



基本篇

**config f.button**  
**f.thumbnail f.window**

解析布局设置配置文件

07 年 11 月 19 日

此部分为 **config f.button f.thumbnail f.window** 内容分析。包括所有的动作定义。

全文如下:

## config

```
1 # vim: syntax=fvwm
2 SetEnv fvwm          ${HOME}/.fvwm/
3 SetEnv thumbnail     ${fvwm}tmp-thumbnail-pic/
4 #SetEnv fvwmbutton    ${fvwm}fvwmbutton/
5 SetEnv theme         ${fvwm}theme/
6 SetEnv button        ${theme}button-theme/
7 SetEnv title         ${theme}title-theme/
8 #SetEnv fvwm_icon_size 128
9
10 # 允许的颜色 r/g/b
11 #SetEnv fcolor svg
12
13 ImagePath +:${fvwm}:${thumbnail}:${fvwmbutton}:${theme}:${button}:${title}
14
15 #SetEnv RegularFont "Shadow=1 UpRight:xft:AR PL ZenKai Uni:bold:size=10:encoding=iso10646-1"
16 SetEnv RegularFont "StringEncoding=utf8:xft:Vera Sans YuanTi Mono:size=9:encoding=iso10646-1"
17 #Style * Font ${RegularFont}
18
19 DesktopSize 2x2                #每个桌面分成 2x2 个页面
20 DesktopName 0 Main              #设置了 1 个桌面
21
22 #Exec xpmroot /home/exp/媒体/桌面图片/ice.png
23 #Exec hsetroot -center /home/exp/媒体/桌面图片/leaf.png
24 #Exec rox --pinboard=exp 桌面
25 #Exec fvwm-root -r /home/exp/媒体/桌面图片/ice.png
26 Exec fvwm-root -r /home/exp/媒体/1280x800/green.png
27 # Exec fvwm-root -r /home/exp/媒体/1280x800/blue.png
28 Exec nm-applet &
29 Exec exec adesklets -d 3
30 Exec exec conkystart.bash
31 Exec killall -9 gnome-screensaver
32 Exec gnome-volume-manager &
33 # Exec xpad -H
34
35 # Exec exec idesk
36 Read f.window
37 Read f.button
38 # Exec gnome-panel
39 Read f.thumbnail
40 #Read f.autohide
41 Read f.action
42 # PipeRead ${fvwm}/auto-menu.bash
43
44 Module FvwmCommandS
45 #now can excute "FvwmCommand FvwmIdent" and other fvwm commands from terminal
46 #FvwmCommand Restart
47 #####
48 DestroyMenu MenuFvwmRoot
49 #Exec adesklets
50 # Exec exec trayer --widthtype pixel --width 90 --align left --expand true --transparent true --alpha 25
51 # cursor name defined in /usr/include/X11/cursorfont.h
52 CursorStyle DEFAULT right_ptr
53 #CursorStyle TITLE front1.xpm
54 #CursorStyle TITLE cross.xpm
55 # Read /usr/share/fvwm-crystal/fvwm/decorations/Aqua/Buttons\ with\ icons/Theme.fvwm-crystal
56
```

