XFree86 DDX Design

The XFree86 ProjectThe X.Org Foundation
Updates for X11R6.7: Jim Gettys

X Version 11, Release 7.7

X Server Version 1.12.1

2012-04-13

Table of Contents

? G G G G : V V 9 9 @ @ @ @ @

! " # " ! $ % " " & " # " ## "
' ( )
* ((
+ " # "
, " "
! <
, " (<
% G
G
% % " - # G
G
! . " 9
! & ! - T
! - ! / " " T
""% 0% 1 T
! <@
01 <@
" %$ <@
! , $ <
, %$ <
' " <
<=
" , " <=
" <=
" " " <?
% <?
<?
" <?
" % <G
! ## " " <G
" <:
(" <V
" <V
2 - 3 <T
& <T
& =
(=
! 4 5 ?
" 6 6 " " ?:

" ?:
& ?9
! # 7 " G
% ' G<
* ' G=
8 9: 8 , " ' 08 1 " + & GT
+ " :T
+ " $:T
% # + " :T
" ; # V<
(+ " VG
% V9
7 V9
VT
(" ## " " 9
& " 9G
% " & " T@
" " - T<
" $ T<
(" - " "$ T<
T=
$ # " T=
% T?
* $ T?
,* ! # @@
,* @@
% # ($ " @=
" @=
" @G
@V

" # " ($ " 6 "
" (" (> $ "
" " " <
6 (((& # & & $ " (

Preface

" # $ & " 8 & 8 9: ? @ 8
8 " " " 8 9: ? ? <
(8 > 6 " " 8 9: 8 & &
- " # & # #
8 9: 8 " 8 < 6 " #
8 9:

8 9: > A (" " 8 9: ? @ $ B

C - (

C D $ #

C D " A $ $

C - #"

C # & (" #"

C (" (& "(# #

C " $ $ " " & #

C 8 ((

C "(($

C "# # E " ## "

C - # # "

C # 6 " # "

C # " " " " " " " &
#

C (! " "

C (

C " ,* # (& #

D 8 9: > " " 8 &
& "# > " 6 " " # 8 # #
# & $ #6 8 " $ #
(" 6 "# - $ & # " #
8 9: 8 & " 8 0 "

" hw/xfree86 16 " " $ $
Xserver-spec " # 8 & ##
8 # # #

The xorg.conf File

(" 8 9:! # # 8 9: ? ?6
$ # "8 9: = 8 9:! # 6 $ $
B

Device section

$ & " (6 " " ("$
" " % # $ - &$ " "" "B

F" # F % # " (" "
" &

3 F(F % ; & " (
" & " # " "
&
6 " # & (" " ! (6
- F(B B F

" " 4 5

Screen section

$ & " (&
- &$ "6 (- &$ " "" " 0 - &$ " # & ("
- &$ " # (& 1 "
(" " & $ ($ #
(& # ## " "
((" " "
4 5 # ## " 6 #

InputDevice section

$ " (#
" " 6 " + -
6 $ # " & - &$ " B " #
(& " " " (&
$ B

F # ! 2 &("F
F-("F

F # ! F
F# F

" "
& (" # &
6 -screen ## " "6
" A & 6
$ " F-("F "F# F " - &(" "
&

ServerLayout section

$ " " & $
(" # "" 6 " &
" $ (" '
" 6 " 6 "
" # & (""
6 # - (" (

(# " (& (" ## "
(6 " $ (0 "1
& & ($ "(& #(& "
(6 " & (& # & & 6
(# 0 1 D
#("6 & #(" " " $ &
" "0 $ (1 # " " #
">" 6 (#6 6
7 # $ "
"6 $ " " B

```
Section "ServerLayout"
  Identifier "Main Layout"
  Screen     0 "Screen 1" ""   ""   ""   "Screen 2"
  Screen     1 "Screen 2"
  Screen     "Screen 3"
EndSection
```

"$ & & & & & A
(#

(6 A "
- &$ " " # "6 (0,0) #" #
($ B

```
Section "ServerLayout"
  Identifier "Main Layout"
  Screen     0 "Screen 1" Absolute 0 0
  Screen     1 "Screen 2" Absolute 1024 0
  Screen     "Screen 3" Absolute 2048 0
```

```
EndSection
```

6 "(& $
-&$ " $"(& # B

!"
"

 # $

D -&$ " "6 # $"(&
" $ A $
$ $

```
Section "ServerLayout"
  Identifier "Main Layout"
  Screen     0 "Screen 1"
  Screen     1 "Screen 2" RightOf "Screen 1"
  Screen     "Screen 3" Relative "Screen 1" 2048 0
EndSection
```

Options

"# & &#& # $
" " (6 " % "
(" " 6 6% 6
("" $ (" ""
" (6 "

Driver Interface

" # # & ; "("
" # /
" $& & "(& " "
"$ ""

& " $ "& - (& #(
" 0 " 6 8 1 ' " # #("
" " #

" " " (&6("
& " "" 6 8 9: ## & "
4 5 - # " "
" " ($ "
(&" $ 6 " (&
(" & "(# &" 6 &
"(# & (
" #& " # (" " " $ "

' " " B

% " # 8 &

' , -

+ ,

" %$! " #"

, $! & $

% % %)"

! % " # 8 &
"

"" 6 $; " (& 8 9:
& B

" & " " & #

($ " " & "$
- $ $ "

6 "
"$ 6 " " # "

, " " ; B

, " " " & $ #" ($
0 ") 1 # & # - , "
" # #

& ; & " $ #
&

Resource Access Control Introduction

* " ") # # & D #
#" " $ # & # ;
$ (" ,* # # & " 9G ?)
D #" (0 - ! 1 (# " "
! 0 ! 1 (& #
" # #

Terms and Definitions

Bus

43 5 #(" " B # & & #
(# & # ! & # - $ $
& # B % (" ! (0 $ ' % 6 ! " ,+
(& " - % (1 43 5 # & & "

(& # $ " (" ! & #
& " ! ((& ! ! ("
! . (& 7 % ! ("

%& # # (& # - ("
3 " ! & # (- !
! (" # & (,* " & (! %
(& " " " # & ! (% (
$ & ,* 0 % (1 (" (& % "
! " ! & 7 % ! (" & " " (& ! & ;
"

Entity

" " & "" (& # ("
& % $ - $ % " ! ! (" !
((& ; (6 " " #(

Resource

4 5 # # &) "" & " "

" (" (" " " ("
! " # " " " " " "
" $ " " " "

& (" " "
" " (

$ "" " &
" (-

& "" (" " " $ & & &
" B

```
address & mask == base
```

"

```
base & mask == base
```

"" " $ & "

Server States

& # - $ $ B %' . "
' /* %' . "$ # " -
" , $ &
" ' /* &
" $ & " $

Control Flow in the Server and Mandatory Driver Functions

6 main() 0dix/main.c1 8
InitOutput() 8 InitOutput()
" (6 " *screenInfo.screen[]* &
7 $ InitOutput() " B

Parse the xorg.conf file

" &

" 6 " # " "
/ " # " "
; " " " # ## & "

" # & " ; #

Initial processing of parsed information and command line options

" &

" # - % ! 6 6 " "
$ 6 "

Enable port I/O access

) " # 8 9: ## & 6 " 4
5 (" " (6 (;
6 "$, $ (- 8 " (" "
6 "$, $ $& # 8

" #& & "

General bus probe

" &

9: # 6 $ (! (#
(" $ ((" #
8 ! 6 ! # " " "
((& xf86GetPciVideoInfo()

```
pciVideoPtr *xf86GetPciVideoInfo(void);
```

,"
6 $ " " !
" " # "$ NULL

! " " $ " "6
NULL

((6 (- "

Load initial set of modules

" &

" &#" "" !
& &#" (# #"

#" "" " & %

#" " #" "(&
" ' #" ""
&

Register Video and Input Drivers

" &

D " #" ""6 " Setup "
" 6 xf86AddDriver() " A 6
$ # " " " & ; "
" & InitOutput() xf86AddDriver() "" (
xf86DriverList[] &

" #6 Identify()6Probe() "
AvailableOptions() & $ " A #"
0 " # " $ " $ ""1 "
$ - - $# & " &" &
; " " " %

6 xf86DriverList[] & " (" #6 "
" #" "

" " " " A Setup
xf86AddInputDriver() " A 6 $
" " " & ; " "
& InitInput() xf86AddInputDriver() "" (
xf86InputDriverList[] & 6 xf86InputDriverList[]
& " (" #

3 xf86DriverList[] " xf86InputDriverList[] & (
"(& "

" "6 ChipIdentify() "

void ChipIdentify(int flags);

" # " "
#6 ## & $ 6 "

#&
9: ! 01 "

```
void xf86PrintChipsets(const char *drvname, const char *drvmsg,
                       SymTabPtr chips);
```

" &$& " A !
" & # " #

Initialise Access Control

" &

! 0 !1 (& # "(&"
#& "$ (# ("
"0 1 0 " 1 " " " ("
(6 " (" " 4 # &5 " "
" # " ("

Video Driver Probe

" & ChipProbe()
" " " "

```
Bool ChipProbe(DriverPtr drv, int flags);
```

" &
"$ "(& "
& @6 PROBE_DEFAULT PROBE_DETECT
PROBE_DETECT " F F F (F #
" # " "
Probe() " (
" # (

(# " "
" (& xf86MatchDevice()
#(# " # # ## #(
" # "6
" FALSE ##" &

(" "(& " "
" # " "(("0-
" (
" (" " " " $& " (""
1 (# (# # (
" > " # "
" " "
(# " (# " &
6 - ("
& "6
"" xf86MatchPciInstances()

(" " & # !
" 6 " # & xf86MatchIsaInstances()
% 0 ! 1 " 0 ! 1
- " #
& D 6 " ("
$ xf86CheckPciSlot() " xf86ClaimPciSlot()
! " " xf86ClaimIsaSlot() % "
0 ! 1

(# (
"($
" $ ##" &
& (" $ xf86ConfigPciEntity()
! " xf86ConfigIsaEntity()
% 0 ! 1 (
# & $ D
6 xf86AddEntityToScreen()
xf86ClaimFixedResources() "
xf86SetEntityFuncs() " (" " 0
! 1

" "
$ " " 0 "
"6 " "# "
" (16 (# FALSE
" A # "(& "

" "
" 6 # FALSE

%
"$ "6 " (# 6
" " & ("
$ xf86ConfigIsaEntity()
% xf86ConfigPciEntity() !
%
& xf86AllocateScreen()
#& (" % &
- " "
" 4 "5

! # "$
" " # #
"$

"$ # " 6
"# " & "$

" ("" 6 "
" # ("

TRUE # "6 "
FALSE $

```
int xf86MatchDevice(const char *drivername,
                    GDevPtr **driversectlist)
```

```
        -    #    "    "
    driversectlist    "        #
   "    #                #
  (    #     "                "
                    -1

        "            xfree()       *driversectlist
  $          ""
```

```
ScrnInfoPtr xf86AllocateScreen(DriverPtr drv, int flags)
```

```
                    $ %
  xf86Screens[]   &          # &   "
  (&    " "          ChipProbe()
                $ &     " %
     scrnIndex6 origIndex6 module "   drv   "
     "                          drv     #
  "          &     &     "                4    5
                 "  "                 privates   "
    "0       "              "
     "1
          6          $
                &  "6  "    "(
   E

     #        6        %          (       "
xf86Screens[]   &6  "       "       "   "        (
  #"      $  %        " #  (      "       B
```

```
    "   ,
    "   /#
       " 0H1
       " 0H1
    " 0H1
    #"  0H1
     #
     (

    %
    '   ,
    +   ,
     #'
       &+
```

```
(*)          " $     %            "6  "        & (&
  "
```

$ % " # (" "
#B

%$ "
"> #
%
, " "

Matching Screens

" &

("6 $ (# #(%
" $ 6 "
" " #(#
@6InitOutput() *screenInfo.numScreens* 0 "

$ " % # ("B

%

Allocate non-conflicting resources

" &

3 " 6 # " # (
" " - !
("" 6 " & (!
6 - (6 " & ""
(

& " " EntityInit() " (6
& "

