#₁$₂+₃ **Here are some buttons to experiment with:**

Buttons can call Viewer functions {ewc MVPUSH,MVPushBox,[U:up.bmp] [D:dn.bmp] [C:History()]}

{ewc MVPUSH,MVPushBox,[U:rndup.bmp] [D:rnddn.bmp] [C:PopupID("PUSH.MVB","pop1>mypop")]}   {ewc MVPUSH,MVPushBox,[U:rndup.bmp] [D:rnddn.bmp] [C:PopupID(`PUSH.MVB',`pop2>mypop')]} Click for helpful hint   Click for more advice.

{ewc MVPUSH,MVPushBox,[U:swoff.bmp] [D:swon.bmp] [C:PaneID(`PUSH.MVB>main',`Movie3>Video',5)]} Flick this switch to see a movie.

{ewc MVPUSH,MVPushBox,[U:run.bmp] [D:halt.bmp] [C:PaneID(`PUSH.MVB>main',`Movie2>Video',5)]}

1#Demo
2$First Page
3+MainSQ:02

This is the panic button!!!!.

$^{\#}_5$

OK. OK, go ahead and panic !!!

$5^{\#}$pop2

OK. Let loose and freak out.

#₇!₈+₉Microsoft did it with Cinamania, and now you can too!

With simple Viewer Commands and the MVPUSH embedded window you can create complex controls. This remote control is made up from a background bitmap and panes for buttons. Try it out...

Developers Documentation

This DLL being released in an unsupported demonstration format.
MVPUSH is copyright Keyboard Publishing 1993. Developed by William Reichardt.
You are permitted to use this code library in your in house development only.
For information about licensing this product for commercial redistribution contact
Alexander Grimwade at (800) 945-4551.

This is a demonstration version of this product and does not support the storage of bitmaps in baggage. Please report any bugs to me (William Reichardt) at compuserve 71151,253.

To Build the sample project

All materials to build the sample are included except a file called BOAT_LHS.AVI
which can be found on your viewer final release disk. Please copy it to the same
Directory as the build MVB file before running it.

To use this product in your own Title
The custom controls in this project are actually custom bitmaps. One bitmap is the up state and another can be defined as a down bitmap. When the button is pushed down a viewer command can be executed.

Here is an example:

```
{ewc MVPUSH,MVPushBox,[U:b2up.bmp] [D:b2dn.bmp] [C:Prev()]}
```

the three parameters in [] are the only part of the command that should ever be changed. The U: command is the up bitmap. The D: command is the down bitmap and the C: command is the Viewer command that should be executed. More than one command can be executed if they are separated with a semicolon (;).

Keep in Mind...

1. This DLL will behave unexpectedly if you use incorrect syntax. The correct syntax is: [U:mybitmap.bmp] [D:mybmp2.bmp] [C:myviewercommand]. Do not omit any parameters either.

2. EW commands cannot get the focus and therefore cannot be controlled by the keyboard.

3. Images stored in baggage are not supported in this version and will not be displayed properly.

7#BigDemo
8!PaneID(`PUSH.MVB>main', `But1>B1', 2);PaneID(`PUSH.MVB>main', `But2>B2', 3);PaneID(`PUSH.MVB>main', `But3>B3', 4)
9+MainSQ:01

#10{ewl MVBMP2, ViewerBmp2, remote.bmp}

10#BackRemote

[#11]{ewc MVPUSH,MVPushBox,[U:b1up.bmp] [D:b1dn.bmp] [C:Next()]}

11[#]But1

[#12]{ewc MVPUSH,MVPushBox,[U:b2up.bmp] [D:b2dn.bmp] [C:Prev()]}

12[#]But2

[#13]{ewc MVPUSH,MVPushBox,[U:b3up.bmp] [D:b3dn.bmp] [C:PaneID(`PUSH.MVB>main',`Movie1>Video',5)]}

13[#]But3

[#14]{ewl MVMCI2, ViewerMCI, [device AVIVideo][autostart]boat_lhs.avi}

14[#]Movie1

#15{ewl MVMCI2, ViewerMCI, [device AVIVideo][autostart]boat_lhs.avi}

15#Movie2

#16{ewl MVMCI2, ViewerMCI, [device AVIVideo][autostart]boat_lhs.avi}

16#Movie3