## f.button

```
1 # vim: syntax=fvwm
2 #####
3 # 配置桌面上的浮动启动栏 (常用程序启动器)
4 #####
5 #不显示桌面序号
6 # *FvwmPager: Geometry 60x60
7 # *FvwmPager: IconGeometry 20x20
8 Colorset 17 RootTransparent
9 *FvwmPager: Font none
10 # *FvwmPager: WindowColors grey lightgrey yellow lightblue
11 *FvwmPager: MiniIcons
12 # *FvwmPager: DeskPixmap 0 b-smile.png
13 # *FvwmPager: Pixmap b-smile.png
14 *FvwmPager: Colorset 0 17
```

```

15 # *FvwmPager: WindowBorderWidth 15
16 # *FvwmPager: Window3DBorders
17 # *FvwmPager: HighlightPixmap b-smile.png
18 *FvwmPager: Highlight white
19 *FvwmPager: WindowColors grey lightgrey grey orange
20 # *FvwmPager: SolidSeparators
21 *FvwmPager: NoSeparators
22
23 Style Xosview WindowListSkip, CirculateSkip, NoTitle, !State 1
24 #Style FvwmApplet-DigitalClock NoTitle, !State 1, WindowListSkip
25 Style cairo-clock WindowListSkip, CirculateSkip, NoTitle, !State 1
26
27 Style "DockButtons" !Title, !Borders, Sticky, WindowListSkip, CirculateSkipIcon, CirculateSkip
28 Style "DockButtons" HandleWidth 1, BorderWidth 0, FixedSize, !Iconifiable, StaysOnBottom
29 #Style "DockButtons" Pixmap button-pressed.png
30 #Style "DockButtons" Shape bg1.png
31
32 Colorset 16 RootTransparent buffer, Tint white 10
33 #, Pixmap trough2-h.png
34 #, bg average, bgTint white 20
35 Colorset 19 RootTransparent, Tint orange 60
36
37 DestroyModuleConfig DockButtons: *
38
39 #不设置行列, 系统自动配置
40 *DockButtons: Columns 2
41 #*DockButtons: Rows 15
42 *DockButtons: Geometry 100x450
43 *DockButtons: Frame 0
44
45 *DockButtons: Colorset 16
46 *DockButtons: ActiveColorset 19
47
48 # *DockButtons: (2x1,Icon fvwm_banner_1.png, Action(Mouse 1) Menu auto-menu)
49 *DockButtons: (2x1,Icon fvwm_top.png, Action(Mouse 1) Menu auto-menu)
50 # *DockButtons: (2x5,Swallow (UseOld,Kill) "conky" "Exec conky")
51 # *DockButtons: (2x2,Swallow "xosview" "Exec xosview")
52 # *DockButtons: (2x2,Swallow (UseOld,Kill) "cairo-clock" "Exec cairo-clock -s -t silvia")
53 # *DockButtons: (2x2,Swallow "trayer" "Exec exec trayer --SetDockType true --widthtype pixel --width 80
--expand true --align left --heighttype pixel --height 80")
54 *DockButtons: (2x2,Swallow (UseOld,Kill) "stalonetray" "Exec exec stalonetray -t --max-width 96")
55 # *DockButtons: (2x1,Swallow "FvwmDate" "FvwmScript ${fvwm}/f.script.clock")
56 # *DockButtons: (2x2,Padding 6 8,Swallow "FvwmPager" "Module FvwmPager")
57 # *DockButtons: (2x2,Icon "t-all.png", ActiveIcon "s-t-all.png" \
58 # , Id B1 \
59 # , Action(Mouse 1) Exec gnome-terminal \
60 # , Action(Mouse 2) Exec exec gksudo -u root -k -m "要重启是不? 输入密码。" reboot \
61 # , Action(Mouse 3) Exec exec gksudo -u root -k -m "要关机是不? 输入密码。" poweroff \
62 # , Action(Mouse 4) Exec amixer set PCM 5%+ \
63 # , Action(Mouse 5) Exec amixer set PCM 5%- \
64 # )
65 # , Action(Mouse 2) Exec ${fvwm}root-重启 \
66 # , Action(Mouse 3) Exec ${fvwm}root-关机 \
67 # , Frame 1, Action Beep, Id Id_term \ Frame 按钮边框, 上下动态的。不好看。
68 # *DockButtons: (2x2,Icon "f-all.png", ActiveIcon "s-f-all.png" \
69 # , Id B2 \
70 # , Action(Mouse 1) Exec rox \
71 # , Action(Mouse 2) Exec nautilus \
72 # , Action(Mouse 3) Exec rox \
73 # )
74 # *DockButtons: (2x2,Icon "n-all.png", ActiveIcon "s-n-all.png" \
75 # , Id B3 \
76 # , Action(Mouse 1) Exec opera \
77 # , Action(Mouse 2) Exec pidgin \
78 # , Action(Mouse 3) Exec /home/exp/应用/mlnet \
79 # )
80
81 *DockButtons: (1x1, Id cursor, Icon "cursor_right.png", Action(Mouse 1) Exec exec '/home/exp/.fvwm/fvwm-
swap-button.bash')
82 *DockButtons: (1x1, Id terminal, Icon "terminal.png", ActiveIcon "terminal-s.png", Action(Mouse 1) Exec
exec gnome-terminal --geometry 99x29, Action(Mouse 2) Exec exec rxvt, Action(Mouse 3) Exec exec rox)
83 *DockButtons: (2x2,Icon gesture.png)
84 *DockButtons: (2x2,Icon ctrl-gesture.png)
85 *DockButtons: (2x2,Icon alt-gesture.png)
86
87 # *DockButtons: (1x1, Id B5, Icon "photo.png", Action(Mouse 1) Exec exec '/home/exp/应用/脚本/import-截
图', Action(Mouse 2) Exec exec scrot -s -b, Action(Mouse 3) Exec exec gnome-screenshot --delay 5)
88
89 # *DockButtons: (1x1, Id record, Icon "r-24-end.png", PressIcon "r-24-rec.png", Action(ActionOnPress,
Mouse 1) Exec '/home/exp/应用/脚本/sound-recorder.bash')
90 # , ActionOnPress(Mouse 1) Exec echo press>~/press)
91