Sort the Screens and pre-check Monitor Information

" &

" # " ; "

#" # -" # A #
"#" "

PreInit

" &

% 6 (" Chip-
PreInit()

```
Bool ChipPreInit(ScrnInfoRec screen, int flags);
```

```
 /*      6        "(  "  "$
xf86SetOperatingState()

 "    # &  (    " "      &
    6   "     # "          "     $
  "  #    (    " "  (         "
      '                 xf86LoadSubModule()
xf86LoadDrvSubModule()        " (   "
 "#"  "  "   $     %
(       "   # &   "  # "  $
        $    &   " "  #    &6  "
xf86UnloadSubModule()        "(   "  "
     $           "       &    "
# "   (        "   -      # "  "
 "     " $     "  (# "     #    &
 )$    "   # "      " "

  ( -     %        "    " (    "


ChipPreInit()         FALSE $
    (     # $ & 0      " "  6     "
# " 6        "  # # &6   16  "                    TRUE
   (

       "                  ChipPreInit()
      TRUE6        &          ( ;
       "    #          -
0 -        1         (
    " ( "  #  " (&                 ChipPreInit()


 %           $                      ChipPreInit()      # "
# 6    InitOutput()     screenInfo.numScreens   0  "

     6       "      %       $  "  #  & (    "
      & # "  &6 (  # &     ;  " (&      &    ")
       "   # &          "  #           &   "
        "    $     &  ;

   $     "      %         " (    "      "
  #B


   #
   "   &
   "
     # 3
   (
   $            0I9(    &1
   # -          0I9(    &1
               0I9(    &1
    (3          09(    &1
     ##
   "    ,
```

7,
,,
8
J
" &D "
8@
J@
8
J
E #+ - "
"
"
"
! - 0 .' - ##(1

#"
-
#! - 0 ##(1
-KL 0 ##(1
" #
(3
#3
#! -
"

```
pointer xf86LoadSubModule(ScrnInfoPtr pScrn, const char *name);
  and
pointer xf86LoadDrvSubModule(DriverPtr drv, const char *name);
```

+ " #" " " "
" #" *name* (#" "
" " & $#"
" 6 $ (NULL
" " & " #" ""
$&6 # (" " "
xf86UnloadSubModule() $ "

```
void xf86UnloadSubModule(pointer module);
```

. " #" " (& *module mod-*
ule " (" & (&
xf86LoadSubModule() xf86LoadDrvSubModule()

Cleaning up Unused Drivers

- $ $ $ (6 "$ " (
" . "" 0 "#" &#& ""1 ""

Consistency Checks

((6 - # # 6 (# (
" 6 (# " # (& " # "
"6 $

Check if Resource Control is Needed

""
" & !$ #" ""

AddScreen (ScreenInit)

6 " - $ AddScreen() " #6
ChipScreenInit() # AddScreen() 8
$ *screenInfo.screen[]* & 0 - pScreen16 " " # (
ChipScreenInit() 6 $ *pScreen*
ChipScreenInit() FALSE6AddScreen() -1
$ " AddScreen() " & (
0 - # # &1
(# "(" " 3' '

```
Bool ChipScreenInit(int index, ScreenPtr pScreen,
                    int argc, char **argv);
```

"

*pScreen*6 (& " # (&
% # & - # 6
#(0 (6 16 "

" $ " (" "

(0 RAC_FB16
0 RAC_CURSOR16 $
0 RAC_VIEWPORT1 " #
0RAC_COLORMAP1 & ;
$ # (" (" " ' /*
"(! ("
& # # & " 0 *racMemFlags*
" *racIoFlags* " %
&1

& " # # & # # &

% " " ' # ("
("

" #" % A
vtSema " "(TRUE >
" "$ A

ChipScreenInit() 0 # & 1 "
% A *devPrivates* (& AllocateScreen-
PrivateIndex() " % % A *devPri-*
vates # " 6

AddScreen() & "6 $ % "
"B

%
#

ChipScreenInit() " *CloseScreen* " *SaveScreen*
" *pScreen* " *pScreen->CloseScreen* " ("
" A " 6 $ (" # Chip-
CloseScreen() # CloseScreen() $ "

Finalising RAC Initialisation

ChipScreenInit() ("6
" ! ; # # " " # $ "
; "(& # " "
" & " $ B

("(& & # "
"(& # & & $ (# - " "
" & 0 ' 1

< # - " 4" ("5 " ' /* $ (" &

= # - " 4 "5 $ & $
& " & " ' /*

? 4 "5 $ " (&" &
& & $ (" (" &

Finishing InitOutput()

InitOutput() "6 " (
" #"

Mode Switching

D %$ " "6 ChipSwitchMode() "0$ 1B

```
Bool ChipSwitchMode(int index, DisplayModePtr mode, int flags);
```

$ #" " "(&
index; $ #& " (">"

Changing Viewport

D ! , $ "6 ChipAdjustFrame() " 0$
1B

```
void ChipAdjustFrame(int index, int x, int y, int flags);
```

! $ " "(& *in-*
dex;

" (" # & #
$ $ # & ("
6 # 6
$ # ("
$ 0 <?(# " 1
A
" (* # " 6
" % A *frameX0*6
*frameY0*6 *frameX1* " *frameY1* " "
" 6 $
" # "$

VT Switching

D , $ "6 xf86VTSwitch() " xf86VTSwitch()
" $ B

'/ C ( )
' B
C " ( )

C %' .

C ChipEnterVT()

C ' /*

C " *!

C (# " #

C ' ("

C % (#
+',' B
C " *!

C %' .

C ChipLeaveVT()

C " ("

C ( )

C " ( )

```
Bool ChipEnterVT(int index, int flags);
```

" " #"
" $ 6 7D
6

% " " (
" #"M

```
void ChipLeaveVT(int index, int flags);
```

" " "
" 7D 6 "
" & #)

&6 ChipLeaveVT()#& # ## & 6 ChipEnterVT()
$ " # # "" &6 " " ## &
# " " # " 6 ChipEnterVT() $ " &
#(& A $ ## & "
$ A ModifyPixmapHeader() 6 $

```
(*pScreen->ModifyPixmapHeader)(pScrn->ppix,
                               -1, -1, -1, -1, -1, NewApertureAddress);
```

$ *ppix* " %
# " (& A SaveRestoreImage()
" A $ $ "

& #& $ ChipEnterVT() " ChipLeaveVT() &
" - # $ "

End of server generation

" 6 8 & ChipCloseScreen()
B

```
Bool ChipCloseScreen(int index, ScreenPtr pScreen);
```

" " " "
&

" " "
(& " / " "

% A *driverPrivate* " " ("
(""(& (;

% A *vtSema* " "(FALSE
" 7D ("

3 " " "
CloseScreen " (" *pScreen-*
*>CloseScreen*6 " "("
"

Optional Driver Functions

" (" # 8 9: ## & 6(

Mode Validation

D #" " "(& 8 9: ## & (
"6 (" - $ #" B

```
ModeStatus ChipValidMode(int index, DisplayModePtr mode,
                         Bool verbose, int flags);
```

! - "#" $ 6 "

#& #" & # " - "#"
(" #"A *Crtc** " *SynthClock* # 6 "
"&("> " 6" (6# " -#
"" " " #" & & #" "6
$ #" & #" # " (& xf86PrintModes()

" &#" "
6$ *flags* MODECHECK_INITIAL (; & "
&#" & (" 6$ *flags*
MODECHECK_FINAL " 6 #" $ & "
"" 6 % A *virtualX*6 *virtualY* " *display-*
Width " $ (#" (" "$ (
#" "

6 $ '!7'!2N / + "" - "
" " & #" (" "6 $ $
'!7'!2N /+ "" - #& #
#"

Free screen data

D " " # %
ChipFreeScreen() "$ " "

```
void ChipFreeScreen(int scrnindex, int flags);
```

& " " " #& (
" " "
ChipScreenInit() $ " "# &(
" "(& ChipPreInit()
$ " " *driverPri-*
*vate*6 " & 4 5 #" #&
"

Recommended driver functions

" (& " & & &
6 " "" & # & #
" $ > # 0 &1 "
"

Save

% " "(" # ChipScreenInit() "0 (&1
ChipEnterVT()

```
void ChipSave(ScrnInfoPtr pScrn);
```

% $ &(
(
& " "
"

Restore

" "(" # ChipLeaveVT() "
ChipCloseScreen()

```
void ChipRestore(ScrnInfoPtr pScrn);
```

" # .
& & " #"

Initialise Mode

" #" " (" # ChipScreenInit()6
ChipSwitchMode() " ChipEnterVT()

```
Bool ChipModeInit(ScrnInfoPtr pScrn, DisplayModePtr mode);
```

"$ " #"

Data and Data Structures

Command line data

```
! ## "          &    &  ( 6  "      "    (     (
   (       "   &  "       (    "
        ## "              "
  &       "       & (&    "          #             " "
      $         "          #     (

% #     #   B


   9:, (            (   &
   9:3             (    #    ## "
   9:              "     #    ## "
   9:D             $      #    ## "
   9:* ##           O 6 6(6P  ##    #     ## "
   9:                      #     ## "
   9:  (   &          (   &   #     ## "
  "    !   ,   !        #     ## "


 $    "    $            ## "       6 $ # &   "      -


     (     "     "   & #    (& "    $       $       B


            int xf86GetVerbosity();


                                 xf86Verbose


            int xf86GetDepth();


                        -depth  ## "
                 ## "   6         -1      "


            rgb xf86GetWeight();


                         -weight  ## "
                  ## "   6          {0, 0, 0}      "


            Gamma xf86GetGamma();


                        -gamma   -rgamma6-ggamma6-bgamma #
              # "                  ## "   6
              {0.0, 0.0, 0.0}      "
```

```
Bool xf86GetFlipPixels();

TRUE -flippixels #
# " 6 " FALSE $

const char *xf86GetServerName();

# 8 # ## "
```

Data handling

```
! " ( 6 " %
" " " #
" - $ (

( " & & ; "(& " 6 " 6 # "
9:

" " # " 6 " 6
" &6 # " # ScrnInfoRecs6
$ # ) " - " ScrnInfoRecs ; "
" " % 6 " " * 6 #
" 6 " " & "

8 9: ## & A " 0 "
1 " ScrnInfoRecs ( " ( 6 ScrnInfoRecs
" ( # 6 " ( & A Probe() "
PreInit() # ( " ( " " "
## " " " " " # ! " B

## " I I ( ")" "

# & "
( " & "

$ " 8 9: ## & 0 "
$ 1 "
6 # " " & D "
$ "" $ " % 6 " ( ( $
" " " E ) ( & %


" " ( ## " "$ %
" " " " " " 6 $
" % A driverPrivate " "
0 " ( - % " 1
$ " & & " " "A "

" # " " 0 # 6 8 # "
1 - " %
A privates "
```

("6 " " ((" "
$ (" # " # #6 " " xf86.h
" xf86str.h (" " " " xf86Priv.h "
xf86Privstr.h / "" ((" " 6 "
" "

Accessing global data

%# (# " #&
$B

```
Bool xf86ServerIsExiting();
```

TRUE "
" 6 " FALSE $

```
Bool xf86ServerIsResetting();
```

TRUE "
" 6 " FALSE $

```
Bool xf86ServerIsInitialising();
```

TRUE (
" 6 " FALSE
$

```
Bool xf86ServerIsOnlyProbing();
```

TRUE (& ## " $
"6 " FALSE $

```
Bool xf86CaughtSignal();
```

TRUE 6 "
FALSE $

Allocating private data

" " &#" #&
% 0 8 1 % 08 9: ## & 1 %
& 6 "%
#

% *devPrivates*" # (") "
$ # &" # ChipScreenInit() 6 "

#" " $& "(
"(& " A ChipCloseScreen() 6 "!
#" $ *devPrivates* & "(& AllocateScreenPrivateIndex()

```
int AllocateScreenPrivateIndex();
```

