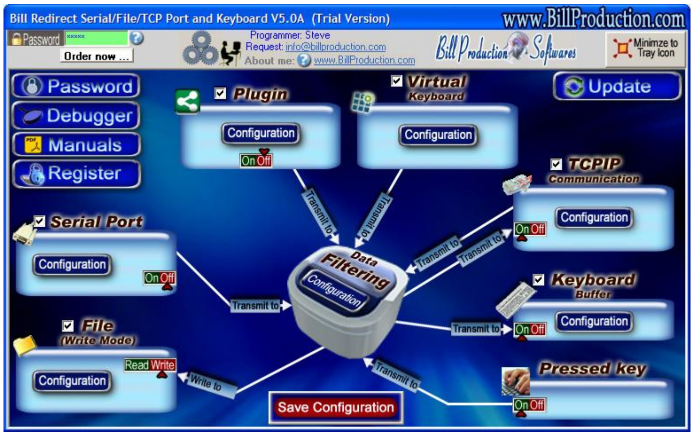
Bill Redirect
Version 8.2C and more

Documentation: 2017/04/26



Bill Redirect Serial/File/TCP Port and Keyboard V5.0A (Trial Version)

Programmer: Steve
Request: info@billproduction.com
About me: www.BillProduction.com

Use this documentation with the product : Bill Redirect

Software configuration and integration service

It's free, fast, and efficient !

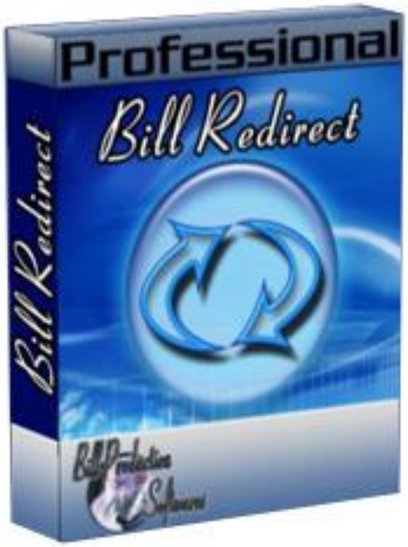
Submit your case to info@billproduction.com

and receive a customized answer !

Our Internet site: http://www.billproduction.com/

Email: info@BillProduction.com

SOFTWARE DESCRIPTION



Resident Tray System software for barcode scanner or other serial data capture devices that have a standard RS232 serial interface / file and or TCP Ethernet connection. The program will **redirect** file/serial/TCP Port and keyboard buffer to any direction. Features include: Debugger, run in System Tray, Configuration protected by PASSWORD, Upper case option, Macro Commands, Search and replace characters, can remove specific ASCII Characters, support Prefix Suffix, Widows API keyboard 32 Buffer or OCX, Can activate Target Application by Barcode and/or default. Redirects inputs (data) to any applications (POS, database, Excel, Word, or your own customized the software). Possibility to control MULTI-PORT (Serial/TCP Ethernet) !
Can send a answer to TCP and or Serial Port; perfect to control protocol !
Support AZERTY and QWERTY keyboard !

Possibility to create **Virtual Keyboards**, each buttons keys appears and disappears depending the window you activate. Each button can send data and or to Keyboard, TPC, Serial Port, file. You can **change or add Buttons on a existing application**, perfect for Touch Screen Application.

Supported Operating Systems: Windows 98,Me,2000,3000,XP,NT4,Vista, Win 8, Win 10.

Simply the best redirect and virtual keyboard software available !

GENERAL INFORMATION

To configure this software you have to enter in the field password www.billproduction.com, In the configuration options you have the possibility to change this password for what you want. To close this software, type END in the password field. To open the mouse position monitor, type MOUSE in the password field. To open the FIELD FINDER, type in the password field: FFINDER. To open the BUTTON FINDER, type in the password field: BFINDER.

To use some interfaces, for example 2 serials ports, start some instance of the software. Use this syntax: **Bill_Redirect_Serial_to_KB.exe /#INSTANCE_NAME#**

Example: Create a Shortcut instance Scanner_001

- Create a shortcut on your desktop to start BillRedirect software

Shortcut properties:

- In the first field (Target) enter:

If your Windows is 64 bits :

"C:\Program Files (x86)\BillProduction\Bill Redirect Serial COM Port to Keyboard Buffer\Bill_Redirect_Serial_to_KB.exe" /#Scanner_001#

If your Windows is 32 bits :

"C:\Program Files\BillProduction\Bill Redirect Serial COM Port to Keyboard Buffer\Bill_Redirect_Serial_to_KB.exe" /#Scanner_001#

- Start my software with this shortcut

Example: Create a Shortcut instance Scanner_002

- Create a shortcut on your desktop to start BillRedirect software

Shortcut properties:

- In the first field (Target) enter:

If your Windows is 64 bits :

"C:\Program Files (x86)\BillProduction\Bill Redirect Serial COM Port to Keyboard Buffer\Bill_Redirect_Serial_to_KB.exe" /#Scanner_002#

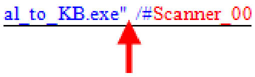
If your Windows is 32 bits :

"C:\Program Files\BillProduction\Bill Redirect Serial COM Port to Keyboard Buffer\Bill_Redirect_Serial_to_KB.exe" /#Scanner_002#

- Start my software with this shortcut

** Each instance can have a different configuration!*

** Warning: don't forget the space here:*

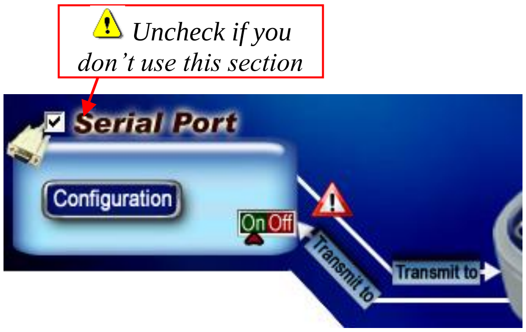


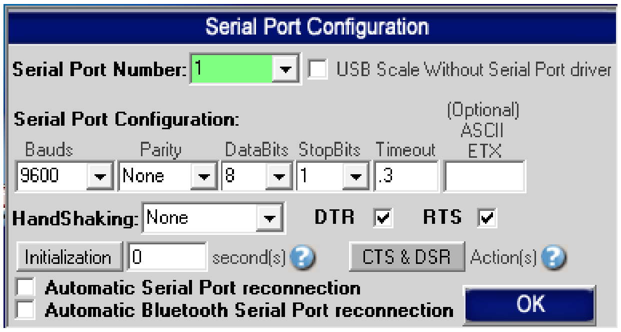
al_to_KB.exe" /#Scanner_00

** Warning: create your shortcut from the Bill_Redirect_Serial_to_KB.exe and not from the existing shortcut ... otherwise you'll have a shortcut with destination as another shortcut!*

** Don't forget to push the Save Configuration button to save your current setting.*

SERIAL PORT DESCRIPTION





Serial Port Configuration

Serial Port Number: 1 ☐ USB Scale Without Serial Port driver

Serial Port Configuration:

Bauds	Parity	DataBits	StopBits	Timeout	(Optional) ASCII ETX
9600	None	8	1	.3	

HandShaking: None DTR ☑ RTS ☑

Initialization 0 second(s) CTS & DSR Action(s)

☐ Automatic Serial Port reconnection

☐ Automatic Bluetooth Serial Port reconnection

OK

You can redirect a standard RS232 serial port data interface to any direction.

Communication through Serial Port connection is very simple with my software. The software can redirect the data receive from the Serial Port to the TCP port, File and or to the Keyboard buffer depending what switch is set to ON.
** See possible commands section*

The switch ON/OFF:

If position is ON, the data received by the peripherals: File, TCP, Keyboard and Virtual Keyboard are automatically redirect to this serial port.
If position is OFF, the data received by the peripherals are not redirect to this serial port.