```

```

92 AddToFunc StartFunction
93 + I Module FvwmButtons DockButtons
94 #+ I FvwmScript /home/exp/fvwm/exp/f.script
95 #SendToModule DockButtons Silent ChangeButton B1 Icon "t-all1.png"

```

## f.thumbnail

```

1 # vim: syntax=fvwm
2 Style * IconBox 260x700+120+50, IconGrid 20 20, IconFill top left
3
4 DestroyFunc Thumbnail
5 AddToFunc Thumbnail
6 + I Raise
7
8 + I Exec exec echo id=${w.id} > ${thumbnail}/w.info
9 + I Exec exec echo name=${w.name} >> ${thumbnail}/w.info
10 + I Exec exec echo class=${w.class} >> ${thumbnail}/w.info
11 + I Exec exec echo resource=${w.resource} >> ${thumbnail}/w.info
12
13 + I Exec exec echo iconname=${w.iconname} >> ${thumbnail}/w.info
14 + I Exec exec echo iconfile=${w.iconfile} >> ${thumbnail}/w.info
15 + I Exec exec echo miniiconfile=${w.miniiconfile} >> ${thumbnail}/w.info
16
17 + I Exec exec echo _NET_WM_ICON=${_NET_WM_ICON} >> ${thumbnail}/w.info
18 + I Exec exec echo WM_HINTS=${WM_HINTS} >> ${thumbnail}/w.info
19
20 + I ThisWindow (!Shaded Iconifiable !Iconic) PipeRead "${fvwm}/thumbnail.bash ${w.id} ${w.class} ${w.resource} && echo WindowStyle IconOverride, Icon ${thumbnail}/thumbnail.${w.id}.png"
21 + I Iconify
22 + I Exec mpg321 ${theme}/ballon.mp3
23
24 #窗口返回到通常状态时，可以使用模块 FvwmEvent 删除 Thumbnail 产生的图标。
25 DestroyFunc DeThumbnail
26 AddToFunc DeThumbnail
27 + I Exec exec rm -f ${thumbnail}/thumbnail.${w.id}.png
28 + I Raise
29 + I WarpToWindow 50 50
30 + I Iconify off
31 + I Exec mpg321 ${theme}/ballon.mp3
32
33 Style * !IconTitle, IconTitleRelief 0, IndexedIconName, IconBackgroundPadding 0, IconBackgroundRelief 0

```

## f.window

```

1 # vim: syntax=fvwm
2 #####
3 # 替换成了 Mac-OS 风格窗口按钮
4
5 # Colorset 0 fg lightgray, bg lightgray, sh lightgray, \
6 # fgsh darkgray, RootTransparent buffer
7 # Colorset 1 fg white, bg white, sh white, \
8 # fgsh rgb:05/46/a0, RootTransparent buffer
9
10 # 允许的颜色 r/g/b
11 SetEnv fcolor g
12
13 Style * Font "xft:经典繁体:bold:size=12:encoding=iso10646-1"
14 #TitleStyle MiniIcon LeftMain
15 DestroyDecor MacOS
16 AddToDecor MacOS
17 # + TitleStyle Height 28 -- flat
18 # + TitleStyle Active (AdjustedPixmap title-active.png) \
19 # InActive (AdjustedPixmap title-inactive.png) \
20 # LeftJustified -- Flat
21 #+ TitleStyle MiniIcon -- Left
22 # + TitleStyle Active (MultiPixmap Main AdjustedPixmap title-active.png, \
23 # LeftMain AdjustedPixmap r-24-female.png, \
24 # RightOfText TiledPixmap r-24-male.png) \
25 # InActive (AdjustedPixmap title-inactive.png) \
26 # LeftJustified -- Flat
27 # + BorderStyle Simple -- HiddenHandles NoInset
28 + TitleStyle \
29 Height 28 \
30 Active (MultiPixmap \
31 Main TiledPixmap t-main-$(fcolor).png, \
32 Buttons TiledPixmap b-bg-$(fcolor).png, \
33 LeftEnd AdjustedPixmap t-left-$(fcolor)-e.png, \
34 RightEnd AdjustedPixmap t-right-$(fcolor).png) \
35 Inactive (MultiPixmap \
36 Main TiledPixmap nt-main.png, \
37 Buttons TiledPixmap nb-bg.png, \