$ # *devPrivates* " & ScreenRecs " $ # *devPrivates* " & *devPrivates* " & . B

```
typedef union _DevUnion {
    pointer             ptr;
    long                val;
    unsigned long       uval;
    pointer             (*fptr)(void);
} DevUnion;
```

$ $ # (" &
(& ## & " "
$ " (
"

$ -1 $
$ "

% *privates* " 6 & "
(" " (" # ChipPreInit() 6
" #" " $&
"("(& " A ChipFreeScreen() 6 "
#" $ *privates* & "(&
xf86AllocateScrnInfoPrivateIndex()

```
int xf86AllocateScrnInfoPrivateIndex();
```

$ # *privates* " & ScrnInfoRecs " $ # *privates* & *privates* " & *DevUnion*B

```
typedef union _DevUnion {
    pointer             ptr;
    long                val;
    unsigned long       uval;
    pointer             (*fptr)(void);
} DevUnion;
```

locateScreenPrivateIndex()1 "

Keeping Track of Bus Resources

Theory of Operation

D Probe() Probe()

EntityInit()6EntityLeave() " EntityEnter()

EntityInit()6EntityEnter() " EntityLeave()

PreInit() " " - & (
"" Probe() (" " " - $ " #
&(" " ("" EntitySetup()
" " "" " "$ " (#(
PreInit() " %
"" ; 6 6 !
" "" # #&(# -"
4" ("5 4 "5 " ' /* "
(#" ScreenInit()6 " (

$ PreInit() ## " #
"" # " &
!$ #" "" ScreenInit() " " " $
" (" " ! (# (6
" # & ;
$ # (" ("" ' /* "(
! (" & ## & "

D ScreenInit() " ## $ " # $ "
; "(& # " "
" & " $ B

("(& & # "
"(& # & &$ (# -" "
& 0 IO MEM1

< # -" 4" ("5 " ' /* $ (" &

= # -" 4 "5 $ & $
& " & " ' /*

? 4 "5 $ " $ &" &
& & $ (" (" &

" # " $&
B

& ((& # ##
7 " & " &

< (&" $ #
&(" $ (
" #- " (" $
PreInit() % # " PreInit()
" $ (" #
" " & #
& ")) &

= " ($ " # ' "
(# " " (#($ "("
$ (" " " " $& & ; "
" %'. $& (# -" 4 "5 " ' /*

$ " "" ' /* $ "(&
" "(# -" "

% $ " 8 6 ##
# # #E #(" (" " !
D $ " "6 EnableAccess() "
(" EnableAccess() - "
> ((" "
(" 6 EnableAccess() " # $ & 0 MEM6
IO1 ; " "& (" # &
" (& (" 6 ("
("

D #" $, $ #" ## $
%'.

Resource Types

D #
"(& " "
' #&(" "

C & 6 "" ## &0 ResMem1) 0 ResIo16

C "" (-0 ResBlock1 0 ResSparse1 6

C

$ - $ B

C ResExclusive $ #& ("$ & " "

C ResShared $ (" (" " ("

& & & & ResAny
"" $ $ & "
& (" "
"" $ "" & 0 ResUnused1 " ' /*
0 ResUnusedOpr1 & (" 0 Enti-
tyInit()6 EntityEnter() " EntityLeave() "("$
ResInit # $ (- " #&(
"$ ResBios # ($
$ "(& & #3

% " " (,* " 9G ?)
common/xf86Resources.h

Available Functions

"" # # " "
"

Probe Phase

" " (" 6 &
" #6 ("
" ""

$ "" # "
" B

```
int xf86MatchPciInstances(const char *driverName, int vendorID,
                          SymTabPtr chipsets, PciChipsets *PCIchipsets,
                          GDevPtr *devList, int numDevs, DriverPtr drvp,
                          int **foundEntities);
```

" # ($! "
" " "
"" ChipProbe() "
! " & (" ! " $
" # *vendorID* " " *devList*
" *numDevs* & & " #
xf86MatchDevice()6 "
" " *PCIchipsets*
(" # ($! "
6 " A - "
"

D " " A # &
& # # & " " % " & "
& # "$ "
# # &

&# ("6
#"(& - "6
! " (" ! & #
chipsets " *PCIchipsets*
PCIchipsets " ! "
"(& " "(#
"$ &$! -1F *chipsets*
(# " A -
6 " "(# "$ NULL &
& $ " &
PCIchipsets " " "
" B 6 ! 6 ("
! "

$ " " ! $ #
" 6 #& *vendorID* 06 "
" $ 0 " W WQQW :1

' & " #(#" #
" & *foundEntities* !
6 - " "
" ' & "(&
"

#(#"
-1 "
" *foundEntities* & "("
(& " $ xfree() $ ""
$ E

```
int xf86MatchIsaInstances(const char *driverName,
                          SymTabPtr chipsets, IsaChipsets *ISAchipsets,
                          DriverPtr drvp, FindIsaDevProc FindIsaDevice,
                          GDevPtr *devList, int numDevs,
                          int **foundEntities);
```

```
        " #   ( $   %   "
   "       "      "
    ""                 ChipProbe()       "
      %   "        devList "   numDevs   &   &
      "  #              xf86MatchDevice()6 "
                "
    "       ISAchipsets   (      "  #
     ( $    "   A          -   "
                      FindIsaDevice  "     "
   "        (     "$   "
      -      "   $  $ "   "6 "
  -1      $ "   "

          "                &6
         -"                   chipsets    D
        &     6             FindIsaDevice
     "   "

  '  &  "   #(       # " #
     "     &            foundEntities
  -   ""          #      "
  '  &        "(&    "

                  #(      # "
  #                        -1  "
        "        foundEntities  &   "(   "
  (&   "   $        xfree() $          ""
       $                 E

$        # -                    (
"     B
```

```
Bool xf86ParsePciBusString(const char *busID, int *bus,
                           int *device, int *func);
```

```
  -       BusID    6  "
  # 6        !          bus6 device6 func
   "        #              "   (
  F ! B( B"   B   F $                    4( 5   6 4"   5
   "   4   5      "  #        FB   F    # &
  (  #   "6   "           # "   (  E  6
  (      A      "   F ! F    # &   (
   #   "       F * F      & ;
  F ! F              A     " ! 3   6
                   FALSE
```

```
Bool xf86ComparePciBusString(const char *busID, int bus,
                             int device, int func);
```

```
  !#          BusID    $   !        bus6 device6 func
       &#           TRUE     "6 "       FALSE   &
  " A
```

```
Bool xf86ParseIsaBusString(const char *busID);
```

```
  !#          BusID    $     % (
  0F % F   F % BF1     & #         TRUE      "6  "
  FALSE    & " A
```

```
Bool xf86CheckPciSlot(int bus, int device, int func);
```

```
  ! -      !             bus:device:func    (
    #"   6              FALSE6 "   $        TRUE
```

```
int xf86ClaimPciSlot(int bus, int device, int func, DriverPtr drvp,
                     int chipset, GDevPtr dev, Bool active);
```

```
            "    #  !   6
      "   &    " "         "
                  "      $ &
    "   &    "6           -1      #
       " $ &     "              xf86CheckPciSlot()
     "    TRUE     # !
```

```
Bool xf86IsPrimaryPci(void);
```

```
                     TRUE     # &  "   !
  "  6 "     FALSE   $
```

```
int xf86ClaimIsaSlot(DriverPtr drvp, int chipset,
                     GDevPtr dev, Bool active);
```

```
             &   "   &  "
  "                    "
    $ &     "   &   "
```

```
Bool xf86IsPrimaryIsa(void);
```

TRUE # & " %
0 !1" 6 " FALSE $

$ "" " B

```
ScrnInfoPtr xf86ConfigPciEntity(ScrnInfoPtr pScrn,
                                int scrnFlag, int entityIndex,
                                PciChipsets *p_chip,
                                resList res, EntityProc init,
                                EntityProc enter, EntityProc leave,
                                pointer private);

ScrnInfoPtr xf86ConfigIsaEntity(ScrnInfoPtr pScrn,
                                int scrnFlag, int entityIndex,
                                IsaChipsets *i_chip,
                                resList res, EntityProc init,
                                EntityProc enter, EntityProc leave,
                                pointer private);
```

"
(&6 " &
*Init*6*Enter* " *Leave* . &
" (" #) !
7 $ "" # & (
" * & ; "
% " *pScrn* #
NULL TRUE $
6 6 " " $ B

```
typedef void (*EntityProc)(int entityIndex,
                           pointer private);
```

& " & " "
(
" Probe() " "" ("
xf86ConfigIsaEntity() " xf86ConfigPciEntity()
#

$ #- (
" B

```
void xf86ClaimFixedResources(resList list, int entityIndex);
```

(
$ (" (" " $ $ "
##" & & $ "
" (((
Probe()

```
Bool xf86SetEntityFuncs(int entityIndex, EntityProc init,
                        EntityProc enter, EntityProc leave, pointer);
```

$ & *init*6*enter*6 *leave* $

```
void xf86AddEntityToScreen(ScrnInfoPtr pScrn, int entityIndex);
```

& "(& *entityIndex*$

PreInit Phase

"(" PreInit() " xf86GetEntityInfo() (' & & (" " - & " *resources* " " ((> " (0*resources* NULL16 " "" " $ "

```
EntityInfoPtr xf86GetEntityInfo(int entityIndex);
```

' & "(& *entityIndex* "' & "("$ xfree()$ ""

% "" # & B

```
Bool xf86IsEntityPrimary(int entityIndex);
```

TRUE & "(& *entityIndex* # &" &" 0 6 " (# " " #"1

```
Bool xf86IsScreenPrimary(int scrnIndex);
```

TRUE # & & "$ "(& *scrnIndex*

```
pciVideoPtr xf86GetPciInfoForEntity(int entityIndex);
```

," " & & ! " 6 NULL "

& B

```
resPtr xf86RegisterResources(int entityIndex, resList list,
                             int access);
```

list

NULL " #
& & $ - "
$ & " &
" " % & ! "
3& " ! " "
0ResShared1 " $ "
& " (& &
" # 0 # "
& (-
(# ; "
$ "
" # & ((#
" ! (
" (& "
xf86ReallocatePciResources() D
" 6 NULL

```
resPtr xf86ReallocatePciResources(int entityIndex, resPtr pRes);
```

- ! "
(" " NULL $
xf86RegisterResources() " (
" " $
(- 6
" (" "

$ "" (& B

```
resRange xf86GetBlock(long type, memType size,
                      memType window_start, memType window_end,
                      memType align_mask, resPtr avoid);
```

" (- E *size*
" & *type* $ " $ (" (& *window_start*
" *window_end* $ # "
align_mask & $
" ("" $ $ " $ ("
E & ResEnd $ (
"

```
resRange xf86GetSparse(long type, memType fixed_bits,
                       memType decode_mask, memType address_mask,
```

```
                                    resPtr avoid);

                    -          6 (    #
                "          "   ( -    7
                        (     "B                    address_mask
            $   # -   (      # -        ""  6
                decode_mask $   # -      (  $
              " " "  "             (
                "           " (
                " (                 "
                    $      E       &                    ResEnd
                 &  #  (    "
                "

% # ! "     ( -           &       " E    #
                  "       &        E  " # -
                  "       "            "
 ""      "  " " (&     "                          xf86FixPciResource()   (
  "  "   B


          Bool xf86FixPciResource(int entityIndex, unsigned int prt,
                                  CARD32 alignment, long type);


                        !                                    prt
              #          #(      ! (
                  "   (   " 0            0-56  "   6     3 %
            (      1     E       " (&     #
            %   !        "
              E    2^n6     #             #(
             ""      $  ( "  " "     "
                &  # -        "    "    &
            !      $               ResShared       (
            -   "   - $       "  #
                          " (    " (
            xf86RegisterResources()                    TRUE
            $          "


          Bool xf86CheckPciMemBase(pciVideoPtr pPci, memType base);


                    -      # # & (   ""
                "#         ! (   ""
                      ! "        #  &  "
              -         &    " " (   ""   0  6
              #       1 #                "
             "

  "   # &                          &
"  $          xf86SetAccessFuncs()B
```

```
void xf86SetAccessFuncs(EntityInfoPtr pEnt,
                        xf86SetAccessFuncPtr funcs,
                        xf86SetAccessFuncPtr oldFuncs);
```