The Serial Port number:

Use this field to chose the serial port number *(where your peripheral is connected)*. If the back color field is GREEN, it's because the COM is present and available.
** To disable the serial port enter: 0*

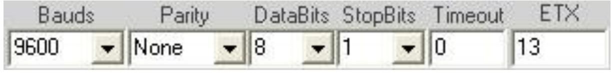
The Serial Port Configuration:

Use this section to sets the baud rate, parity, data bit, stop bit, timeout *(in seconds) and ASCII ETX* parameters.

Timeout: The timeout option is optional. Perfect to received the packet in one shot.
* Example if the timeout is .2 then the serial port wait during .2 second to send the string in one shot.

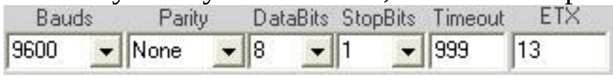
ASCII ETX: The ETX *(end of text character)* option is optional.
Perfect if your string ends always with the same character.
*ASCII characters from 0 to 255.

*For example: If your string ends with a ENTER:



Bauds	Parity	DataBits	StopBits	Timeout	ETX
9600	None	8	1	0	13

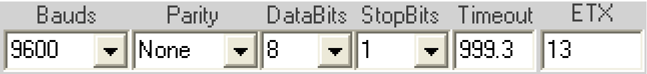
*To receive the data only if they are different, set timeout option to 999 example:



Bauds	Parity	DataBits	StopBits	Timeout	ETX
9600	None	8	1	999	13

* To receive the data only if they are different or if the **x** delay is exceed, set timeout option to **999.x**

Example **3** seconds:

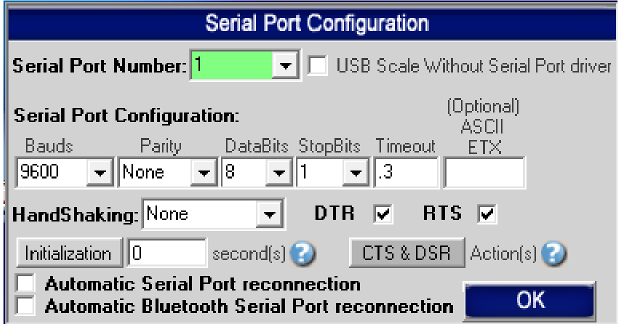


Bauds	Parity	DataBits	StopBits	Timeout	ETX
9600	None	8	1	999.3	13

* To sampled a continuous string every X seconds use this syntax 888.x Example for every 3 seconds use: 888.3

SERIAL PORT DESCRIPTION
(continuation)

Uncheck if you don't use this section


Serial Port Configuration

Serial Port Number: 1 ▾ □ USB Scale Without Serial Port driver

Serial Port Configuration:

Bauds	Parity	DataBits	StopBits	Timeout	(Optional) ASCII ETX
9600	None	8	1	.3	

HandShaking: None ▾ DTR ☑ RTS ☑

Initialization 0 second(s) ? CTS & DSR Action(s) ?

□ Automatic Serial Port reconnection

□ Automatic Bluetooth Serial Port reconnection

OK

The HandShaking:

Use this field to chose the serial port hand shaking protocol configuration.
None: No handshaking
XonXoff: Software XON/XOFF handshaking
Rts: RTS/CTS (Request to send/ Clear to send) hardware handshaking
RtsXonXoff: Both request to send and XON/XOFF handshaking

The DTR/RTS:

You can activate or deactivate the hardware line:
DTR, Enable the data terminal ready line.
RTS, Enable the request to send line.

The Initialisation button:

If the file: "C:\BillProduction.CFG\init_serial.txt" is present, then the contents of this file will be sent automatically:

- At startup.
- If you click to the button initialization.
- At interval of X second(s) (0 = disable).
  - * Enter the interval in the field beside the button 'Initialization' … perfect to poll a peripheral at fixed interval.
  - * To control the interval see the command: SERIALINIT
  - * To detect USB plug/unplug save this file http://www.billproduction.com/INIT_SERIAL.TXT to your C:\BillProduction.cfg\INIT_SERIAL.TXT and set interval to: 1 second(s)

* To send special ASCII character you can use this sequence in the file: {ASCII:XXX}
Example: Hello{ASCII:13}{ASCII:10}

The CTS & DSR button:

With the input pins CTS and DSR of your serial port you can generate action in your computer. Example: If you put a hardware switch between the pin 7 and 8 of your serial port DB9. In the configuration CTS (action push) you enter {F1}. With this configuration when you activate the hardware switch, F1 is simulated in your computer.

The Automatic Serial Port reconnection:

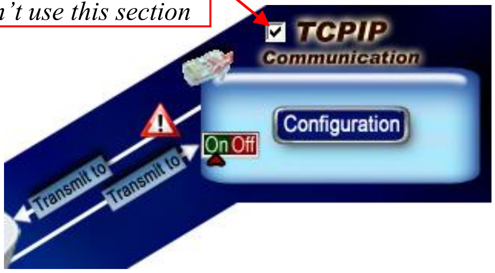
With this option the software detect automatically if the peripheral is disconnected or reconnected. Use this option with USB or Virtual Serial Com Port devices.

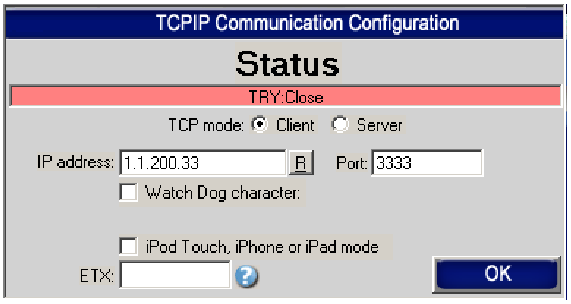
The Automatic Bluetooth Serial Port reconnection:

With this option the software detect automatically if the Bluetooth peripheral is disconnected or reconnected. Use this option with Bluetooth and USB devices.

TCP CONNECTION DESCRIPTION

Uncheck if you don’t use this section




TCPIP Communication Configuration

Status

TRY:Close

TCP mode: ● Client ○ Server

IP address: 1.1.200.33 R Port: 3333

□ Watch Dog character:

□ iPod Touch, iPhone or iPad mode

ETX:

You can redirect a standard TCP connection to any direction.

Communication through TCP connection is very simple with this software. The software can redirect the data receive from a TCP connection to the Serial port, File and or to the Keyboard buffer depending what switch is set to ON.

** See possible commands section*

The Status:

CLOSE, There are no TCP connection.
CONNECT, The connection is establish with your peripheral.
ERROR, Can’t establish the connection. *You can see this error if you place your mouse cursor to the status field*
** If the connection is lost, the software retry automatically to re-connect*

The IP address:

In this field enter the IP address of your peripheral.
* Make sure that your TCP peripheral address is in the same range that your computer IP address.

The PORT:

In this field enter the TCP PORT that use your peripheral.

The R button:

Use this button to RESET the TCP connection.

The Watch Dog character:

Use option this option to send a character after every 15 seconds to detect connection loss.

The mode Client or Server:

Client Mode, in this mode the software try to open the connection with your TCP peripheral.
Server Mode, in this mode the software switch the TCP port in listen mode and wait that your peripheral open the connection.
**Support multiple TCP connections in Server mode.*

The ETX:

In this field you can enter the optional ETX *(end of text character)* if your string ends always with the same character. *You must enter an ASCII value from 0 to 255.

The iPod Touch, iPhone or iPad mode:

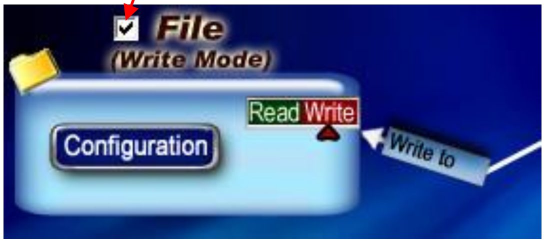
http://www.billproduction.com/iPod_iPhone_iPad_TCPIP_Remote_Control/index.html

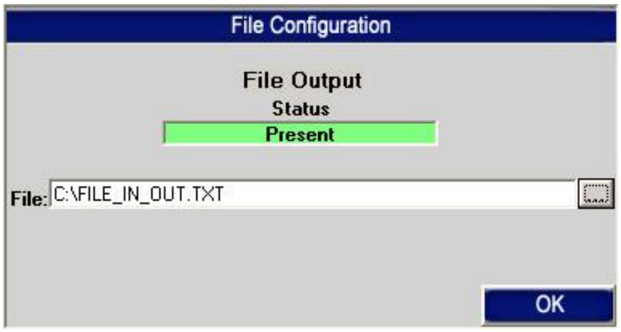
The switch ON/OFF:

If position is ON, the data received by the: File, Serial port, TCP, Keyboard and Virtual Keyboard are automatically redirect to this TCP connection.
If position is OFF, the data received by the peripherals are not redirect to the TCP connection.