```

```

38             LeftEnd             AdjustedPixmap nt-left.png, \
39             RightEnd            AdjustedPixmap nt-right.png) -- flat
40
41 # + ButtonStyle 6 Active (Pixmap tb_min_on.png)
42 # + ButtonStyle 4 Active (Pixmap tb_max_on.png)
43 # + ButtonStyle 2 Active (Pixmap tb_close_on.png)
44
45 # + ButtonStyle 9 Active (Pixmap tb_dot_on.png)
46 # + ButtonStyle 7 Active (Pixmap tb_shade_on.png)
47 # + ButtonStyle 5 Active (Pixmap tb_all_on.png)
48 # + ButtonStyle 3 Active (Pixmap tb_on.png)
49 # + ButtonStyle 1 Active (Pixmap tb_top_on.png)
50 # + ButtonStyle All Inactive (Pixmap tb_minoff.png)
51 + ButtonStyle 8 Active (Pixmap b-smile.png)
52 + ButtonStyle 6 Active (Pixmap b-min.png)
53 + ButtonStyle 4 Active (Pixmap b-max.png)
54 + ButtonStyle 2 Active (Pixmap b-close.png)
55 + ButtonStyle 0 MiniIcon
56
57 + ButtonStyle 9 Active (Pixmap b-off.png)
58 + ButtonStyle 7 Active (Pixmap b-shade.png)
59 + ButtonStyle 5 Active (Pixmap b-all.png)
60 + ButtonStyle 3 Active (Pixmap b-no.png)
61 + ButtonStyle 1 Active (Pixmap b-top.png)
62 + ButtonStyle All Inactive (Pixmap b-no-no.png)
63
64 # + ButtonStyle 2 InActive (Pixmap g-close.png) Active (Pixmap c-close.png)
65 # + ButtonStyle 4 InActive (Pixmap g-max.png) Active (Pixmap c-max.png)
66 # + ButtonStyle 6 InActive (Pixmap g-iconify.png) Active (Pixmap c-iconify.png)
67 # + ButtonStyle 8 InActive (Pixmap g-shade.png) Active (Pixmap c-shade.png)
68
69 # + ButtonStyle 7 (Pixmap r-24-screenshot.png)
70 # + ButtonStyle 1 InActive (Pixmap g-stick.png) Active (Pixmap c-stick.png)
71 # + ButtonStyle 3 InActive (Pixmap g-transparent.png) Active (Pixmap c-transparent.png)
72 # + ButtonStyle 5 InActive (Pixmap g-stack.png) Active (Pixmap c-stack.png)
73 + ButtonStyle All -- UseTitleStyle -- Flat
74 # RootTransparent buffer
75 # + ButtonStyle All Active (RootTransparent buffer -- flat)
76 # + ButtonStyle All Active (Colorset 0 -- flat) Inactive (Colorset 1 -- flat)
77
78 #####
79 # ToggledInActive (Pixmap c-clock.png) \
80 # ToggledActive (Pixmap c-kill.png)
81 #+ ButtonStyle 5 - MwmDecorStick
82 #+ ButtonStyle 4 - MwmDecorMax
83 #+ ButtonStyle 8 - MWMDecorShade
84 #MwmDecorMenu
85 #MwmDecorMin
86 #Raised
87 #+ BorderStyle Active -- Flat InActive -- Sunk
88 #+ BorderStyle InActive (bg grey, bgTint red 20, HandleWidth 5) Active (HandleWidth 2)
89 #+ HilightBorderColor Colorset 26
90 #+ BorderColorset Colorset 26
91 #####
92 Style MacOS-X UseDecor MacOS, HandleWidth 2
93 # , TitleAtLeft
94 Style * UseStyle MacOS-X #启用
95 Style rox* UseDecor MacOS, HandleWidth 2, TitleAtTop
96 Style beep-media-player NoTitle
97
98 *FvwmAnimate: Delay 5
99 *FvwmAnimate: Effect Random
100 *FvwmAnimate: Width 5
101 AddToFunc StartFunction
102 + I Module FvwmAnimate
103 #+ I Module FvwmIconBox
104 # FvwmAnimate #图标化时的动画
105 #+ I Module FvwmAuto 500 Raise Nop #自动上升窗口
106 #+ I Module FvwmTaskBar

