MButtton

COLLABORATORS

	TITLE : MButton		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		July 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

MButton

1.1 main

```
                MButton V36.8

            © 1994,95 TVI Interactive,  by Robert Hardy

            UUCP: robert@tvinet.com
                or
            Fido: 1:153/250.0


i) Contents

        1)  Legal Stuff
        2)  System Description
        3)  Startup
        4)  Rexx Port
        5)  Installation
```

1.2 legal_stuff

```
1) Legal Stuff

1.1) Disclaimer

    MButton is provided "AS IS", WITHOUT ANY WARRANTY to its
    quality, performance or fitness for a particular purpose.  In
    no event shall the author be liable or responsible to the user
    or any other person, for any kind of damage caused by the use
    of this software.

1.2) Conditions of Use and Distribution

    MButton is MessageWare, if you like and use it, send me a
    message (Internet or Fido).  Feel free to include complaints or
    suggestions.
```

```
  MButton may be freely distributed provided:

       1. The files are ALL left intact and unmodified.

       2. No charge is made for MButton (other than a reasonable copy
          charge)

       3. MButton is not packaged as a part of a commercial product
          without prior written consent.

       4. MButton may not be uploaded to any electronic service that
          claims a copyright to it's files and programs.

1.3) System Requirements and Limitations

    MButton requires a three button mouse and AmigaDOS V2.04 or higher.
```

1.3 system_desc

```
2) System Description

MButton is a quick hack I threw together after getting my 3 button
mouse.  It is built from one of Commodore's example commodities (mba)
and some existing bits and pieces.

MButton allows the user to program the middle button down and/or up
event to insert any character, string or input event into the input
stream.  Input events are surrounded by angle brackets <>, see
Input  Description Strings.  Anything not surrounded by angle brackets
is inserted as is.

Sixteen different strings can be set.  With any one of the qualifier
keys (CTRL, SHIFT, ALT, Left AMIGA and Right AMIGA), Left or Right
buttons, or the mouse event alone.
```

1.4 startup

```
3) Program Startup

USAGE: MButton [DOWN=<string>]        [UP=<string>]
               [SHIFT_UP=<string>]    [SHIFT_DOWN=<string>]
               [ALT_UP=<string>]      [ALT_DOWN=<string>]
               [RCMD_UP=<string>]     [RCMD_DOWN=<string>]
               [LCMD_UP=<string>]     [LCMD_DOWN=<string>]
               [CRTL_UP=<string>]     [CRTL_DOWN=<string>]
               [RBUTTON_UP=<string>]  [RBUTTON_DOWN=<string>]
               [LBUTTON_UP=<string>]  [LBUTTON_DOWN=<string>]
               [PORT_NAME=<rexx_port_name>]

All arguments are optional but it don't do much but use up some memory if
you don't set at least one of the up or down strings.
```

```
DOWN         - Sets the string for the mid button down event.
               The string can contain any number of characters
               and/or Input Description Strings.
               Anything not surrounded by angle brackets is
               inserted as is.

UP           - Sets the string for the mid button up event. See above.

SHIFT_DOWN   - Sets the string for the mid button down plus SHIFT key.
               See above.

SHIFT_UP     - Sets the string for the mid button up  plus SHIFT key.
               See above.

ALT_DOWN     - Sets the string for the mid button down plus ALT key.
               See above.

ALT_UP       - Sets the string for the mid button up plus ALT key.
               See above.

LCMD_DOWN    - Sets the string for the mid button down plus Left
               Amiga key. See above.

LCMD_UP      - Sets the string for the mid button up plus Left
               Amiga key. See above.

RCMD_DOWN    - Sets the string for the mid button down plus Right
               Amiga key. See above.

RCMD_UP      - Sets the string for the mid button up plus Right
               Amiga key. See above.

CTRL_DOWN    - Sets the string for the mid button down plus CTRL key.
               See above.

CTRL_UP      - Sets the string for the mid button up plus CTRL key.
               See above.

LBUTTON_DOWN - Sets the string for the mid button down plus left button.
               See above.

LBUTTON_UP   - Sets the string for the mid button up plus left button.
               See above.

RBUTTON_DOWN - Sets the string for the mid button down plus right button.
               See above.

RBUTTON_UP   - Sets the string for the mid button up plus right button.
                See above.

PORT_NAME    - Sets the Rexx port name.  Default is midserver_rx.


EG:
    CLI:        MBuuton "DOWN=<rshift ralt return>"
                MBuuton SHIFT_DOWN=:
```

```
        ToolTypes:  DOWN=<rshift ralt return>
                    SHIFT_DOWN=:
```