FILE CONFIGURATION

Uncheck if you don't use this section





File Configuration

File Output

Status

Present

File: C:\FILE_IN_OUT.TXT

OK

You can redirect a standard FILE to any direction.

FILES COMMUNICATION

Communication through file is very simple with this software. A file can contain what you want *(see possible commands section)*. The software check the file, and when this file change the software redirect the change to the Serial port, TCP and or to the Keyboard buffer depending what is set to ON.

The Status:

Present, The file is present.
Not Found, This file is not present in this directory.
Error, There are a error in your filename and or in the directory.

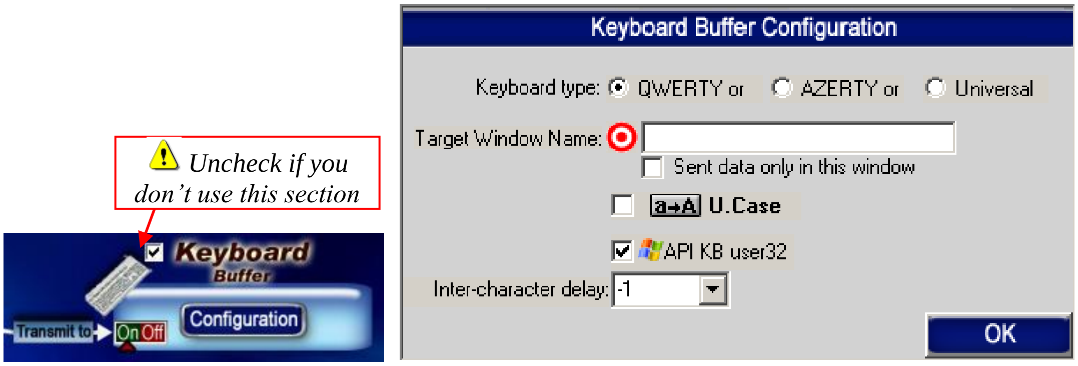
The switch Read/Write:

If position is Write, Then the file is in OUTPUT MODE, the data received by the peripherals: Serial port, TCP and Virtual Keyboard are automatically redirect to this OUTPUT FILE.

- *To receive each string in a different file, use DATETIMESEQ.TXT as a filename.*
- *To receive data in a file name with DATE and or TIME, use this syntax in File:* DATETIMEF,C:\BILLPRODUCTION.CFG\,YYYY-MM-DD ,HH-MM-SS.TXT
- *To write data in to a Excel File use for example:* C:\BILLPRODUCTION.CFG\myfile.xls

If position is Read, Then the file is in INPUT MODE, the data received by the file are redirect to the Serial port, TCP and or to the Keyboard buffer depending what is set to ON.
* If *"Delete the file after reading"* option is enable then the file will be deleted after the redirection.

KEYBOARD BUFFER CONFIGURATION



⚠ *Uncheck if you don't use this section*

Keyboard Buffer Configuration

Keyboard type: ● QWERTY or ○ AZERTY or ○ Universal

Target Window Name: ____

□ Sent data only in this window

□ a→A U.Case

☑ API KB user32

Inter-character delay: -1

OK

You can redirect the Serial port, File, TCP port and or the Virtual Keyboard to the keyboard buffer if you set the Keyboard Switch to ON.

** See possible commands section*

The QWERTY option (Default option):

Select this option if you use a QWERTY keyboard.

**If your keyboard is QWERTY you can see just across the letter Q the letters WERTY*

The AZERTY option:

Select this option if you use a AZERTY keyboard.

**If your keyboard is AZERTY you can see just across the letter A the letters ZERTY*

The Universal option:

Select this option for other keyboards or if you do not receive some special characters.

The Target Window Name:

With this option you can activate automatically a target window.
Enter the title of the window where you want redirect the data
* If you enter nothing, the software redirect the data to the current window.
* You can enter just a part of the title, for example if the title is "**my application software**" you can enter just a part of the title for example: **my application**
** Enter in field Password FFINDER to find your application window title.*

The "Sent data only in this window":

With this option no data are sent if the Target Window is not in focus.

The U. Case option:

The software can switch all lower case characters automatically to upper case.

The API KB user 32 (ON by default):

If you activate this option the data is sent directly to the Keyboard buffer this method work's with all kind of applications.
*Usually this option is set to ON.

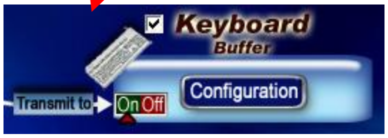
The Inter-character delay option:

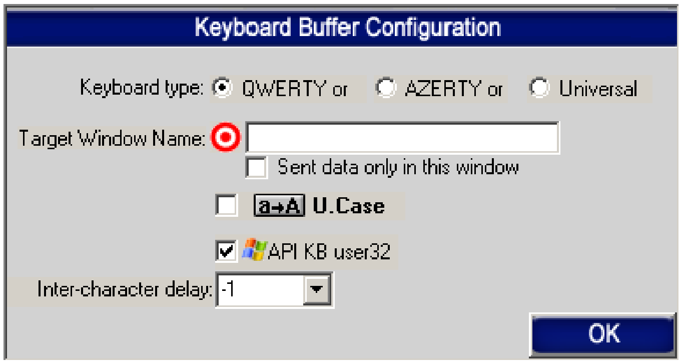
With this option you can change the output inter character delay.
*Habitually you don't have to change this option (-1 is the fastest value)

KEYBOARD BUFFER OUTPUT

(continuation)

Uncheck if you don’t use Keyboard




Keyboard Buffer Configuration

Keyboard type: ● QWERTY or ○ AZERTY or ○ Universal

Target Window Name:

□ Sent data only in this window

□ a→A U.Case

☑ API KB user32

Inter-character delay: -1

Keyboard Buffer switch ON/OFF:

> <u>If position is ON</u>, the data received by the: File, Serial port, TCP, Keyboard and Virtual Keyboard are automatically redirect to the Keyboard buffer.
>
> * *Inputs data directly into any Windows programs as if it was typed in using the keyboard.*
>
> <u>If position is OFF,</u> the data received by the peripherals are not redirect to the TCP connection.

Pressed Key switch ON/OFF:

> <u>If position is ON</u>, the Keyboard pressed keys will be redirected to: File, Serial port, TCP, Keyboard, Plugin and Virtual Keyboard

To receive two events for each key pressed (Key Pressed and Key Released)

Edit Bill Redirect Shortcut properties:

- In the first field (Target) enter:

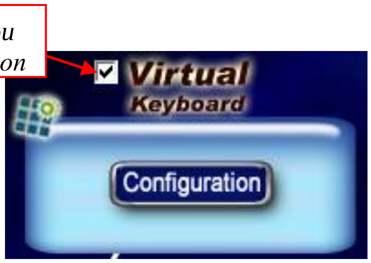
"C:\Program Files\BillProduction\Bill Redirect Serial COM Port to Keyboard Buffer\Bill_Redirect_Serial_to_KB.exe" /KP2EVENTS

- Start BillRedirect software with this shortcut

**If you don’t use the section Pressed Key set the switch to OFF*

VIRTUAL KEYBOARD

⚠ *Uncheck if you don't use this section*



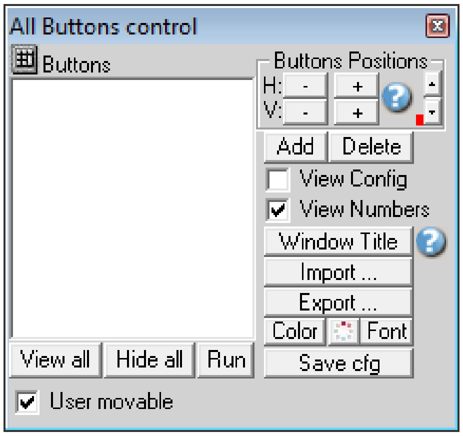
You can redirect each button of your Virtual Keyboard to any direction.