```

开始分部分拆解分析：

## config

这个文件，就是 `fvwm` 缺省调用的文件。位于 `~/ .fvwm/config`。缺省的，一般我并不放在此隐藏目录里面，而是在自己喜欢的目录里面。然后作一个软链接。`ln -s xxxxx/xxxxx ~/ .fvwm`。这样作的好处是，你如果想调试，或者切换你的不同配置。简单的修改这个链接，然后直接执行 `fvwm -replace`，甚至是直接执行 `Restart` 命令，比如 `FvwmCommand Restart` 就可以了。

```
2 SetEnv fvwm           ${HOME}/.fvwm/
3 SetEnv thumbnail      ${fvwm}tmp-thumbnail-pic/
```

这些是定义一些环境变量。配置文件，或者你的函数等，可以调用，其实是简化书写而已。编程过的，基本都容易理解。

```
13 ImagePath +:${fvwm}:${thumbnail}:${fvwmbutton}:${theme}:${button}:${title}
```

缺省的图片搜索路径。如果不设置，你的配置文件会很难看的。

```
16 SetEnv RegularFont "StringEncoding=utf8:xft:Vera Sans YuanTi Mono:size=9:encoding=iso10646-1"
```

设置字体变量，也是内部环境变量。

```
19 DesktopSize 2x2           #每个桌面分成2x2个页面
20 DesktopName 0 Main        #设置了1个桌面
```

设置桌面布局。这里是 `2x2` 的页面，作为一个桌面。然后定义了桌面 `0` 的名字叫 `Main`。基本，对于我来说，`4` 个页面够用了。软件都分类在不同页面打开的。

```
26 Exec fvwm-root -r /home/exp/媒体/1280x800/green.png
```

定义 `root` 桌面的图片。很多方法可以实现的。你自己作一个 `shell`，自动换图片也行。也可以实现每个页面使用不同的图片，不过很占用资源的，鼠标在穿越页面的时候，会有些卡的感觉。反正我不需要。平时根本不看什么桌面。实现的方法，基本如下。不推荐的。

```
Colorset 10 TiledPixmap foo.xpm
```

```
Colorset 11 Pixmap bar.png
```

```
FvwmBacker: Command (Desk *, Page * 0) Colorset 10
```

```
FvwmBacker: Command (Desk *, Page * 1) Colorset 11
```

```
30 Exec exec conkystart.bash
```

这是启动的时候，执行些你需要的东西。我加了一个 `conky` 在桌面。只是看资源信息而已。



```
32 Exec gnome-volume-manager &
```

这个进程，是为了让系统自动挂载外部磁盘，当然，完全遵循 `gnome` 的设置。这样，在 `rox` 里面，省得要自己去输入 `pmount`。

```
36 Read f.window
37 Read f.button
39 Read f.thumbnail
41 Read f.action
```

上面这些，就像是 `C` 里面的 `include` 啦。自动读入其他配置文件。方便自己的配置和修改。

```
44 Module FvwmCommandS
```

加载这个 `fvwm` 内部模块。这样，你就可以使用 `FvwmCommand Restart` 这样的命令。直接在终端里面执行的哦。用处其实不大。偶尔调试些命令有用。

```
48 DestroyMenu MenuFvwmRoot
```

销毁 `fvwm` 本身的 `root` 菜单。我自己的菜单都不用了。要它的干嘛。如果有人需要自动菜单的说明。我以后在详细写。反正菜单不如鼠标手势方便。常用的软件，没几个的。

```
52 CursorStyle DEFAULT right_ptr
```