$ B

```
typedef struct {
    xf86AccessPtr mem;
    xf86AccessPtr io;
    xf86AccessPtr io_mem;
} xf86SetAccessFuncRec, *xf86SetAccessFuncPtr;
```

```
 "            B     )    6
   ## &     "     #( "## &
 " )       ## &     " #( "
        "      ##      #
   ## &       (     " "
" & )    6     )         "
 #( "            #      #"
  )    (     " "  "  &  ## &
 " )    (     "
  "(    #                  NULL       "
  "  #          "    ""  $
      "      "      "  #
NULL  $  (   #"             "
(  ( "(
 $   $  ( " ( "   "   #&  (   #
            "        ""     #
     #
#&( "  ( "(&    ##
        #            "  &#
                  &      "
     &
```

```
 "                    $
xf86ChkConflict()    #&(  "B
```

```
memType xf86ChkConflict(resRange *rgp, int entityIndex);
```

```
        -                                  rgp
     "  &     $  $
        "6   ""
        "          E  $
```

```
 ' /*          &     "  "       (
$        xf86SetOperatingState()    B
```

```
resPtr xf86SetOperatingState(resList list, int entityIndex,
```

```
                                        int mask);

                    "
          "    ' /*                  list  "      $
          mask    (    "     #                 mask # &
                    ResUnusedOpr "  ResDisableOpr
              "(  "        A  "(&  "
          "    ' /*          " ""(&
          "   6 $          "
             " """    ' /*    /
                     #         ""
                        (      "
          A  "  "          B  "          C         &
          A  <  C  <  B   # &      &
          (A,B) (&          $                (A,C)  "   (C,B)

    $   $          ""        B

        void xf86RemoveEntityFromScreen(ScrnInfoPtr pScrn, int entityIndex);

                # & (   "   #      &  #
                 & # -          #
              &    "          ( "   " /
             # "          &

        void xf86DeallocateResourcesForEntity(int entityIndex, long type);

                 "                  &
              "         &  #      ( -
```

ScreenInit Phase

```
      ;  "              !    /
# (                        !       "
racIoFlags "  racMemFlags  "    %              &
 & $          #   ;         "         (
   "     &   ## &  " )          (      "   "
  rac/xf86RAC.h   &  B

RAC_FB           # (        0   "   $       1

RAC_CURSOR      !        0MMM  A#      $   "     %D
               "  "   $        $      " $ 1

RAC_COLORMAP      #

RAC_VIEWPORT              ChipAdjustFrame()

       "
```

Config file “Option” entries

" # " (
$ # B

F #F (

F #FF $ (&
F

" "(& 6 " "
""$ " "
" " " " # ; " 6
" " # " " & "6 ## &
" #&(" 6 - "#
"

6 $ " # " $ " "
" &

```
XF86OptionPtr xf86FindOption(XF86OptionPtr options, const char *name);
```

- " #6 "
" & #
NULL # "

```
const char *xf86FindOptionValue(XF86OptionPtr options, const char *name)
```

- " #6 "
"$ &
# # #
6 # & 0 ""1 " NULL
"

```
void xf86MarkOptionUsed(XF86OptionPtr option);
```

- " 6 "# -
"

```
void xf86MarkOptionUsedByName(XF86OptionPtr options, const char *name);
```

- " # "# -
& # #
"

/ 6 # " $ "

```
void xf86CollectOptions(ScrnInfoPtr pScrn, XF86OptionPtr extraOpts);
```

! #
"(& 0 *pScrn*1 " # "
pScrn->options ;
*pScrn->confScreen*6 *pScrn->display*6 *pScrn->moni-*
*tor*6 *pScrn->numEntities*6 " *pScrn->entityList*
" *extraOpts* # & & (
"" (#(" $
" " *extraOpts*6
" &6 % 6 # 6 "

```
void xf86ProcessOptions(int scrnIndex, XF86OptionPtr options,
                        OptionInfoPtr optinfo);
```

" #
& 0 *optinfo*1
# " *value* "
optinfo *found* "
TRUE $ $ &
"6 " FALSE $ *type* " "
" # " &
' $
0(& & # 1
-" " D # "$
" A # & "
"

/ 'B "(" A
#(- $ 0 6 # ChipProbe() 1
scrnIndex -1 "("

/ ' <B * Op-
tionInfoRecs " (& *optinfo*6 "
OptionInfoRecs 0 1 "(
6 &
" " "

" " $ B

```
typedef struct {
    double freq;
    int units;
} OptFrequency;

typedef union {
    unsigned long       num;
    char *              str;
    double              realnum;
    Bool                bool;
```

```
            OptFrequency        freq;
        } ValueUnion;

        typedef enum {
            OPTV_NONE = 0,
            OPTV_INTEGER,
            OPTV_STRING,  /* a non-empty string */
            OPTV_ANYSTR,  /* Any string, including an empty one */
            OPTV_REAL,
            OPTV_BOOLEAN,
            OPTV_PERCENT,
            OPTV_FREQ
        } OptionValueType;

        typedef enum {
            OPTUNITS_HZ = 1,
            OPTUNITS_KHZ,
            OPTUNITS_MHZ
        } OptFreqUnits;

        typedef struct {
            int                 token;
            const char*         name;
            OptionValueType     type;
            ValueUnion          value;
            Bool                found;
        } OptionInfoRec, *OptionInfoPtr;
```

```
OPTV_FREQ   (   "
 ;                      #
( $            #    " "
 #    (      F7EF6  F-7EF6  F-F6  F 7EF6  F F
  #          " $            "
freq.units6  "      "   ;  &      "
freq.freq D            "6            freq.units
     06  "   freq.freq      "

OPTV_PERCENT   (   "
   "       0   F<@RF1
       #( $            " "
        #   6      $      #


 &                &
$      "      "              value  "   found   "
   (&    xf86ProcessOptions()      $
       #   #  6  "      "
 &OPTV_STRING6  "              #
          "      " "                          % #
```

```
void xf86ShowUnusedOptions(int scrnIndex, XF86OptionPtr options);
```

$ #
A # -" "
"" $ " A
" $ " # &(" "
ChipScreenInit() 6 (& $ server-
Generation == 1

```
OptionInfoPtr xf86TokenToOptinfo(const OptionInfoRec *table,
                                 int token);
```

table$
- "# *token* NULL #
"

```
Bool xf86IsOptionSet(const OptionInfoRec *table, int token);
```

found " *table*
$ *token* "# *token* ("
& / &
OPTV_BOOLEAN6 A - "
FALSE
"

```
char *xf86GetOptValString(const OptionInfoRec *table, int token);
```

value.str "
table$ - "# *token* NULL
"

```
Bool xf86GetOptValInteger(const OptionInfoRec *table, int token,
                          int *value);
```

**value* *value.num* "
table $ *token* "# *token*
**value* & "$ # "
(& "$ "

xf86IsOptionSet()

```
Bool xf86GetOptValULong(const OptionInfoRec *table, int token,
                        unsigned long *value);
```

+- xf86GetOptValInteger()6
" "

```
Bool xf86GetOptValReal(const OptionInfoRec *table, int token,
                       double *value);
```

```
  +-     xf86GetOptValInteger()6
  value.realnum   "
```

```
Bool xf86GetOptValFreq(const OptionInfoRec *table, int token,
                       OptFreqUnits expectedUnits, double *value);
```

```
  +-    xf86GetOptValInteger()6
  value.freq "       "     ;  &
    "        "   " (&              expectedUnits
         $       $      "
   &       A       $  6                 expectedUnits
    "    "      $  "
    6                      1000     #"
  (     "        #          #  6
              F @@@@F  "     ".
  OPTUNITS_MHZ6         "           10
```

```
Bool xf86GetOptValBool(const OptionInfoRec *table, int token, Bool *valu
```

```
              "      -  (
  0OPTV_BOOLEAN1                              FALSE6
   #        $  A      $                    *value
     (        "   " (&       A
  /          value        "          TRUE
  #         TRUE    F F6 F&  F6 F  F6 F   F6   "
     #            FALSE    F@F6 F  F6 F   F6 F    F
     # (  $     F  F          # 6  "
  $         #  "       -"  "   " "
     (    $ &          *value       "$
      A          "  #  & (      "
     (
```

```
Bool xf86ReturnOptValBool(const OptionInfoRec *table, int token, Bool de
```

```
              "      -  (
  0OPTV_BOOLEAN1              6
    "            6   "
   "(&   def      "
   #          xf86GetOptValBool()
```

```
int xf86NameCmp(const char *s1, const char *s2);
```

```
          "(   "$    #
   #        $      "      $ -  -
```

strcmp()6(" 6 (6 "
4_5 " #
A "
& (" &$ $ &
; "

Modules, Drivers, Include Files and Interface Issues

/ 'B #

Include files

$ " & & ; "(& " " B

" " " B

"xf86.h"
"xf86_OSproc.h"
"xf86_ansic.h"
"xf86Resources.h"

D () (0 " " 1 " $ "
(""B

"compiler.h"

/ B " 6 # ("" "xf86_ansic.h"

" ! " B

"xf86Pci.h"

%D " B

"mipointer.h"

" # # (- " B

"mibstore.h"

" # # " " B

```
"micmap.h"

"       $#" 6   "   B

"vgaHW.h"

    ,*  7   #   #      "B

"xf1bpp.h"

    ,*  '*! :        "B

"xf4bpp.h"

  (  "B

#define PSZ 8
#include "cfb.h"
#undef PSZ

    (  :6 <?  =< $   (  "    #   B

"cfb16.h"
"cfb24.h"
"cfb32.h"

"    8 6   "   B

"xaa.h"
"xaalocal.h"

"      (#  6   "   B

"xf86fbman.h"

/ "  #"    "  "              "xf86_ansic.h"        $
 /%  !) (

  #"  #  /   "  & &  #   "   6      $ B

   "xf86Priv.h"
```

```
    "xf86Privstr.h"
    "xf86_OSlib.h"
    "Xos.h"
```

"" 6F 9:N (F# ("" & "" # &(&
F 9:N F

Offscreen Memory Manager

" ## & #& (" "(& 8 9:
(# ## & # 6"
#& 6 E " ## & $
0" - # xf86fbman.h1B

```
    typedef struct _FBArea {
        ScreenPtr    pScreen;
        BoxRec       box;
        int          granularity;
        void         (*MoveAreaCallback)(struct _FBArea*, struct _FBArea*)
        void         (*RemoveAreaCallback)(struct _FBArea*)
        DevUnion     devPrivate;
    } FBArea, *FBAreaPtr;

    typedef void (*MoveAreaCallbackProcPtr)(FBAreaPtr from, FBAreaPtr to)
    typedef void (*RemoveAreaCallbackProcPtr)(FBAreaPtr)

    FBAreaPtr xf86AllocateOffscreenArea (
        ScreenPtr pScreen,
        int width, int height,
        int granularity,
        MoveAreaCallbackProcPtr MoveAreaCallback,
        RemoveAreaCallbackProcPtr RemoveAreaCallback,
        pointer privData
    )

    void xf86FreeOffscreenArea (FBAreaPtr area)

    Bool xf86ResizeOffscreenArea (
 FBAreaPtr area
 int w, int h
    )
```

B

```
    Bool xf86FBManagerRunning(ScreenPtr pScreen);
```

("(& - " E" ## &#
(FALSE " (
$

xf86AllocateOffscreenArea() (" ; "#
*width*WS*Weight* 0 1 # " ## & *granularity*
" # # # *granularity*
ty & E # # &
MoveAreaCallback ("" & ;
$ # " *MoveAreaCallback* "
" " (## (*privData* "$ (" # A
" "$ (" ;
FBArea " *MoveAreaCallback* *RemoveAreaCallback*
"" " " " "("
$ $ & % #&(# "$ & ;
0 " A *RemoveAreaCallback*1 #" D "6
" $ & " $ " " "
$ 6(# (
&# $ ##" &

xf86AllocateOffscreenArea() NULL $ (
; " D ""6 " (" $
xf86FreeOffscreenArea()

xf86ResizeOffscreenArea() E FBArea
xf86ResizeOffscreenArea() TRUE E $
xf86ResizeOffscreenArea() FALSE6 FBArea #"
" E # A granularity6devPrivate6 "
MoveAreaCallback xf86ResizeOffscreenArea() " (& "
" $ E6 "("$
(