Possibility to create **Virtual Keyboards**, each buttons keys can appears and disappears depending the window you activate. Each button can send data to the Keyboard buffer, TPC connection, Serial Port and or to a file. You can **change or add Buttons on a existing application**, perfect for TouchScreen Application.

Configuration:

Use the EDIT option to open the "All Buttons control" window.

VIRTUAL KEYBOARD (All Buttons control)



Buttons list:

Just under "Buttons" you can see the existing buttons list.

View all:

Use this option to view all **Buttons list** on the screen.

Hide all:

Use this option to hide all **Buttons list** on the screen.

Run:

Use this option to try the Virtual Keyboard buttons.

User movable:

Use this if you want that the user can move them with the right button of the mouse.

Buttons Positions:

Use this section to change the position of the selected button in the **Buttons list**.
* Use the vertical scroll bars to change the moving speed.

Add:

Use this option to add a new Virtual Keyboard Button.

Del:

Use this option to delete the selected buttons in the **Buttons list**.

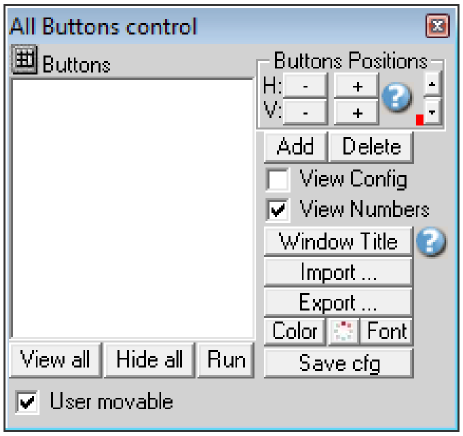
View Config:

Use this option to show or hide the configuration of the selected button in the **Buttons list**.

View Number:

Use this option to show or hide the number of the selected button in the **Buttons list**.

VIRTUAL KEYBOARD (All Buttons control) (continuation)



Window Title:

Use this option to change the "Window Title" of the selected button in the **Buttons list**.

Option "Window Title" Description: With this option each button of your Virtual Keyboard can appear and disappear when a specific window title is activate.

*Example: If the window title is "Application weight expert Version 1.2b" and in the field "Window Title" you enter "WEIGHT EXPERT" then each time that this window take the focus this button is visible.

* To enter more of one window title, separate the title name by a comas:
Example: WEIGHT EXPERT, NOTEPAD, PAGE COUNT

* If this the field **"Window Title"** is blank, then the button is always visible.
* To hide this button at startup enter : **START_HIDE**
* To show this button when you click on the desktop enter : **Program Manager**

Import:

Use this option to import a virtual keyboard.
* Some Virtual Keyboard Samples is available on www.BillProduction.com

Export:

Use this option to export to a FILE the selected button in the **Buttons list**.

Color:

Use this option to change the Color of the selected button in the **Buttons list**.

Transparency:

Use this option to change the transparency of the selected button in the **Buttons list**.
(Click on RUN to see the effect)

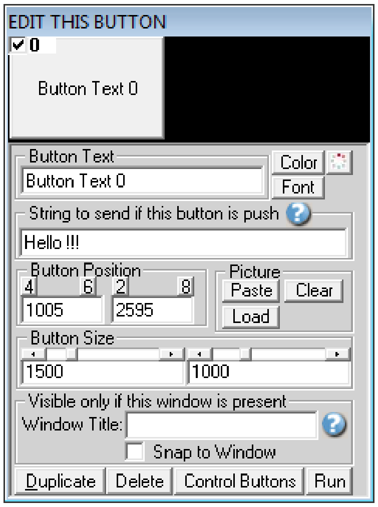
Font:

Use this option to change the Font of the selected button in the **Buttons list**.

Save cfg:

Use this option to save the current configuration.

VIRTUAL KEYBOARD (individual Buttons configuration)



Button Text: Button Text 0

String to send if this button is push: Hello !!!

Button Position: 1005 2595

Button Size: 1500 1000

Visible only if this window is present

Window Title:

□ Snap to Window

To view the configuration of each button, activate the switch box in the left top corner.

Button Text:

Use this field to set what you want to see on the button.

String to send if this button is push:

Use this field to set the string to send if the user push this button. The string is redirect to Serial port, TCP port, File and or to the Keyboard buffer depending what switch is set to ON.

To send something when the button is pressed and another thing when it is released, use the following syntax:
{*P}Pressed{*P}{*R}Released{*R}

To sent something for the first time this button is pressed and another thing when it's pressed again, use the following syntax::
{EV1}First Time{EV1}{EV2}Second Time{EV2}

** The string is send in mode **RUN** only*
The string can contain: **See possible commands section*

Button Position:

Use this section to change the position if this button.
** Tips: Activate the NumLock Click the button and use the number 4,6,2 and 8 to change the position.*

Button Size:

Use this section to change the size of the button.

Window Title:

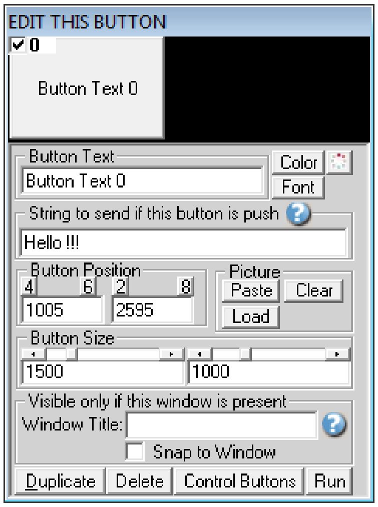
Option "Window Title" Description: With this option each button of your Virtual Keyboard can appear and disappear when a specific window title is activate.

*Example: If the window title is "Application weight expert Version 1.2b" and in the field "Window Title" you enter "WEIGHT EXPERT" then each time that this window take the focus this button is visible.

* To enter more of one window title, separate the title name by a comas:
Example: WEIGHT EXPERT, NOTEPAD, PAGE COUNT

* If this the field **"Window Title"** is blank, then the button is always visible.

VIRTUAL KEYBOARD (individual Buttons configuration) (continuation)


EDIT THIS BUTTON

☑ 0

Button Text 0

Button Text: Button Text 0

String to send if this button is push: Hello !!!

Button Position: 4 6 2 8 — 1005, 2595

Button Size: 1500, 1000

Visible only if this window is present — Window Title:

□ Snap to Window

Duplicate:

Use this option to duplicate this button.

Color:

Use this option to change the button back color.

Transparency:

Use this option to change the transparency of the button *(Click on RUN to see the effect)*.

Font:

Use this option to change the button text font.

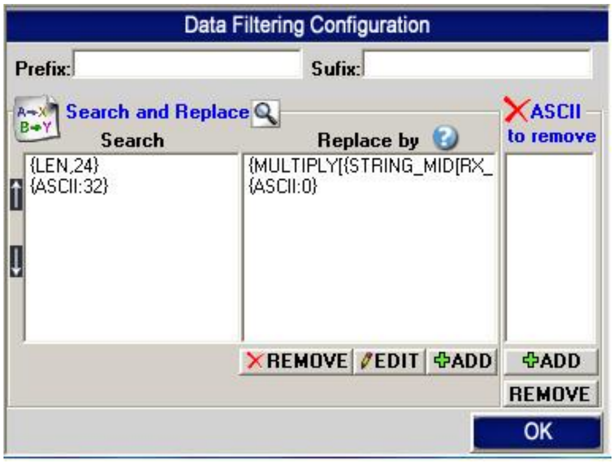
Picture:

Use this section to:
Paste the picture from the ClipBoard to the button.
Clear the picture button.
Load the picture button.

Run:

Use this option to try the Virtual Keyboard buttons.