1.5 rexx_port

4) Rexx Port

The rexx port allows you to change the up and down event strings on the fly.

```
SET_UP - Sets the up event to the supplied string.  Note any leading or
         trailing blanks will be fed into the input stream. Also see
         UP under Program Startup.

    /* Very simple rexx example */
    address 'midserver_rx'

    SET_UP '<lcommand m>'        /* Set the up event to Left Amiga m */
    SET_DOWN                     /* Clear the down event string */
```

SET_DOWN - Sets the down event to the supplied string.

    See above example :)

SET_SHIFT_UP - Sets the SHIFT up event to the supplied string.

SET_SHIFT_DOWN - Sets the SHIFT down event to the supplied string.

SET_ALT_UP - Sets the ALT up event to the supplied string.

SET_ALT_DOWN - Sets the ALT down event to the supplied string.

SET_LCMD_UP - Sets the Left AMIGA up event to the supplied string.

SET_LCMD_DOWN - Sets the Left AMIGA down event to the supplied string.

SET_RCMD_UP - Sets the Right AMIGA up event to the supplied string.

SET_RCMD_DOWN - Sets the Right AMIGA down event to the supplied string.

SET_CTRL_UP - Sets the CTRL up event to the supplied string.

SET_CTRL_DOWN - Sets the CTRL down event to the supplied string.

SET_LBUTTON_DOWN - Sets the left button down event to the supplied string.

SET_LBUTTON_UP - Sets the left button up event to the supplied string.

SET_RBUTTON_DOWN - Sets the right button down event to the supplied string.

SET_RBUTTON_UP - Sets the right button up event to the supplied string.

BYE - Request MButton to close up shop and go home.

```
    /* Very simple rexx example */
    address 'midserver_rx'

    'BYE'
```

1.6 installation

5) Installation

Installation is fairly easy. Drag MButton's icon into your WBStartup drawer or any place you like (Sys:Tools/Commodities ?). Next, single click it's icon then select 'Information' from Workbench's 'Icons' menu. Set the DOWN and UP strings as desired and click on save. Then double click the icon and try it out.

1.7 input_strings

Note:

    This description is copied directly from the docs for one of
    Stefan Sticht's commodities.

Input Description Strings

With input description strings you can specify almost any input action, for example the action lshift f1, which means that pressing the left shift and the f1 key together is the action. In this commodity you can specify the action to open the commodity's window, as described above.

Input description strings have the following template:

```
      [class] (([-]qual)|syn)* [[-]upstroke] [highmap|ANSIcode]
```

(* means zero or more occurances of the of the expression in brackets)

```
class   is one of the following strings:
        rawkey, rawmouse, event, pointerpos, timer, newprefs,
        diskremoved, diskinserted.
        If not specified, the class is taken to be "rawkey".

qual    is one of the strings:
        lshift, rshift, capslock, control, lalt, ralt, lcommand,
        rcommand, numericpad, repeat, midbutton, rbutton, leftbutton,
        relativemouse
        A preceding '-' means that the value of the corresponding
        qualifier is to be considered irrelevant.
```

```
syn     (synonym) is one of the strings: shift, caps, alt
        shift means "left or right shift"
        caps means "shift or capslock"
        alt means "either alt key"

upstroke (literally "upstroke")
        if this token is absent, only downstrokes are considered
        for rawmouse (mousebuttons) and rawkey events.  If it is
        present alone, only upstrokes count.  If it preceded by
        '-' it means that both up and down strokes are included.

highmap one of the strings:
        comma, space, backspace, tab, enter, return, esc, del, up, down,
        right, left, help, f1, f2, f3, f4, f5, f6, f7, f8, f9, f10,
        0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (, ), /, *, -, +

ansicode a single character token is interpreted as a character code,
        which is looked up in the system default keymap.
```