B

```
Bool xf86QueryLargestOffscreenArea(
  ScreenPtr pScreen,
  int *width, int *height,
  int granularity,
  int preferences,
  int priority
);
```

"" ; & $" " 3 (
& *preferences* ($ "
$ $" 6 "(" "$ " # $
3 (

```
FAVOR_AREA_THEN_WIDTH
FAVOR_AREA_THEN_HEIGHT
FAVOR_WIDTH_THEN_AREA
FAVOR_HEIGHT_THEN_AREA
```

priority $ B

PRIORITY_LOW

(- ($ &
A " &
3 $ RemoveAreaCallback
""

PRIORITY_NORMAL

(- ((
$ & # #
" & 3 $
" RemoveAreaCallback

PRIORITY_EXTREME

(- (3
A -""$ $ " ### &
" & #" " &
xf86PurgeUnlockedOffscreenAreas()

B

```
Bool xf86PurgeUnlockedOffscreenAreas(ScreenPtr pScreen);
```

"" # # " ## & $ #
(3

E 8 9: #(# "

```
Bool xf86InitFBManager(ScreenPtr pScreen, BoxPtr FullBox);
```

FullBox #(# $" #
& & ($ $" *pScrn->displayWidth* "
# & ($ " ## &6 $ 6 "
# (& # #

xf86InitFBManager() # ("(8 E" 8
A

"" $ " E #(
$ (

```
Bool xf86InitFBManagerRegion(ScreenPtr pScreen,
                             RegionPtr FullRegion);
```

xf86InitFBManagerRegion()6 - xf86InitFBManager()6" #
" (" (""
" xf86InitFBManagerRegion() $
((# "6 " ; "$ # #(
" " ## & 0 $ & " "
" " #(## &1

Colormap Handling

" & "" $ 8 9: ## &
& - # " 6 " & ; #
" " "$ $; " # & 6
" xf86HandleColormaps()

```
Bool xf86HandleColormaps(ScreenPtr pScreen, int maxColors,
                         int sigRGBbits, LoadPaletteFuncPtr loadPalette,
                         SetOverscanFuncPtr setOverscan,
                         unsigned int flags);
```

(" "
(" *pScrn->gamma* "#
("6 (&(& xf86SetGamma()
maxColors #(
sigRGBbits E (#
!A *loadPalette* " ""
" # "$ 6 "
" ("($ *setOverscan*
#& ("" $
" # "" +. "$ " (
"> " - (- (
setOverscan # "
"A # &(" " 9 *se-*
tOverscan "(NULL $ A ""
flags#&($ 0$ #&("
1B

CMAP_PALETTED_TRUECOLOR !
" " >

! &
" bpp > 8

CMAP_RELOAD_ON_MODE_SWITCH " #
& #"
$
$ "

"$ " #"
$ "
"$

CMAP_LOAD_EVEN_IF_OFFSCREEN " #

$ "
A ,!
"" $
$ "
(- 6 #"

$
$ " "
- -

& # & "
, & "
" $ $, "
6 $ 6 "
" "

```
void LoadPalette(ScrnInfoPtr pScrn, int numColors, int *indices,
                 LOCO *colors, VisualPtr pVisual);
```

LoadPalette() " "" "
"$ *colors* &
*3 # *indices*
" &
" " " (" " *num-*
Colors #(" (" "

```
void SetOverscan(ScrnInfoPtr pScrn, int overscan);
```

SetOverscan() " ""
overscan " " (" (6
& & +. # " $
(" &6 ($
" 0 & &
9(,* # (+. 1 A ; "
$ (

DPMS Extension

% " % "" 8 9: ## &
" " ($ # "6 " #
" % $ & ("
" 6 " $ & (6 $ %
"(& 0 $ 1

```
Bool xf86DPMSInit(ScreenPtr pScreen, DPMSSetProcPtr set, int flags);
```

" A % #
set - *pScrn->options*
F" # F 6 " $ # - %
((" *set*
" $ % 6 " "
; " *flags* &
"6 " "(0 &

6 " $ %
6 FALSE

% # " $ 6
"$ % "B

```
void ChipDPMSSet(ScrnInfoPtr pScrn, int level, int flags);
```

% "(& *level* , "
level DPMSModeOn6DPMSModeStandby6DPMS-
ModeSuspend6 DPMSModeOff " "
"extensions/dpms.h"

DGA Extension

8 9: * 0 * 1 (&
" #" " # *

```
Bool DGAInit(ScreenPtr pScreen, DGAFunctionPtr funcs,
             DGAModePtr modes, int num);
```

```
/** The DGAModeRec **/

typedef struct {
  int num;
  DisplayModePtr mode;
  int flags;
  int imageWidth;
  int imageHeight;
  int pixmapWidth;
  int pixmapHeight;
  int bytesPerScanline;
  int byteOrder;
  int depth;
  int bitsPerPixel;
  unsigned long red_mask;
  unsigned long green_mask;
  unsigned long blue_mask;
  int viewportWidth;
  int viewportHeight;
  int xViewportStep;
  int yViewportStep;
  int maxViewportX;
  int maxViewportY;
  int viewportFlags;
  int offset;
  unsigned char *address;
  int reserved1;
  int reserved2;
} DGAModeRec, *DGAModePtr;
```

num

mode

flags

DGA_CONCURRENT_ACCESS

DGA_FILL_RECT
DGA_BLIT_RECT
DGA_BLIT_RECT_TRANS

DGA_PIXMAP_AVAILABLE

(

"

$

(

#

DGA_INTERLACED "
DGA_DOUBLESCAN

"
" (

#"

imageWidth imageHeight "#
#(((&

pixmapWidth pixmapHeight "#
#(((&

bytesPerScanline #((&

byteOrder . & # *pScrn->imageByteOrder*

depth " #(
#"

bitsPerPixel #((
#"

*red_mask*6 *green_mask*6 *blue_mask* *3# - #"6 (

*viewportWidth*6 *viewportHeight* # (
#(. & *mode->HDisplay*
" *mode->VDisplay*

xViewportStep yViewportStep & "& $
"
#"

maxViewportX maxViewportY # ## $
"(& " #"

viewportFlags $ #& (A" B

DGA_FLIP_IMMEDIATE "
##"
$

DGA_FLIP_RETRACE "
$

offset #(
" 0@6@1
#"

address "" #(
"(& " "
"$ * N 8 N, +3+'

```
/** The DGAFunctionRec **/

typedef struct {
  Bool (*OpenFramebuffer)(
       ScrnInfoPtr pScrn,
       char **name,
       unsigned char **mem,
       int *size,
       int *offset,
       int *extra
  );
  void (*CloseFramebuffer)(ScrnInfoPtr pScrn);
  Bool (*SetMode)(ScrnInfoPtr pScrn, DGAModePtr pMode);
  void (*SetViewport)(ScrnInfoPtr pScrn, int x, int y, int flags);
  int  (*GetViewport)(ScrnInfoPtr pScrn);
  void (*Sync)(ScrnInfoPtr);
  void (*FillRect)(
       ScrnInfoPtr pScrn,
       int x, int y, int w, int h,
       unsigned long color
  );
  void (*BlitRect)(
       ScrnInfoPtr pScrn,
       int srcx, int srcy,
       int w, int h,
       int dstx, int dsty
  );
  void (*BlitTransRect)(
       ScrnInfoPtr pScrn,
       int srcx, int srcy,
       int w, int h,
       int dstx, int dsty,
       unsigned long color
  );
```

```
} DGAFunctionRec, *DGAFunctionPtr;
```

```
Bool OpenFramebuffer (pScrn, name, mem, size, offset, extra);
```

OpenFramebuffer() "
& " - $ (
(#(
% "
("(& %
(&

name # "
NULL
" NULL
#
" $ "
$ " &
& ## &

mem & ""
#(

size E #(
(&

offset & " 6
(

flags & "" #
#& " ! &6
& DGA_NEED_ROOT
" "

```
void CloseFramebuffer (pScrn);
```

CloseFramebuffer() # & #
" 0 1
" #(
" &

```
Bool SetMode (pScrn, pMode);
```

SetMode() " E
#" " *pMode* NULL6
" "
* #"

```
void SetViewport (pScrn, x, y, flags);
```

SetViewport() " #-
" (
" " (x,y)
#(*flags* &"
" B

DGA_FLIP_IMMEDIATE $
"
##"
&

DGA_FLIP_RETRACE $
"

6
(
"

(

(x,y) $ ("
" #6 $ 6 "
" "
"$ "
"(& *xViewportStep* "
yViewportStep #"

```
int GetViewport (pScrn);
```

GetViewport()
% ("
" $;
" 6 (E
% , $;
" 6((
" 6

```
void Sync (pScrn);
```

" &

" "
"

```
void FillRect (pScrn, x, y, w, h, color);
```

"
w × h " *(x,y)*

```
void BlitRect (pScrn, srcx, srcy, w, h, dstx, dsty);
```

" &
w × h " *(srcx,srcy)*
(dstx,dsty) $
" " &"

```
void BlitTransRect (pScrn, srcx, srcy, w, h, dstx, dsty, color);
```

#
3
" -& *color*
"(- "

The XFree86 X Video Extension (Xv) Device Dependent Layer

8 9: 8 ," ' $ $ " &
" 4 5 " " $(3&" 6
" " ((8 & (E"
" E 8 & (& # XF86VideoAdaptorRecs
" (" " # " XF86VideoAdaptorPtr
$ B

```
Bool xf86XVScreenInit(ScreenPtr pScreen,
                      XF86VideoAdaptorPtr *adaptPtrs,
                      int num);
```

" 6 $ " " (XF86VideoAdaptorRecs $ "
(6 " " xf86XVScreenInit() " # " " #&
E ## 6 8 & "
D "$" $(# " $ (" &
("$ # "

8 9:," " B

```
typedef struct {
 unsigned int type;
 int flags;
 char *name;
 int nEncodings;
 XF86VideoEncodingPtr pEncodings;
 int nFormats;
 XF86VideoFormatPtr pFormats;
 int nPorts;
```

```
 DevUnion *pPortPrivates;
 int nAttributes;
 XF86AttributePtr pAttributes;
 int nImages;
 XF86ImagePtr pImages;
 PutVideoFuncPtr PutVideo;
 PutStillFuncPtr PutStill;
 GetVideoFuncPtr GetVideo;
 GetStillFuncPtr GetStill;
 StopVideoFuncPtr StopVideo;
 SetPortAttributeFuncPtr SetPortAttribute;
 GetPortAttributeFuncPtr GetPortAttribute;
 QueryBestSizeFuncPtr QueryBestSize;
 PutImageFuncPtr PutImage;
 QueryImageAttributesFuncPtr QueryImageAttributes;
} XF86VideoAdaptorRec, *XF86VideoAdaptorPtr;
```

' " $ $ 89:," " " $B

type (& $ A"

XvInputMask XvOutputMask
" $("
D "$A
XvInputMask
" "
"
" $(
XvOutputMask "
"
" #
" $(

XvVideoMask XvStillMask "
XvImageMask "
" 6 #
#
&

XvWindowMask XvPixmap- "
Mask & " $(
" (
"
6
"
" " XvPixmap-
Mask "

flags ! &6 $ " "B

VIDEO_NO_CLIPPING "
" " "

" $

4 5 ; $

" # " (& *drw_x*6
*drw_y*6 *drw_w* " *drw_h*
("
& 4* 5
; 7 "$
(
4* 5 # &

"

VIDEO_INVERT_CLIPLIST "
" " ;

$ (
"
$
(

VIDEO_OVERLAID_STILLS # # %
"$ "
" &
($-$ "
A $
"