设置下鼠标的形状。`linux` 的鼠标，还不太统一。在不同的软件环境开发的，都不统一。这里是使用 `x` 的鼠标。这是一个麻烦的事情。

```
55 # Read /usr/share/fvwm-crystal/fvwm/decorations/Aqua/Buttons\ with\ icons/Theme.fvwm-crystal
```

补充下，有人如果使用 `fvwm-crystal` 的主题。可以直接这样的。甚至用 `FvwmCommand` 随时加载。

## f.button

这个文件是定义一个 `FvwmButton` 的按钮工具栏。类似表格的。

```
8 Colorset 17 RootTransparent
```

颜色集，是 `fvwm` 控制外观的一种形式。可以定义很长的样式。当然，不如 `css` 那么容易。

```
9 *FvwmPager: Font none
```

具体的模块，这个是页面缩略图的模块。参数不动，就可以用。要好看，只能细调。`man FvwmPager` 吧。

```
23 Style XOswiew WindowListSkip, CirculateSkip, NoTitle, !State 1
```

`Style` 是设置一种特定的样式管理。可以定义一类 / 全部 / 单个的软件样式。用 `*` 通配符就是。后面的具体参数，也说不完的。`man fvwm`，搜索吧。

```
27 Style "DockButtons" !Title, !Borders, Sticky, WindowListSkip, CirculateSkipIcon, CirculateSkip
```

```
28 Style "DockButtons" HandleWidth 1, BorderWidth 0, FixedSize, !Iconifiable, StaysOnBottom
```

这里是自己定义一个名字叫 `DockButtons` 的样式类型。设置了没标题栏，没边框，全部页面都显示，不进窗口列表，也不进入 `fvwm` 内部命令的管理（`fvwm` 的函数，可以用很多命令，比如 `Next` 等，遍历窗口，进行管理的，和普通编程是一样的）。第 2 行，是设置边框宽度，固定尺寸（其实就一 `dock` 一样的东西），不能缩略图，置底。下面的行还有透明等，调整颜色的，以及表格那样的布局，尺寸定义。

```
49 *DockButtons: (2x1,Icon fvwm_top.png, Action(Mouse 1) Menu auto-menu)
```

开始在表格里面，添加事件。`2x1` 表示表格的占用情况。从左上角开始计算。贴个小图片。定义鼠标主键动作是显示一个菜单。`auto-menu` 菜单在现在的设置里面，已经废弃了。格式类似于：

```
#DestroyMenu Menu_App
```

```
#AddToMenu Menu_App
```

```
#      + "GXmame%D5.png%"      Exec  exec gxmame
```

```
#      + "Calculator%D1.png%"  Exec  exec gnome-calculator
```

其中，第一行是销毁这名字的菜单定义（防止多个地方定义了相同的菜单）。接着是添加项目。`%%` 包括的部分，是菜单图片。我以前是用一个 `bash` 处理的，自动的找需要的软件的 `desktop` 文件，分析下 `Icon` 图片和 `Exec` 执行部分。自动使用 `PipeRead` 读取的菜单。

```
54 *DockButtons: (2x2,Swallow (UseOld,Kill) "stalonetray" "Exec exec stalonetray -t --max-width 96")
```

`UseOld`, `Kill` 这样的语法，是针对某些只要一个进程的软件。如果 `fvwm` 经常重启的话，可能会启动多个相同的进程。使用 `UseOld` 的话，`fvwm` 就会自动去识别是否有相同的进程已经启动了。

`Swallow` 是一个绝好的东西，什么软件的界面，都可以在 `fvwm` 的多个模块中，被吞下。这里是在 `FvwmButtons` 模块中使用。比如你想在这 `Buttons` 里面显示一个时钟，显示一个 `mplayer`？这个我没试过。鼓励大家都试试。

```
55 # *DockButtons: (2x1,Swallow "FvwmDate" "FvwmScript ${fvwm}/f.script.clock")
```

`FvwmScript` 模块，可以画出界面的。和普通编程类似。画一个窗口，一个执行程序界面。当然别想画太复杂的软件。这不是此模块的特长，否则，这模块应该改成 `FvwmC++` 了。

```
81 *DockButtons: (1x1, Id cursor, Icon "cursor_right.png", Action(Mouse 1) Exec exec '/home/exp/.fvwm/fvwm-swap-button.bash')
```