DATA FILTERING CONFIGURATION



Search	Replace by
{LEN,24}	{MULTIPLY[{STRING_MID[RX_
{ASCII:32}	{ASCII:0}

All options in that section apply to: Serial port, File, TCP port, Keyboard and Virtual Keyboard

The Prefix/Sufix:

When the software receive data from a peripheral, it add the PREFIX at the beginning of data and the Sufix at the end.
[Prefix][data from a peripheral][Sufix]
The Prefix and the Sufix can contain any characters and or commands that you want.
**see possible commands section*

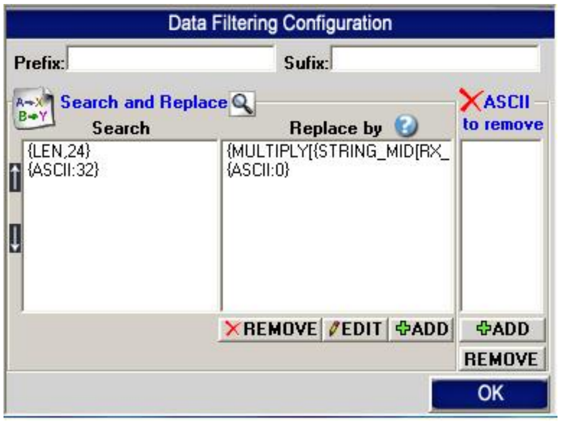
The Remove ASCII section:

With this section you can remove ASCII characters directly from the received data. With the button Add you can add a new ASCII characters from 0 to 255.
*Example: To remove Enter, ADD: 13
*Example: To remove Line Feed, ADD: 10

The Search and Replace section (up and down arrows):

Use the arrows up and down to change the order of the lines present in this section.

DATA FILTERING CONFIGURATION



Search	Replace by
{LEN,24}	{MULTIPLY[{STRING_MID[RX_
{ASCII:32}	{ASCII:0}

The Search and Replace section:

With this section you can search and replace string from any peripheral. In the field "Search" enter the search string and in the string "Replace by" enter the replacement string. The "Replace by" string can contain what you want:
**see possible commands section*

To search a string put the string
Example to search 123: 123

To search a integral string put the search sequence between "
Example to search 123: "123"

To search a string that start with something, add {START_WITH} before your search sequence:
Example to search sequences that start with 123: {START_WITH}123

To search a string that contains a specific sequence add {IN_STRING} before your search sequence:
Example to search a string that contains 123: {IN_STRING}123

To search a string with a specific length use this syntax : {LEN,XX}
*Example to search a sequence with a length of 22 : {LEN,22}

To replace all that comes from the virtual button use this syntax: {BUTTON_RX}

To replace all that comes from the file use this syntax: {FILE_RX}

To replace all that comes from the serial port use this syntax: {SERIAL_RX}

To replace all that comes from the plugin use this syntax: {PLUGIN_RX}

To replace all that comes from the Keyboard use this syntax: {KB_RX}

To replace all that comes from the TCP use this syntax: {TCP_RX}

To replace all that comes from the USB scale use this syntax: {USB_RX}

To replace all use this syntax : {ALL_RX}

To replace all that comes from the virtual button if no rules have been found use this syntax: {BUTTON_RX_ELSE}

To replace all that comes from the file if no rules have been found use this syntax: {FILE_RX_ELSE}

To replace all that comes from the serial port if no rules have been found use this syntax: {SERIAL_RX_ELSE}

To replace all that comes from the plugin if no rules have been found use this syntax: {PLUGIN_RX_ELSE}

To replace all that comes from the Keyboard if no rules have been found use this syntax: {KB_RX_ELSE}

To replace all that comes from the TCP if no rules have been found use this syntax: {TCP_RX_ELSE}

To replace all that comes from the USB if no rules have been found scale use this syntax: {USB_RX_ELSE}

To replace all if no rules have been found use this syntax : {ALL_RX_ELSE}

Search and Replace Examples:

Example 1: On all data received: From the position 2 extract 5 characters

Search	Replace by
{ALL_RX}	{STRING_MID[RX_DATA,2,5]}

Example 2: To remove the spaces

Search	Replace by
{ASCII:32}	

Example 3: To remove text: LB

Search	Replace by
LB	

Example 4: To remove carriage return line feed

Search	Replace by
{ASCII:13}	
{ASCII:10}	

Example 5: If the string contain HELLO then start the calculator

Search	Replace by
HELLO	{RUN_PRG[C:\Windows\system32\calc.exe,1]}

Example 6: To receive only the value

Search	Replace by
{ALL_RX}	{MULTIPLY[RX_DATA,1,#.##]}

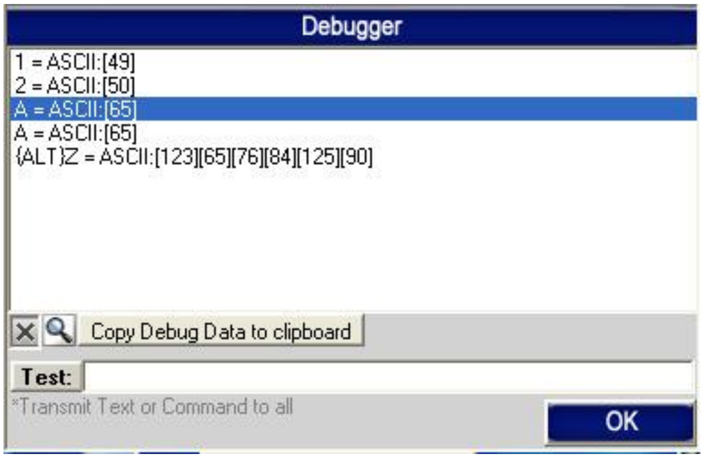
Example 7: To receive only the last 4 digits.

Search	Replace by
{ALL_RX}	{STRING_RIGHT[RX_DATA,4]}

Example 8: To multiply by 2 the number received.

Search	Replace by
{ALL_RX}	{MULTIPLY[RX_DATA,2,#]}

Debugger


Debugger
Copy Debug Data to clipboard
Test:
*Transmit Text or Command to all
OK

```
1 = ASCII:[49]
2 = ASCII:[50]
A = ASCII:[65]
A = ASCII:[65]
{ALT}Z = ASCII:[123][65][76][84][125][90]
```

The Test Button:

With this button you can send a string to the Serial port, File, TCP port and or Keyboard buffer. Use the ON/OFF switch to activate a output peripheral.
**see possible commands section*
**Sequence Example: Activate the Keyboard Buffer switch, Enter 123 in the field just across the button Test, open the NotePad click the button test, select the NotePad and wait … after 10 seconds you can see 123 appear in the NotePad.*

The Password:

To access the configuration you have to enter a password. By default the password to is www.billproduction.com but with this field you can change this password for what you want.

POSSIBLE COMMANDS SECTION

You can use the commands on any fields (Prefix, Sufix, Search&Replace…) and or from any peripheral (Serial, File, TCP, Keyboard and Virtual Keyboard)

Special ASCII character:

To use special ASCII character use this sequence:
{ASCII:XXX}
XXX = 0 to 255
Example to search ASCII 2 character: {ASCII:2}
Example to search ASCII 13 (Enter) character: {ASCII:13}

Send data to Serial Port:

You can send data directly to the Serial Port connection
{TX_SERIAL[String to Send]}
*The String to Send can contain "Special ASCII character": {ASCII:XXX}
Example: {TX_SERIAL[RX_DATA and Hello{ASCII:13}{ASCII:10}]}

Send data to a specific Serial Port:

You can send data directly to a specific Serial Port connection
{TX_SERIALX[String to Send, Port Number, Setting]}
*The String to Send can contain "Special ASCII character": {ASCII:XXX}
Example: {TX_SERIALX[RX_DATA and Hello{ASCII:13}{ASCII:10},1,9600,N,8,1]}

Send data to TCP Port:

You can send data directly to the TCP connection
{TX_TCP[String to Send]}
*The String to Send can contain "Special ASCII character": {ASCII:XXX}
Example: {TX_TCP[RX_DATA and Hello{ASCII:13}{ASCII:10}]}

Send data directly to a UDP port:

You can send data directly to a UDP port
{TX_UDP[String to Send, IP address, Port number, number of times]}
*The String to Send can contain "Special ASCII character": {ASCII:XXX}
* because the UDP protocol can't guarantee that data will be sent to the IP, you can with the option "number of times" repeatedly send the same packet.
Example of Broadcast: {TX_UDP[Hello RX_DATA,255.255.255.255,23,1]}
Example of Unicast: {TX_UDP[Hello RX_DATA,10.1.200.44,23,1]}

Send data to KEYBOARD BUFFER:

You can send data directly to the KEYBOARD BUFFER
{TX_KEYB[String to Send]}
*The String to Send can contain "Special ASCII character": {ASCII:XXX}
Example: {TX_KEYB[RX_DATA and Hello{ASCII:13}{ASCII:10}]}

Send data to a VIRTUAL KEYBOARD BUTTON:

You can change the TEXT of a virtual button with this command
{TX_BUTTON_TXT[Button_Number, Text]}
Button_Number = The number of the button *(Add the letter* S *to save the Text button Example:*S0 *)*
Text = The new text for this button
*The Text can contain "Special ASCII character": {ASCII:XXX}
*To clear the text button use EMPTY in the Text
Example without Text button saved: {TX_BUTTON_TXT[0,Hello !!!]}
Example with Text button saved: {TX_BUTTON_TXT[S0,Hello !!!]}

POSSIBLE COMMANDS SECTION (continuation)

Send data to FILE:

With this command you can send data directly to a FILE,

{TX_FILE[FILE_NAME,STRING]}

FILE_NAME: Directory and file name

STRING : Sequence to save

*The String to Send can contain "Special ASCII character": {ASCII:XXX}

Example 1: {TX_FILE[C:\Barcode.txt,Hello !!!]}

Example 2: {TX_FILE[C:\Barcode.txt,RX_DATA]}

Send data to PLUGIN:

With this command you can send data directly to PLUGIN,

{TX_PLUGIN[STRING]}

STRING : Sequence to send

*The String to Send can contain "Special ASCII character": {ASCII:XXX}

Example: {TX_PLUGIN[Hello{ASCII:9}world !!!]}

Send data to another software field:

With this command you can send data directly in a field of another software,

{TX_FIELD[AppTitle,FieldNumber,TextToWrite]}

AppTitle: Enter the title of the window where is the field.

*Example: *If the window title is "Application weight expert Version 1.2b" then you can enter just a part of the title for example: WEIGHT EXPERT*

FieldNumber: To open the FIELD FINDER, type FFINDER in BillRedirect Password field.

TextToWrite: String to Send.

Example 1: {TX_FIELD[Notepad,15,Hello !!!]}

Example 2: *Add this rules in the section "Search and Replace"*

Search	Replace by
{ALL_RX}	{TX_FIELD[Calculator,403, RX_DATA]}

Click a button in another software:

With this command you can click a button in another software,

{CLICK_BUTTON[AppTitle,ButtonNumber]}

AppTitle: Enter the title of the window where is the Button.

*Example: *If the window title is "Application weight expert Version 1.2b" then you can enter just a part of the title for example: WEIGHT EXPERT*

ButtonNumber: To open the BUTTON FINDER, type BFINDER in BillRedirect Password field.

Example: {CLICK_BUTTON[Calculator,126]}

Copy data to CLIPBOARD:

You can copy data directly to the CLIPBOARD,

{COPY_CLIP[string to copy]}

*The String to Send can contain "Special ASCII character": {ASCII:XXX}

Example: {COPY_CLIP[Hello !!!]}

Example: {COPY_CLIP[RX_DATA]}

Set DTR and RTS to ON/OFF:

Commands to control Serial port pin DTR and RTS,

{DTR_ON} {DTR_OFF} {RTS_ON} {RTS_OFF}

POSSIBLE COMMANDS SECTION (continuation)

Set DTR and RTS to ON x second(s):

Commands to control serial port pin DTR and RTS,

{DTR_PULSE[Delay]}

{RTS_PULSE[Delay]}

Delay: Delay in second

Example 1: {DTR_PULSE[2]}

Example 2: {RTS_PULSE[0.5]}

Run an executable program:

You can start a executable program

{RUN_PRG[Software_Name, Window_Style]}

The **Window_Style** has these values:

0 - Window is hidden and focus is passed to the hidden window.
1 - Window has focus and is restored to its original size and position.
2 - Window is displayed as an icon with focus.
3 - Window is maximized with focus.
4 - Window is restored to its most recent size and position. The currently active window remains active.
6 - Window is displayed as an icon. The currently active window remains active.

Example: {RUN_PRG[C:\Windows\system32\calc.exe,1]}

Mouse action:

With this command you can control the mouse

{MOUSE_ACTION[X,Y,SCR_WIN,ACTION]}

X = Horizontal position *(use 0 for the current position,also supports negative values)*

Y = Vertical position *(use 0 for the current position,also supports negative values)*

SCR_WIN = SCREEN or WINDOW or CURRENT

SCREEN: Position on the screen
WINDOW: Position on the current window
CURRENT: current position

ACTION = LEFT_CLICK, LEFT_DBL_CLICK, MIDDLE_CLICK, MIDDLE_DBL_CLICK, RIGHT_CLICK , RIGHT_DBL_CLICK or NO_ACTION

Example: {MOUSE_ACTION[100,100,SCREEN,LEFT_CLICK]}

Example: {MOUSE_ACTION[200,300,WINDOW,LEFT_DBL_CLICK]}

Example: {MOUSE_ACTION[-500,0,CURRENT,NO_ACTION]}

To open the mouse position monitor, type* **MOUSE *in the password field.*

String manipulation MID:

With this command you can extract a specific number of characters,

{STRING_MID[RX_DATA,Start,Length]}

Start = Starting position.

Length = Number of character to return (use 0 to extract all character from the Start)

Example: {STRING_MID[RX_DATA,2,0]}

Example: {STRING_MID[RX_DATA,2,3]}

Example, from character 2 extract all characters except the last 3: {STRING_MID[RX_DATA,2,-3]}

String manipulation RIGHT:

With this command you can extract from the right a specific number of characters,

{STRING_RIGHT[RX_DATA,Length]}

Length = Number of character to return

Example: {STRING_RIGHT[RX_DATA,4]}

POSSIBLE COMMANDS SECTION (continuation)

String manipulation LEN:

This command returns the length of the specified string,
{STRING_LEN[RX_DATA,AddOrSub]}
AddOrSub = You can add or subtract a value to the result.
Example: {STRING_LEN[Hello !!!,-3]}
Result is: 6
Example: {STRING_LEN[Hello !!!,+3]}
Result is: 12
Example: {STRING_LEN[Hello !!!,0]}
Result is: 9

String manipulation NOSPACE:

With this command you can remove all space separator from a string and replace it by what you want,
{STRING_NOSPACE[RX_DATA,ReplaceBy]}
ReplaceBy = String should be replaced by what spaces.
Example: {STRING_NOSPACE[1 2 3.22 4.55 2 5,*]}
Result is: 1*2*3.22*4.55*2*5

String manipulation REPLACE:

With this command you can replace a sequence of characters in a string with another set of characters.
{STRING_REPLACE[RX_DATA,Find,Replacement]}
Find = Is the string that will be searched for in RX_DATA.
** You can use **NOT:** to replace all characters not included by the Replacement.*
Replacement = Will replace Find string in RX_DATA
Example: {STRING_REPLACE[RX_DATA,2,A]}
Example: {STRING_REPLACE[123456,2,A]}
Result is: 1A3456
Example: {STRING_REPLACE[123456,**NOT:**246,A]}
Result is: A2A4A6

Conversion FROM BASE:

With this command you can convert base ten numbers to other bases,
{CNV_TOBASE[TheNumber,NewBase]}
TheNumber = Number in base 10
NewBase = New base (2 = Binary, 10 = Decimal, 16 = Hexadecimal)
Example: {CNV_TOBASE[255,02]}
Example: {CNV_TOBASE[RX_DATA,02]}

Conversion TO BASE:

With this command you can convert any base numbers to base ten,
{CNV_FROMBASE[TheNumber,OldBase]}
TheNumber = Number in any base
OldBase = The base (2 = Binary, 10 = Decimal, 16 = Hexadecimal)
Example: {CNV_FROMBASE[FF,16]}
Example: {CNV_FROMBASE[RX_DATA,10]}