D
6 % ," $ (
"$ "
"
"
(

VIDEO_OVERLAID_IMAGES % #
VIDEO_OVERLAID_STILLS
(#

VIDEO_CLIP_TO_VIEWPORT "
" "
" (
" (

$ $
#
" -

name #

nEncodings pEncodings #(" " (
" 8 9:, " ' " &
8 9:, " ' " " ("
" & 8 # " (
" # " F8,N *'F " $ " "
" & # ## E #
"

nFormats pFormats

#(# " (
" 8 9:, " # &
8 9:, " # " ("

nPorts pPortPrivates

#(#("
$ " "
& & # 6 "
" ((" # $
" $ # *pPortPrivates* &
A " $
(" " & # ; "
" # - " "
$ # & (# & 6
(" - $ $;
" " ## &6 $
" # (
86 "
" (" &

nAttributes pAttributes

#((E " (& "
" & 8 9: (
8 9: (" ("

nImages pImages

#(8 9: # " (&
" " &
8 9: # 8 9: # " ("

PutVideo PutStill
GetVideo GetStill
StopVideo SetPortAttribute GetPortAttribute
QueryBestSize PutImage
QueryImageAttributes

" 8 I"
6 *data* " "
" ("
(" ; " "
$ " B

*PutVideo*6*PutStill* " # *PutImage* " *QueryImageAttributes* ; " $ " & " XvInputMask

< *GetVideo* " *GetStill* ; " $ " & " XvOutputMask

= *GetVideo* " *PutVideo* ; " $ " & " XvVideoMask

? *GetStill* " *PutStill* ; " $ " & " XvStillMask

G *PutImage* " *QueryImageAttributes* ; " $ " & " XvImageMask

D *QueryImageAttributes*6 " Success $ # " & & XvBadAlloc

$ *QueryImageAttributes* E
8 # ; "

VIDEO_NO_CLIPPING 6 clipBox-
es # & ("(& " ClipBoxes X-
Y (" " " "
3 (
" # "(& drw_x6drw_y6drw_w
" drw_h *) (
" 6 "
" # & $ (" "
" VIDEO_INVERT_CLIPLIST6clipBox-
es $ " # $
(" (

```
typedef  int (* PutVideoFuncPtr)( ScrnInfoPtr pScrn,
               short vid_x, short vid_y, short drw_x, short drw_y,
               short vid_w, short vid_h, short drw_w, short drw_h,
               RegionPtr clipBoxes, pointer data );
```

" " " - (
vid_w (& *vid_h* *(vid_x,vid_y)* #
" # " " *drw_w* (&
drw_h *(drw_x,drw_y)* 6
& (
"$ " ("$
6 " "$ $ (
(" & " ("
" "> " 4 (65
& $ "
" $ 6 "
> "

```
typedef  int (* PutStillFuncPtr)( ScrnInfoPtr pScrn,
               short vid_x, short vid_y, short drw_x, short drw_y,
               short vid_w, short vid_h, short drw_w, short drw_h,
               RegionPtr clipBoxes, pointer data );
```

PutVideo " "
& # # #

```
typedef int (* GetVideoFuncPtr)( ScrnInfoPtr pScrn,
              short vid_x, short vid_y, short drw_x, short drw_y,
              short vid_w, short vid_h, short drw_w, short drw_h,
              RegionPtr clipBoxes, pointer data );
```

PutVideo "
" # " " "
" (; ""#

$ " # ;
"

```
typedef int (* GetStillFuncPtr)( ScrnInfoPtr pScrn,
              short vid_x, short vid_y, short drw_x, short drw_y,
              short vid_w, short vid_h, short drw_w, short drw_h,
              RegionPtr clipBoxes, pointer data );
```

GetVideo "
" & # #
#

```
typedef void (* StopVideoFuncPtr)(ScrnInfoPtr pScrn,
                                  pointer data, Bool cleanup);
```

" " " " &
" " (" "
cleanup " " " (
"(; " (
,
" "" & ## &
0 &1(" "
cleanup 6 " ("
& " # $ "$6
" " $ - &(" "
" " &
"$

```
typedef int (* SetPortAttributeFuncPtr)(ScrnInfoPtr pScrn,
                           Atom attribute,INT32 value, pointer data);
```

```
typedef int (* GetPortAttributeFuncPtr)(ScrnInfoPtr pScrn,
                           Atom attribute,INT32 *value, pointer data);
```

#& (6
6 (8 "
((& " (
0 #1 % ; "
" ## "" "
" $ (# "
8 (&" B

XV_ENCODING
XV_HUE
XV_SATURATION
XV_BRIGHTNESS
XV_CONTRAST

(" #& E # & # $
; " (- $ (& "
" BadMatch XV_ENCODING (
"" &$ " "
"(0 "
8 9:," ' " ($1 ;
" " "6 " "
XvBadEncoding " " "
BadValue # & (" Success "(
" $

```
typedef void (* QueryBestSizeFuncPtr)(ScrnInfoPtr pScrn,
               Bool motion, short vid_w, short vid_h,
               short drw_w, short drw_h,
               unsigned int *p_w, unsigned int *p_h, pointer data);
```

QueryBestSize " $ $& ; &
$ " "# $ " " (
& $; *vid_w* (& *vid_h*
# " #(" *drw_w*
(& *drw_h* % "
"$ $ (("#
&6 # " "

```
typedef  int (* PutImageFuncPtr)( ScrnInfoPtr pScrn,
               short src_x, short src_y, short drw_x, short drw_y,
               short src_w, short src_h, short drw_w, short drw_h,
               int image, char *buf, short width, short height,
               Bool sync, RegionPtr clipBoxes, pointer data );
```

PutStill
" (" " & ###
& (*buf* " # " "
(& *image*" " E
width(& *height* *sync* .' " "
"
" # *buf* $ *sync* .' "
" *buf* ("6 "6
#" "

```
typedef  int (* QueryImageAttributesFuncPtr)( ScrnInfoPtr pScrn,
                              int image, short *width, short *height,
                              int *pitches, int *offsets);
```

" " & $"
image E *width*(& *height* "

(" % # # & E " "$ "
" "" *pitches* " *off-*
sets /.++ E ## & ; "
"(& *width* "
height ; " # ("(&
" # # 0 #
" 1 *pitches*
" *offsets* /.++6 $ (& $
& #
image # " " & 0
(& 1 $
0 (& 1 # (
#

8 9:," ' " B

```
typedef struct {
 int id;
 char *name;
 unsigned short width, height;
 XvRationalRec rate;
} XF86VideoEncodingRec, *XF86VideoEncodingPtr;
```

8 9:," ' " $ "
" " >
" A (6 " $ $ (
"(& XvQueryEncodings *id* "
" (; " $
; " XV_ENCODING (

8 9:," # B

```
typedef struct {
 char  depth;
 short class;
} XF86VideoFormatRec, *XF86VideoFormatPtr;
```

$ " $(
depth " 0 (1 *class*
TrueColor6 DirectColor
PseudoColor E " $
"

8 9: (B

```
typedef struct {
 int   flags;
 int   min_value;
```

```
 int   max_value;
 char  *name;
} XF86AttributeListRec, *XF86AttributeListPtr;
```

' " # & & "
(! & " "
XvGettable " XvSettable $ # & (A"
" (4 (5 4
(5 (& min " max " &
" Name " (
((& #

8 9: # B

```
typedef struct {
 int id;
 int type;
 int byte_order;
 char guid[16];
 int bits_per_pixel;
 int format;
 int num_planes;

 /* for RGB formats */
 int depth;
 unsigned int red_mask;
 unsigned int green_mask;
 unsigned int blue_mask;

 /* for YUV formats */
 unsigned int y_sample_bits;
 unsigned int u_sample_bits;
 unsigned int v_sample_bits;
 unsigned int horz_y_period;
 unsigned int horz_u_period;
 unsigned int horz_v_period;
 unsigned int vert_y_period;
 unsigned int vert_u_period;
 unsigned int vert_v_period;
 char component_order[32];
 int scanline_order;
} XF86ImageRec, *XF86ImagePtr;
```

8 9: # " ($ " " "
& " $ B

id ; "
"
. !!
$ (

type XvRGB XvYUV

byte_order LSBFirst MSBFirst

guid * (& . ;
D
(6 " (
/.++

bits_per_pixel #((- 0(
& "1 (&
/ # #
$ (
0 @T1 #(# & (
" " N" $ N

format XvPlanar XvPacked

num_planes #(
" (
- " #

depth (
*3 # 0
" # # 1

*red_mask*6 "6 " ((# -
*green_mask*6 - " *3 #
blue_mask

*y_sample_bits*6 &6 " # E 0 (1
*u_sample_bits*6
v_sample_bits

*horz_y_period*6 &6 " # "
*horz_u_period*6 E "
horz_v_period

*vert_y_period*6 &6 " # "
*vert_u_period*6 "
vert_v_period

component_order .
" #
" $ - " #
#
" . "
=< (&
" (/.++

scanline_order XvTopToBottom XvBot-
tomToTop

% # # 0 # J.,
# 1 # & (# & " " (& #

(6 "6 $ (6 " "
"

The Loader

" (#" " "
(" "" $ B & (8 9: ##
& 6 " (#"

Loader Overview

" (" #" (> # 6 " - $
" # (" 2 $ " $ (> #
("" " " $ " # #- (
" % " " #" 0 ! . & 1 ""
6 " " #" % "" dlopen(3)
$ (% #" # " " 6 " #
dlopen() # & # # #"
& % dlopen() #" " # & #
"" #

%&#(" (& " 0 (8 1 #"
" # " # # & &#(& " (
#" &#(" "#" " #" 6 "
8 " (; " - " &#(
& #6 " (- " &#(("
(& " & & " $ &#(&
" "$

/ 'B / $ % ! $ " "
" 6 # $ $ " (
"

Semi-private Loader Interface

$ # " (8 9:
&

```
void LoaderInit(void);
```

LoaderInit() "6 "
(" (& "
(
" &#(6 " & # " (
"

```
void LoaderSetPath(const char *path);
```

LoaderSetPath() "
#" # ("

(#" $ & &
#" *path*# (
"(
" " ("($ 6
(& ("

```
pointer LoadModule(const char *module, const char *path,
                   const char **subdirlist, const char **patternlist,
                   pointer options, const XF86ModReqInfo * modreq,
                   int *errmaj, int *errmin);
```

LoadModule() " #" "
module #" " 6 "#&
(" " ;
""#" #" # *module*
& #" A #6 $ " A
&" & # 6 & (>)
(& ! & #
") # " #
B

path ## "
#" D NULL6
" "

subdirlist NULL # " (
" D NULL6
" (" 0
stdSubdirs loadmod.c1 "
(" *sub-*
dirlist $ DEFAULT_LIST
#- (#
" " %(
" # # (6 "#
"../" &
; # 6 "

patternlist NULL # "
%8 "
#" # $
#" # ' "
& (
" #"
D NULL6 " (
" 0 stdPatterns
loadmod.c1 " (
" *patternlist* $
DEFAULT_LIST #-
(# "
"

options # "
$& "" #" A Setup-

```
Proc      0        1
 #      #  &            NULL  #   "
        6  "#  (       "
   $ & (& # "     " " "    & (&
  8   9:  ##   &   7 $   6
# & (   "
 #

D    "          4     5      # "   0# "
     " A         " "   &
   6     #           "
(  &1         #     (
   EXTERN_MODULE         "
  >    # "  $    "   A   "
    " "   &

modreq          8 9: " ;   H
                ) 3 )  "     #
 ;  #      -    $ &   " "
# "          #
        $     "      &
 # "          & )    (
                SetupProc

    XF86ModReqInfo       "   "
   $ B


typedef struct {
 CARD8        majorversion;  /* MAJOR_UNSPEC */
 CARD8        minorversion;  /* MINOR_UNSPEC */
 CARD16       patchlevel;    /* PATCH_UNSPEC */
 const char * abiclass;      /* ABI_CLASS_NONE */
 CARD32       abiversion;    /* ABI_VERS_UNSPEC */
 const char * moduleclass;   /* MOD_CLASS_NONE */
} XF86ModReqInfo;


     #            #  "
       ;        #
   # "  A   8 9: "  ,
  " 0$     "   ( " (  $1
        ##     (     "
4" A     5
  "     #     # "
 $ B

majorversion        #
               # "  A  # >
                  &

minorversion      # "  A   #
                    #    (

                  #       &
```