`Id` 是标识一个部件的识别号。比如这里调用的一个 `bash`。我在 `bash` 里面可以反过来修改指定了 `Id` 的部件的各种特性。实际配置里面，我是修改了这个按钮的图标。因为是鼠标光标的按钮，切换左右手的，当然，图标要跟着变，以反应实际的鼠标模式。

```
82 *DockButtons: (1x1, Id terminal, Icon "terminal.png", ActiveIcon "terminal-s.png", Action(Mouse 1) Exec exec gnome-terminal --geometry 99x29, Action(Mouse 2) Exec exec rxvt, Action(Mouse 3) Exec exec rox)
```

这是一个按钮对应多个事件触发的行。鼠标有 1 - 9，甚至 1 - 12 个按钮的。发挥你的想象吧。

```
83 *DockButtons: (2x2,Icon gesture.png)
```

什么动作都不定义。这是我的鼠标手势的说明书。呵呵。

```
92 AddToFunc StartFunction
```

```
93 + I Module FvwmButtons DockButtons
```

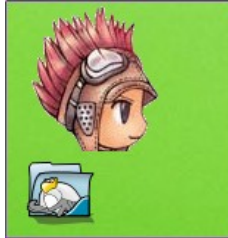
上面，都只是定义了一个 `DockButtons` 而已。这里，把这个 `FvwmButtons` 模块加载到启动函数 `StartFunction`。

## f.thumbnail

这个文件是定义缩略图样子的配置文件。估计自己喜欢情况修改。

```
2 Style * IconBox 260x700+120+50, IconGrid 20 20, IconFill top left
```

定义一个缩略图的位置，包括分割成什么样的表格形式，以及 *Icon* 进来的时候，填充的起始方向。其实是 *IconBox* 管理的。如果你对窗口直接发 *Notify* 命令。一样会有图标显示。类似这样：



只是，这样似乎不太好看。所以，使用下面的函数来处理下缩略图的效果。

```
4 DestroyFunc Thumbnail
5 AddToFunc Thumbnail
```

老套的定义函数的方式。

```
6 + I Raise
```

点击的窗口，上升。

```
20 + I ThisWindow (!Shaded Iconifiable !Iconic) PipeRead "${fvwm}/thumbnail.bash ${w.id} ${w.class} $
[w.resource] && echo WindowStyle IconOverride, Icon ${thumbnail}/${thumbnail}.${w.id}.png"
```

针对当前窗口，指定一下状态（没窗口卷帘的，不是图标化的，允许图标化的），执行命令。PipeRead，就是把后面命令的输出，直接传给 *fvwm*，如果你的输出不对，那效果也就不对哦。输出应该是 *fvwm* 可以认到的命令。

命令里面，先传入给 *bash* 的，有 2 个参数，就是当前窗口的 *id* 窗口号和 *class* 类名，*resource* 资源名。这些，大家用 *xwininfo* 自己理解吧。接着，这个 *bash* 会截图当前窗口，生成一个缩略图，放到指定的位置，使用指定的名称（*thumbnail.\${w.id}.png*）。命令，接着就 *echo* 一行出来。为什么是 *echo* 呢。就是输出嘛。*echo* 的内容，就是设置窗口的图标，注意，是 *IconOverride*，就是覆盖掉原始的窗口图标信息（不是对软件本身了，是对系统的缓冲设置）。

```
21 + I Iconify
```

然后，执行图标化的命令。这时候的图标，就不是软件本身的了。而是你给的那图片。

```
22 + I Exec mpg321 ${theme}/ballon.mp3
```

给他一个声音吧。没声音的动作，没劲。

```
26 AddToFunc DeThumbnail
```

以下的 *DeThumbnail* 函数，基本能看懂了吧。删除那临时的图片，提升窗口，使用一个动画效果来恢复窗口，去图标化。也响一声提示。

效果大致如图。窗口带了原软件的图标的：（其中，没在 *desktop* 里面找到图标的软件，直接用带阴影的文字来显示的，这都是 *imagemagick* 强大的功能。）