Delete a FILE:

With this command you can delete a FILE,
{DEL_FILE[FILE_NAME]}
FILE_NAME: Directory and file name
Example: {DEL_FILE[C:\Barcode.txt]}

POSSIBLE COMMANDS SECTION (continuation)

Conversion BYTE to Hexadecimal:

With this command you can convert data received from Byte to Hexadecimal format,
HEX_DATA
** In other words, use HEX_DATA instead of RX_DATA*
** To receive the data in Hexadecimal inverted use HEXR_DATA*

Multiply by:

With this command you can multiply any numbers,
{MULTIPLY[TheNumber,Mutiply_by,Format]}
TheNumber = Number
Mutiply_by = Number
Format = *Optional: For example the format for 2 decimal is #.##
Example: {MULTIPLY[10,2]}
Example: {MULTIPLY[RX_DATA,10,#.##]}

Divide by:

With this command you can divide any numbers,
{DIVIDE[TheNumber,Divide_by,Format]}
TheNumber = Number
Divide_by = Number
Format = *Optional: For example the format for 2 decimal is #.##
Example: {DIVIDE[10,2]}
Example: {DIVIDE[RX_DATA,10,#.##]}

Addition:

This command return the sum of two numbers,
{SUM[TheNumber1, TheNumber2,Format]}
TheNumber1 = Number
TheNumber2 = Number
Format = *Optional: For example the format for 2 decimal is #.##
Example: {SUM[10,2]}
Example: {SUM[RX_DATA,10,#.##]}

Subtraction:

This command return the subtraction of two numbers,
{SUB[TheNumber1, TheNumber2,Format]}
TheNumber1 = Number
TheNumber2 = Number
Format = *Optional: For example the format for 2 decimal is #.##
* Example: {SUB[10,2]}
* Example: {SUB[RX_DATA,10,#.##]}

POSSIBLE COMMANDS SECTION (continuation)

Hide Virtual Button(s):

With this command you can make virtual button(s) invisible(s),
{BUTTON_HIDE[XX,XX,XX,...]}
XX = The number of the button *(ALL = All virtual buttons)*
Example to make virtual buttons 1 to 3 invisibles: {BUTTON_HIDE[1,2,3]}
Example to make all virtual buttons invisibles: {BUTTON_HIDE[ALL]}
* To control buttons «In a Window» the syntax is: {WBUTTON_HIDE[XX,XX,XX,...]}

* To make a virtual button invisible at startup: In the configuration of the button in the field «Window Title» enter : **START_HIDE**

Show Virtual Button(s):

With this command you can make virtual button(s) visible(s),
{BUTTON_SHOW[XX,XX,XX,...]}
XX = The number of the button *(ALL = All virtual buttons)*
Example to make virtual buttons 1 to 3 visible: {BUTTON_SHOW[1,2,3]}
Example to make all virtual buttons visible: {BUTTON_SHOW[ALL]}
* To control buttons «In a Window» the syntax is: {WBUTTON_SHOW[XX,XX,XX,...]}

Change the color of Virtual Button(s):

With this command you can change the color of virtual button(s) ,
{BUTTON_COLOR[COLORNO,XX,XX,...]}
COLORNO = Color number
XX = The number of the button *(ALL = All virtual buttons)*

* *To find the value of a color: In a virtual button choose a color with the button* Color ... *after this operation, move the mouse cursor to the button* Color ... *and wait… the value will appear.*
Example to change the color of buttons 1 2 and 3 in red: {BUTTON_SHOW[255,1,2,3]}
Example to change the color of all buttons in red: {BUTTON_COLOR[255,ALL]}
* To control buttons «In a Window» the syntax is: {WBUTTON_COLOR[COLORNO,XX,XX,...]}

Load a configuration button « In a Window » section:

With this command you can load a configuration of buttons ,
{WBUTTON_LOAD[CFG_NAME]}
CFG_NAME = Name of the configuration to be loaded.
Example to load the configuration 12345: {WBUTTON_LOAD[12345]}
* If the configuration does not exist then the configuration named ERROR will be loaded.

String manipulation RWPAR:

With this command you can extract character from a string using delimiters,
{STRING_RWPAR[RX_DATA,Delimiter,Position]}
Delimiter = Delimeter ASCII characters value from 0 to 255 *(Example use 9 for TAB)*
Position = Parameter number to extract
Example to extract parameter #1: {STRING_RWPAR[RX_DATA,9,1]}

Set the variable LST_TXCMD:

With this command you can set the variable LST_TXCMD (Last command sent),
{TX_LSTCMD[STRING]}
STRING = Sequence to save in variable LST_TXCMD
Example to save HELLO in to LST_TXCMD: {TX_LSTCMD[HELLO]}
Example to clear the variable LST_TXCMD: {TX_LSTCMD[]}
* By default the variable LST_TXCMD contain the last command sent by the software
* You can use the variable LST_TXCMD at any place that you want.

POSSIBLE COMMANDS SECTION (continuation)

Proximity RFID AWID readers data conversion:

With this command you can convert AWID data to card number,
{CNV_AWID[TheNumber,Facility]}
Facility = 0 for disable or 1 to enable
TheNumber = Number
Example to receive the facility code and the card number:
Add this rules in the section "Search and Replace"

Search	Replace by
{ALL_RX}	{CNV_AWID[RX_DATA,1]}

Bioscrypt finger readers data conversion:

With this command you can convert Bioscrypt Finger data to Template ID number,
{CNV_BIOSCRYPT[TheNumber]}
TheNumber = Number to convert
Example: To convert a Bioscrypt Finger to "Template ID number":

Add this rules in the section "Search and Replace"

Search	Replace by
{START_WITH}{ASCII:64}	{CNV_BIOSCRYPT[RX_DATA]}{ENTER}

BluePrint Symcod finger readers data conversion:

With this command you can convert BluePrint Finger data to Template ID number,
{CNV_BLUEPRINT[TheNumber]}
TheNumber = Number to convert
Example: To convert a BluePrint Finger to "Template ID number":

Add this rules in the section "Search and Replace"

Search	Replace by
{START_WITH}{ASCII:64}	{CNV_BLUEPRINT[RX_DATA]}{ENTER}

Set the serial Port Number:

With this command you can change the serial port number.
** Perfect to enable and disable the serial port via a virtual button.*
{SERIALPORTNUMBER[TheNumber]}
TheNumber = Port number
Example to use the serial port number 2: {SERIALPORTNUMBER[2]}
Example to disable the serial port: {SERIALPORTNUMBER[0]}

Displays an input box to ask the user to enter a string:

With this command you can ask a question and use the answer.
{INPUT_BOX[Prompt,Title,X,Y,Default]}
Prompt = The question
Title = Window title
X = Horizontal position (optional)
Y = Vertical position (optional)
Default = String expression displayed in the text box as the default (optional)
Example to ask a username: {INPUT_BOX[Your name,Identification]}
**To open the mouse position monitor, type MOUSE in the password field.*
**To use the answer, use {ANSWER01} for question 1 and {ANSWER02} for question 2 …*

POSSIBLE COMMANDS SECTION (continuation)

Change Prefix:

With this command you can change the Prefix,
{PREFIX[String]}
Example: {PREFIX[Hello !!!]}

Change Sufix:

With this command you can change the Sufix,
{SUFIX[String]}
Example: {SUFIX[Hello !!!]}

Activate a target window:

With this option you can activate a target window application,
{APP[Window_Title_Name]}
Window_Title_Name = Title of the window you activate
* You can enter just a part of the title, for example if the title is “my application software” you can enter just a part of the title for example: my application
Example: {APP[my application]}

Change serial port Timeout:

With this command you can change the serial port “Timeout” option,
{SERIAL_TIMEOUT[Value]}
Example: {SERIAL_TIMEOUT[999.03]}
Example: {SERIAL_TIMEOUT[0.03]}
** For more information about timeout Value see section “SERIAL PORT DESCRIPTION”*

Change serial port Timeout:

With this command you can change the serial port “Timeout” option,
{SERIAL_TIMEOUT[Value]}
Example: {SERIAL_TIMEOUT[999.03]}
Example: {SERIAL_TIMEOUT[0.03]}
** For more information about timeout Value see section “SERIAL PORT DESCRIPTION”*

Change serial port Initialisation interval (Polling):

With this command you can change the serial port “Initialisation xx second(s) ”,
{SERIALINIT[Seconds,NbSent]}
Seconds = Interval between each sending.
NbSent = Number of times the file C:\BillProduction.CFG\init_serial.txt will be sent.
Use 9999 to send the sequence to the infinite.
Use 1 for one time or to stop polling.
Note: You can use this syntax to use/show the current count value: {SER_NB_INIT}
Note: At the end of acquisition an event {SER_NB_END} will be generated.
Example: {SERIALINIT[5,50]}
Example: {SERIALINIT[0.2,10]}

POSSIBLE COMMANDS SECTION (continuation)

RFID TAG ISO 11784 / 11785 HITAG ALLFLEX data conversion:

With this command you can convert animal TAG identification number from binary or hexadecimal to Canada format (number indicated on the TAG)

*** Tested with low cost reader / scanner: Cybortra and Sumlung Technology.

{CNV_ISO_11784_11785_QC[NumberHEXorBIN]}

NumberHEXorBIN = Number received from the scanner (Bin or Hex)

Example to receive the number indicated on the TAG from a Binary scanner:

Add this rules in the section "Search and Replace"

Search	Replace by
{ALL_RX}	{CNV_ISO_11784_11785_QC[RX_DATA]}

Set a variable with SET_VAR:

With this command you can set a variable

{SET_VAR[NO,STRING]}

NO = Variable number from 1 to 50

STRING = Sequence to save in variable {GET_VAR[NO]}

Example to save HELLO in variable #1: {SET_VAR[1,HELLO]}

Example to clear the variable #1: {SET_VAR[1,]}

* You can use the variable {GET_VAR[NO]} at any place that you want.

Define a file header for FILE section and command TX_FILE:

With this command you define a FILE header

{FILE_HEADER[STRING]}

STRING = The STRING used for the header

Example for a Excel file:

{FILE_HEADER[Date{ASCII:9}Time{ASCII:9}Weight{ASCII:13}{ASCII:10}]}RX_DATA

POSSIBLE COMMANDS SECTION (continuation)

Keyboard buffer output, possible values

Each key is represented by one or more characters. To specify a single keyboard character, use the character itself. To represent the letters A, B, and C, use ABC for string. To specify characters that aren't displayed when you press a key, such as ENTER or TAB, and keys that represent actions rather than characters, use the codes shown below:

Description	**Command**
SEND TO WINDOWS DESKTOP	{WinDesk}
SEND TO CURRENT WINDOWS	{WinCur}
START MENU WINDOWS	{StartWin}
BACKSPACE	{BACKSPACE}, {BS}, or {BKSP}
BREAK	{BREAK}
CAPS LOCK	{CAPSLOCK}
DEL or DELETE	{DELETE} or {DEL}
Comma	{COMMA}
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}
Current prefix	{CURPREFIX}
Current suffix	{CURSUFIX}
Current System Time	{CURTIME}
Current System Time with millisecond	{CURTIMEMS}
Current System Date	{CURDATE:DD-MM-YYYY}
Get clipboard text	{CLIPBOARD}
XX Button text (XX = The virtual button number)	{BUTTON_TXT[XX]}
XX Button text (XX = The “In a Window” button number)	{WBUTTON_TXT[XX]}
F1 to F16	{F1} to {F16}
{SHIFT_DOWN}	Key Shift press
{SHIFT_UP}	Key Shift release
{CTRL_DOWN}	Key Ctrl press
{CTRL_UP}	Key Ctrl release
{ALT_DOWN}	Key Alt press
{ALT_UP}	Key Alt release
Data received (example the Barcode)	RX_DATA
Data received in Hexadecimal	HEX_DATA or HEXR_DATA for inverted
Last command sent by the software	LST_TXCMD
In TCP SERVER mode the IP address	{TCP_ADR}
{PAUSE.01SEC}	Do a pause of .01 second
{PAUSE.1SEC}	Do a pause of .1 second
{PAUSE.5SEC}	Do a pause of .5 second
{PAUSE1SEC}	Do a pause of 1 second
{PAUSE2SEC}	Do a pause of 2 seconds

To specify keys combined with any combination of the SHIFT, CTRL, and ALT keys, precede the key code with one or more of the following codes:

{SHIFT}	{CTRL}	{ALT}

Example, to send ALT and A use this syntax: {ALT}A

PERSONALIZE

**To open the mouse position monitor, type MOUSE in the field password and move the main window in the left top corner.*

How change the default configuration directory:
**By default the configuration of my software is in the directory: C:\BillProduction.CFG\*
To change this directory:

- In the same directory that my .exe application or in your C:\ modify the file: **BPCFGDIR.TXT**
  ** This file contain the default configuration directory.*
  ** Several BillRedirect softwares in differents computers, can use the same directory over the network. If the configuration changed in a computer, then other computers automatically synchronize in the next 5 seconds.*

To hide the software interface: Create the following file: C:\BillProduction.cfg\hideprg.txt

All the configuration is by default in C:\BillProduction.cfg\*.* then to duplicate the configuration from a computer to another, just copy this directory.

** Don't forget to close Bill Redirect on target computer before copy the configuration.*

Step by step how rebuild your own installation package

- *Unzip http://www.billproduction.com/Bill_COMtoKB.ZIP in to your C:\BillRPackage\*
- *Copy your default configuration from your C:\BillProduction.cfg\*.* to C:\BillRPackage\*
- *Rename the file **Setup.exe** to **Z.EXE***
- *Create the file **Setup.bat** ... this file contain:*

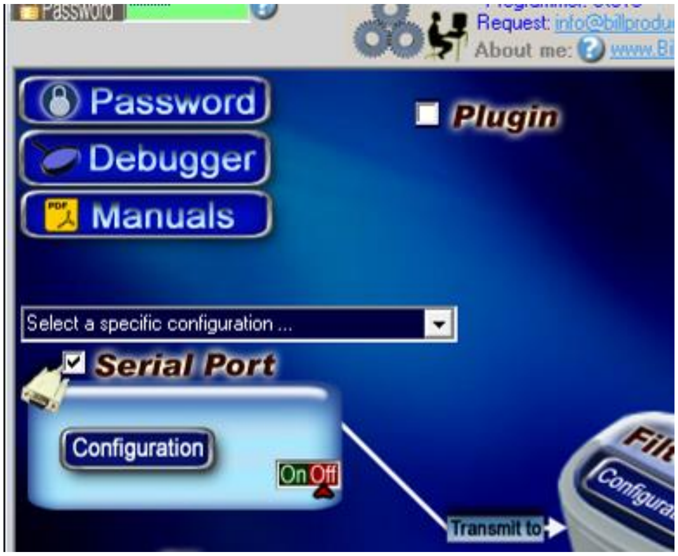
```
@echo off
md c:\BillProduction.CFG
copy *.cfg c:\BillProduction.CFG
copy skin*.* c:\BillProduction.CFG
copy *.TAG c:\BillProduction.CFG
copy *.BMP c:\BillProduction.CFG
copy *.TXT  c:\BillProduction.CFG
Z.EXE
```

- *Zip all files in C:\BillRPackage\ to My_Package.ZIP*

**** Your package is ready !*

The pull-down multi-configuration

Included in this version: The ability to activate a pull-down list that allows the user to select the configuration they want.



To activate the switch and create a new configuration:

- First, configure the software Bill Redirect as you want.
- In the directory C:\BillProduction.CFG\ rename all files ending with .CFG by adding: CFGB_CONFIGNAME_
- Example the file GENERAL.CFG become CFGB_CONFIGNAME_GENERAL.CFG

** In the preceding syntax, replace CONFIGNAME by the name you want for this configuration.*
** Note, the configuration name does not contain the caracter « _ »*

You must do the same thing with every file. CFG in this

To create another configuration, please do the same steps but with a different CONFIGNAME

For more flexibility/options or an installation package build for your company:
info@BillProduction.com