#" *majorversion*
" "#

patchlevel #" A
(
#
& #"
minorversion
" "#

abiclass % # #
#" A (

abiversion (
$
#" A (
0# > ; 6 #
" 1

moduleclass % # #
#" A #"

errmaj ("
#> "
D "'6 **errmaj* "
$ LoadModule()

errmin +- *errmaj*6 (#
"

```
void UnloadModule(pointer mod);
```

" #" " (&
" #" "#" ""
& # (" " & "
#" "#" 0 6
("'$ LoadSubModule() " ("(
$1

Module Requirements

" # " # (# "6 "#&
& " & F F "F "$F 0 $
" SetupProc " TearDownProc1

#" # " 89: " , 6 $
" " $B

```
typedef struct {
```

```
    const char * modname;       /* name of module, e.g. "foo" */
    const char * vendor;        /* vendor specific string */
    CARD32       _modinfo1_;    /* constant MODINFOSTRING1/2 to find */
    CARD32       _modinfo2_;    /* infoarea with a binary editor/sign tool */
    CARD32       xf86version;   /* contains XF86_VERSION_CURRENT */
    CARD8        majorversion;  /* module-specific major version */
    CARD8        minorversion;  /* module-specific minor version */
    CARD16       patchlevel;    /* module-specific patch level */
    const char * abiclass;      /* ABI class that the module uses */
    CARD32       abiversion;    /* ABI version */
    const char * moduleclass;   /* module class */
    CARD32       checksum[4];   /* contains a digital signature of the */
                                /* version info structure */
} XF86ModuleVersionInfo;
```

" " $B

modname #" A # " & & #
6(" #&(#" " ;
#" A

vendor #" " " # &

*_modinfo1_* " " ("
(& " $& ("
MODINFOSTRING1

*_modinfo2_* " " " ("
(& " $& ("
MODINFOSTRING2

xf86version 8 9: $ #" $ # "
& #)" "
(" XF86_VERSION_CURRENT6 $ " "
xf86Version.h

majorversion #" #> #" $
" # # & # 6 #>
" & 0(# "1$ 3 # (
" "6 3 # # "

minorversion #" # #" $
" # # & # 6 #
" & 0(# "1$ 3 ""
#" (-$ " # ($& "(E $
#> "

patchlevel #" "
$ $ #" $ 3 6
" "(E $ # "

abiclass 3 #" ; "
& (& " " $ 0 &1
8 A (3 #" 6 "
& 3 3 3 &" " B

```
ABI_CLASS_NONE

ABI_CLASS_ANSIC        & ;      /% !

ABI_CLASS_VIDEODRV     ;       " "    3

ABI_CLASS_XINPUT       ;     8   "    3

ABI_CLASS_EXTENSION    ;           #"   3

ABI_CLASS_FONT         ;        #"   3
```

abiversion (#" ;
#> " # # #> #
" # # ($ ""
(& #" , " (
& " " B

```
ABI_ANSIC_VERSION
ABI_VIDEODRV_VERSION
ABI_XINPUT_VERSION
ABI_EXTENSION_VERSION
ABI_FONT_VERSION
```

moduleclass # ("6 " &
#" 3 ; # 6
" " ; " " 3 6 #"
; " " 3 " " " (
#" $ #" ! & " "#"
B

```
MOD_CLASS_NONE
MOD_CLASS_VIDEODRV
MOD_CLASS_XINPUT
MOD_CLASS_FONT
MOD_CLASS_EXTENSION
```

checksum / & "

#" # 6 " SetupProc " TearDownProc
& "(& " (& " (> #" "
F#" # " F6 $ F#" #F # #"
" # " (> 6 " # (" "$ (6
(# # "6 " $ & B

```
typedef struct {
    XF86ModuleVersionInfo *     vers;
    ModuleSetupProc             setup;
    ModuleTearDownProc          teardown;
} XF86ModuleData;
```

(" & "
8 9: " , $ # 6 " "
(" NULL $; " # " "

```
typedef pointer (*ModuleSetupProc)(pointer, pointer, int *, int *);

typedef void (*ModuleTearDownProc)(pointer);

pointer SetupProc(pointer module, pointer options,
                  int *errmaj, int *errmin);
```

D " "6 "(& "
& " #" #" "
$ & " "#" 6 "#&("(&
SetupProc " ;
# #
$ " LoadModule() 0 LoadSub-
Module()16 " " (" (D Setup-
Proc # NULL
" - 6 " NULL " #"
" LoadMod-
ule() SetupProc" " (
" $ #" ""6 ""
TearDownProc6 " TearDown-
Proc " $ ("

D #" ""# # 6 SetupProc
" # ""

```
void TearDownProc(pointer tearDownData);
```

D " "6 "$ "
" #" *tearDownData* #
SetupProc() $ "$
#" $ ""
(#" ""0 # 6
(& " 1

Public Loader Interface

$ + " (& 6 "
#& (" #$ #"

```
pointer LoadSubModule(pointer parent, const char *module,
                      const char **subdirlist, const char **patternlist,
                      pointer options, const XF86ModReqInfo * modreq,
                      int *errmaj, int *errmin);
```

- LoadModule() "
(" (6 #" ""
" " #" *parent*
#" A " " ""
$ # & ""$
#" "" "
#& (" #"
" #" ""$
" "$ Load-
erSetPath()

```
void UnloadSubModule(pointer module);
```

" #" $ " *module*
#" " 6 &
"" - UnloadModule()6
" "#"

```
pointer LoaderSymbol(const char *symbol);
```

"" &#($
symbol #&(" #"
& $ - $ #

```
char **LoaderlistDirs(const char **subdirlist,
                      const char **patternlist);
```

NULL # "
#" # #" " "
#" *subdirlist* " *patternlist*
" (" (6 " ("
" # "
#" "6 NULL
" "("(& Loader-
FreeDirList() $ ""

```
void LoaderFreeDirList(char **list);
```

#" " (& Load-
erlistDirs()

```
void LoaderReqSymLists(const char **list0, ...);
```

$; " &#
($ " # & "(&
LoadSubModule() & &#("

$ & " ("$ LoaderCheckUn-
resolved() " LoaderCheckUnresolved()
$ - #
NULL # " &#("
" "(& NULL

```
void LoaderReqSymbols(const char *sym0, ...);
```

- LoaderReqSymLists()
&#(&#(
$
("6 & $
" " # "
" "(& NULL

```
void LoaderRefSymLists(const char **list0, ...);
```

$ (&
" &#($ " D LoaderCheckUn-
resolved() $ A $ &#
(" $ & & $
" ; " &#(-
NULL # " &#("
" "(& NULL

```
void LoaderRefSymbols(const char *sym0, ...);
```

- LoaderRefSymLists()
&#(&#(
$
("6 & $
" " # "
" "(& NULL

```
int LoaderCheckUnresolved(int delayflag);
```

- " &#(
$ " &#(
("$ LoaderRefSymLists()6 "#
"##& (# &
" &#("
("$ LoaderReqSymLists() Load-
erReqSymbols() E
&#("
E 6 "

delayflag # " # & (
LD_RESOLV_IFDONE

```
LoaderErrorMsg(const char *name, const char *modname,
               int errmaj, int errmin);
```

" "
4 " " # " 5 6 # " # *mod-*
*name*6 # *errmaj* 6 "
errmin *name* NULL6 "
" & # 0 $ "(& 4B5

Special Registration Functions

" # # #"
& (# " "

```
void LoadExtension(ExtensionModule *ext);
```

& "
"(& *ext* ' " " " B

```
typedef struct {
    InitExtension       initFunc;
    char *              name;
    Bool                *disablePtr;
    InitExtension       setupFunc;
} ExtensionModule;
```

```
void LoadFont(FontModule *font);
```

&
#" " "(& *font* "
" " B

```
typedef struct {
    InitFont    initFunc;
    char *      name;
    pointer     module;
} FontModule;
```

Helper Functions

" (4 5 " " # " D
" " ; " & (" " 4 # 5 6
& " # &
" (

Functions for printing messages

```
ErrorF(const char *format, ...);
```

```
        (          $
  0 &    & "    ")       1 ," "      "
     &   "       "    &          #
       "      "   ("(  $
            #    $   "(     "
```

```
FatalError(const char *format, ...);
```

```
         #     "      8      (
      "    & (   "$      "  "   6   #
      "      "(    "(&
     "             $        &
  "  "  $      "        6      (   "
  $    "             "  "       "
```

```
xf86ErrorF(const char *format, ...);
```

```
       -      ErrorF()6         #       &
     "$     8    A   (   &
  "    0    11         #        #
       "$             "$                          -qui-
  et     &    &        $  "   & (   "
        #       "$        #
      "      "   ("(  $
```

```
xf86ErrorFVerb(int verb, const char *format, ...);
```

```
  +-   xf86ErrorF()6      # ## (   &
     $    #      (    "
    &           verb     E  #     #
   $ &     "                          1   (   "
     #    $  "   #  &   (    ""6 (  $
  #   (     $  "       ( #
```

```
xf86Msg(MessageType type, const char *format, ...);
```

```
        -      xf86ErrorF()6          #
       "$    # -  "   #  "(&
  type    # -     "    "      &    #
    0$    6    6   ("    6       6   1
  /         xf86Verbose        "   #
   &    X_ERROR
```

- B

X_PROBED , $ ("

X_CONFIG , $

X_DEFAULT , "

X_CMDLINE , $ ## "

X_NOTICE /

X_ERROR ' #

X_WARNING D #

X_INFO # #

X_NONE /

X_NOT_IMPLEMENTED #
& & # # "

```
xf86MsgVerb(MessageType type, int verb, const char *format, ...);
```

+- xf86Msg()6($ (&
&

```
xf86DrvMsg(int scrnIndex, MessageType type, const char *format, ...);
```

- xf86Msg() " A #
0 *name* " % 1 $" (&
scrnIndex " $
"("(& " " #
& " $ ") #
scrnIndex 6 (
& - xf86Msg()

/ 'B &(" %
" *name* " (" / # &6
("('/
ChipProbe() 6 xf86Msg()6
" " A # & / #
(("

```
xf86DrvMsgVerb(int scrnIndex, MessageType type, int verb,
                               const char *format, ...);
```

+- xf86DrvMsg()6($ (&
&

Functions for setting values based on command line and config file

```
Bool xf86SetDepthBpp(ScrnInfoPtr scrp, int depth, int bpp,
                     int fbbpp, int depth24flags);
```

*depth*6*pixmapBPP* " *bitsPer-Pixel* " % " # " & $ " (" # # $ 6 " " & (# ")(" # & (" & & # Chip-PreInit()

; *confScreen* " % (" " (& 8 9: ## & ChipPreInit()

" B

depth " A " " " E 6 "

bpp % # 6 (# (

fbbpp % # 6 (#((

depth24flags "
<?)=<(" $
($ " #(
" # # "
" "
$ 6 " # # & (
" B

NoDepth24Support / " <?
#
"

Support24bppFb <?(
#(
"

Support32bppFb =<(
#(
"

SupportConvert24to32 !
<?(
#
=<((

SupportConvert32to24 !
=<(
#
<?((

ForceConvert24to32 <?(
#
=<((

ForceConvert32to24 =<(
#
<?((

" 6 6 " "
" " " #
" "(" -
Chip-
PreInit() " FALSE

& ")(6
(0 " 1 "

" " *depth24flags*
(# ; # "
& (# &
" # $ " *depth24flags*

6 TRUE
" FALSE

$ " % "
(& B

*depth*6 *bitsPerPixel*6 *display*6 *im-*
*ageByteOrder*6 *bitmapScanlinePad*6
*bitmapScanlineUnit*6 *bitmapBitOrder*6
*numFormats*6 *formats*6 *fbFormat*

```
void xf86PrintDepthBpp(scrnInfoPtr scrp);
```

(" " "
("("
xf86SetDepthBpp()

```
Bool xf86SetWeight(ScrnInfoPtr scrp, rgb weight, rgb mask);
```