## f.window

这个文件是定义窗口的装饰的 (decor)。这里使用了自己画的 svg 文件，渲染效果后，生成的 png 来画边框。

```
11 SetEnv fcolor g
```

设置一个宏变量。我使用这个来随时换标题栏的颜色的。

```
13 Style * Font "xft:经典繁体:bold:size=12:encoding=iso10646-1"
```

独立的设置下字体。因为在 config 主文件里面读入的配置，不使用前面定义的字体。

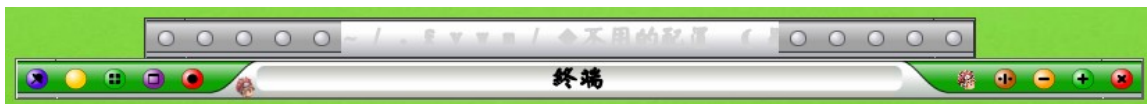
```
15 DestroyDecor MacOS
16 AddToDecor MacOS
```

样式定义开始。

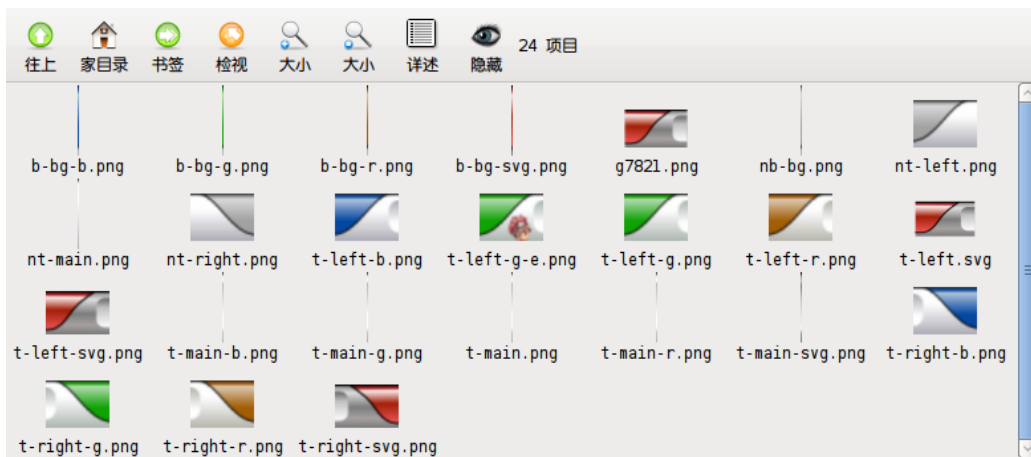
```
28 + TitleStyle \
29     Height 28 \
30     Active      (MultiPixmap \
31                 Main      TiledPixmap    t-main-[$fcolor].png, \
32                 Buttons   TiledPixmap    b-bg-[$fcolor].png, \
33                 LeftEnd   AdjustedPixmap t-left-[$fcolor]-e.png, \
34                 RightEnd  AdjustedPixmap t-right-[$fcolor].png) \
35     Inactive    (MultiPixmap \
36                 Main      TiledPixmap    nt-main.png, \
37                 Buttons   TiledPixmap    nb-bg.png, \
38                 LeftEnd   AdjustedPixmap nt-left.png, \
39                 RightEnd  AdjustedPixmap nt-right.png) -- flat
```

看着复杂的定义部分，其实好简单的。Active 是活动窗口，Inactive 是非活动的（使用的是灰度的图片）。标题栏高度 28。使用了 MultiPixmap 的模式，这样才能一部分一部分的指定窗口图片。注意，那个 fcolor 的变量，就是直接用的。因为我有几组图片。Tiled 就是平铺图片，Adjusted 就是展览图片。Main 是标题栏中间部分，Buttons 是按钮的背景部分。LeftEnd 就是标题栏显示文字的区域的小块。

实际效果如下。



使用到的图片如下。



```
92 Style MacOS-X UseDecor MacOS, HandleWidth 2
94 Style * UseStyle MacOS-X #启用
```

这里是启用这个样式。Style \* 是所有窗口。当然，你可以指定某些窗口使用红色，某些使用其他颜色，甚至边框样式。因为这桌面管理器的主人，就是你啊。

```
102 + I Module FvwmAnimate
```

这里是一个动画模块，就是缩略图缩小放大时候的那动画。其实应该写到 f.thumbnail 文件里面去。