*weight*6 *mask*6 *offset* " *rgb-*
Bits " % $ " # & (
" & & ChipPreInit()
" WIW9(

; *depth* " *display* "
% ("

" B

weight " A "" $
E 6 "

mask %#6(# -

" 6 6 ""
" " " #
$ " # - "
# $ " " "
- #
ChipPreInit() " FALSE

6 # $
$ "6 " TRUE

"
FALSE

$ " % "
(& B

*weight*6*mask*6*offset*

```
Bool xf86SetDefaultVisual(ScrnInfoPtr scrp, int visual);
```

defaultVisual "
% $ " # &(" & &
# ChipPreInit()

; *depth* " *display* "
% ("

" B

visual " A ""
-16 "

" 6 6 ""
" " " #
" -
ChipPreInit()
" FALSE

6 # $
" "6 " TRUE

"
FALSE

```
Bool xf86SetGamma(ScrnInfoPtr scrp, Gamma gamma);
```

gamma " %
$ " # &(" & & # Chip-
PreInit() $ "
##

; *monitor* " % (
"

" B

gamma " A ""
E 0 < 0.0116
"

" 6 6 ""
" " " #
" -
ChipPreInit()
" FALSE

6 # $
"6 " TRUE

"
FALSE

```
void xf86SetDpi(ScrnInfoPtr pScrn, int x, int y);
```

xDpi " *yDpi* "
% " & ""
(& *x* " *y* E -dpi #
" "
$ 6 & & A
% 6 " $
E "
"(" #" (
"

```
void xf86SetBlackWhitePixels(ScrnInfoPtr pScrn);
```

blackPixel " *whitePixel*
" % " $
-flipPixels ## "

```
const char *xf86GetVisualName(int visual);
```

($ # #
""

" " 6 NULL
"

Primary Mode functions

&#" $ $ " # &("(&
" 6 ; # $ (" (&

```
int xf86ValidateModes(ScrnInfoPtr scrp, DisplayModePtr availModes,
                      char **modeNames, ClockRangePtr clockRanges,
                      int *linePitches, int minPitch, int maxPitch,
                      int pitchInc, int minHeight, int maxHeight,
                      int virtualX, int virtualY,
                      unsigned long apertureSize,
                      LookupModeFlags strategy);
```

(& #"
(" ("
" # # &
" " ChipPreInit()

" B

availModes + #" (#

modeNames + #" #
;

clockRanges - $"(&
" ' " $
" # #"
" % ($
clockRanges

linePitches + "(&
" "("
NULL $ "

minPitch ## "(&
" # (" $
linePitches NULL6 " "
$

maxPitch ## "(&
" ; "$ *minPitch*
; "

pitchInc * & E
"(&
" (# ("

minHeight # ## $" E
6 # # "

maxHeight # ## $" E
6 # # "

virtualX E 6
$" $ ("
$ 6 $"
(# ##"
#" "

virtualY E 6
$ ("
$ 6
(# ##"
#" "

apertureSize E 0 (& 1 "
" ## &

strategy & $
#" $
B

LOOKUP_DEFAULT MMM

LOOKUP_BEST_REFRESH #" $
(

LOOKUP_CLOSEST_CLOCK #" $

#
-

LOOKUP_LIST_ORDER (
#"
$ (#
("0 A"1 $ (B

LOOKUP_CLKDIV2 $
"
-

LOOKUP_OPTIONAL_TOLERANCES $
#

E
&
")

LOOKUP_OPTIONAL_TOLERANCES "
& ("$ "
#" &
6 "# 6 #
" 7 E &
")
"(& $ ("
0 " " 1

; $ "
% " B

clock[] + " - 0$
##(1

numClocks /#(" - 0$
##(1

progClock D - ##(

monitor (#

fdFormat # (

videoRam " ## & E 0 (& 1

maxHValue ## E # $"

maxVValue ## # $"

xInc 7 E # # 0"
91

$ % "B

modePool (#" (
$ # ($
"

modes #" &
; "#" 6$ "
" " #"
("
#"

virtualX $"

virtualY

displayWidth

virtualFrom D E $ " # "
#

- *virtu-*
alX " *virtualY* "0 E 1
$ " *maxHeight* #

6 # "6 "
−1

" #" 6 #
##" & & #" " " A
6 " $" "
0 E 1 #" # "
" (& 2
#" " (# &6 $ #
"6 " 0

- #" #6 "
"
"6 # "6 "
−1 $ 6 #(
"#" "0 0 "1

' "#" # " " 6
$ # $ # # #"
# 6 "#" # " & " "
#" 0 " $ # "
16 $ -
#" " #" -6
#" "

& "(& $
"# (# " ""
#&("# #
" $ -$

−16 ChipPreInit()
" FALSE

clockRanges -" - $"(&
" #" " A & " "
*clockRanges*6 > " clockRange
; # " "
(E" /.++$ "

clockRanges $ "B

*minClock*6*max-* $ " #" -
Clock (" $
- #
& % #"
-6 & "$
ClockMulFactor " *Clock-*
DivFactor "
"> &
" " -

clockIndex 0 "& 1 −1 ##
(-

interlaceAllowed TRUE $"

doubleScanAllowed TRUE " (#
$ "

*ClockMulFactor*6 %
ClockDivFactor # " - /+J (
- " 0$
(-1
SynthClock
"
"
- (
"$ 0 - "
<?(" 9 (
! " -1

/ # "
($ # ("
#" - "
- ($
" !
2:1
6 $
! -
$! - (
-
7 6 ClockMulFactor=1 "
ClockDivFactor=2 #
- " -
0 6" #
- " #
" - 1
SynthClock "
(- B 0 mode-
>Clock * ClockMulFactor) /
ClockDivFactor

PrivFlags " "
mode->PrivFlags " $
clockRange " (&
xf86ValidateModes() $
" " $ -
$ "6 - $
" #
& "
" "
& " " "B # &
(6 " &
0 INT1

/ *mode->SynthClock* " $ & "
(& xf86ValidateModes()B $ 4"
-5 6 $ - $ (
" $ ## (
-6 - $ (- " # -
$ ## (B

```
mode->SynthClock = (mode->Clock * ClockMulFactor) / ClockDivFactor
```

```
void xf86PruneDriverModes(ScrnInfoPtr scrp);
```

```
        "   #"     #"   "
  %          (  # -"     "
   # &                          xf86ValidateModes()
        #     #"    "  "6 $
    #        "   "
  (  "  "
```

```
void xf86SetCrtcForModes(ScrnInfoPtr scrp, int adjustFlags);
```

```
              ! H  "       #"
       modes  "   %                    adjustFlags
    #  "  #   $        ! !
      "       "#"   &     "
     INTERLACE_HALVE_V        ! !
  " ( "  " (   #" 6 "      #
    "(&       VScan
```

```
              # &    "
  xf86PruneDriverModes()
```

```
void xf86PrintModes(ScrnInfoPtr scrp);
```

```
                    E     6  "
          (    "           $
       #" (    "           "
  #"A       -6   E    &    6       6
   " $            "6 " (    "  ")
  #     "     "      #"A  "
```

```
              # &    "
  xf86SetCrtcForModes()
```

Secondary Mode functions

```
     " & #"                $      #  &  " (&
    # & #"        6 "$       #  &   ""   & (&
 "     "        ;  #   "  "  "   $ #"   "   6
  #  ( (   #-     #       " & #"
```

```
int xf86GetNearestClock(ScrnInfoPtr scrp, int freq, Bool allowDiv2,
                        int *divider);
```

" -
; & *freq* 0 -7E1 #
#(- E ;
numClocks " *clock* " %
" *allowDiv2* "" # -
(" **divider* "
$ -" "$ " #
- "

& ##(-

```
const char *xf86ModeStatusToString(ModeStatus status);
```

status "
(

```
ModeStatus xf86LookupMode(ScrnInfoPtr scrp, DisplayModePtr modep,
                          ClockRangePtr clockRanges, LookupModeFlags strategy);
```

- #" $ #
" 6 " - #" *modePool* $
#"
" **modep* *clockRanges* " *strategy*
xf86ValidateModes()
(

; *modePool*6 *clock[]*6 *num-*
Clocks " *progClock* " % (
"(("

MODE_OK #" $ "
$ " $& # #" " (
"

```
ModeStatus xf86InitialCheckModeForDriver(ScrnInfoPtr scrp,
                                         DisplayModePtr mode, ClockRangePtr clockRang
                                         LookupModeFlags strategy, int maxPitch,
                                         int virtualX, int virtualY);
```

- " #" #
(" #
&6 *maxHValue* " *maxVValue* "
% " *ValidMode* "
% 6 "
- #" / 6 #" -"
A

#" $ 6
MODE_OK $ "
$ $ A #

```
void xf86DeleteMode(DisplayModePtr *modeList, DisplayModePtr mode);
```

" *mode* # *mod-*
eList &# 6
; "

Functions for handling strings and tokens

(" # - #("$ $
" # $& " # 6 "
- ((" B

```
typedef struct {
    int                 token;
    const char *        name;
} SymTabRec, *SymTabPtr;
```

(" & %&#(- # (
#&(# " - (# "$
$ *token* −1 " NULL *name*

```
const char *xf86TokenToString(SymTabPtr table, int token);
```

table
token # "6 NULL "
0/ '6 " $ "
F - F # "1

```
int xf86StringToToken(SymTabPtr table, const char *string);
```

- *table*
string xf86NameCmp() "
" # # # "6 −1
"

Functions for finding which config file entries to use

(" #
" " "$ & " (" ((" (

Probing discrete clocks on old hardware

xf86GetClocks() #&(" " "
- " "$

```
void xf86GetClocks(ScrnInfoPtr pScrn, int num,
                   Bool (*ClockFunc)(ScrnInfoPtr, int),
                   void (*ProtectRegs)(ScrnInfoPtr, Bool),
                   void (*BlankScreen)(ScrnInfoPtr, Bool),
                   int vertsyncreg, int maskval, int knownclkindex,
                   int knownclkvalue);
```

```
            #     #   #  "
  #     "         -       #(
   "     -  #       (&                 num clock-
  Func                            nA   -     "
            &      "(&   ##
     - $     "    "                 CLK_REG_SAVE
  CLK_REG_RESTORE ProtectRegs         "
  $       ;  "          "$
  $         $  -            BlankScreen
    ( -                 vertsyncreg "  maskval
        "( # -    -
      &             knownclkindex "  knownclkval-
  ue     "  "     - $   -
   - $        $     #    #
   #    ( "    #(     -  ("
       pScrn->numClocks6 "   ("  -
       pScrn->clock[]  &       -
       -7E
```

```
void xf86ShowClocks(ScrnInfoPtr scrp, MessageType from);
```

```
              -            scrp->clock[]  from  "
    $       - $   ("   #
```

Other helper functions

```
Bool xf86IsUnblank(int mode);
```

```
          TRUE $          # "     "(&
  mode ;        (  ( -"6  "                  FALSE
   $           # "     ;  ( -
      SCREEN_SAVER_ON  "   SCREEN_SAVER_CYCLE6  "
           # "     ;    ( -
  SCREEN_SAVER_OFF  "  SCREEN_SAVER_FORCER
  # &          #                SaveScreen()
               # "
```

The vgahw module

```
$ # "     "            6       "    ##
" ",*     6  "    "   ,*    #
```

Data Structures

(" "(& $ #" " 7
D & " " vgaHW.h.

General vgahw Functions

```
Bool vgaHWGetHWRec(ScrnInfoPtr pScrn);
```

7D 6 "
- % A *privates* + -
-" *privates*6 6 "
& " ("
" # &(" # " A Chip-
PreInit() 7D E "6
" $ " & "B

ModeReg.DAC[] " $ "
#

ModeReg.Attribute[0x11] " $ "
"

ShowOverscan " "
F% $ F

paletteEnabled " +%'

cmapSaved " +%'

pScrn " %

"" (6 vgaHWSetStdFuncs() "
" $
" ",*

"6 7D
((" # ScrnInfoPtr $
VGAHWPTR(pScrn) #

```
void vgaHWFreeHWRec(ScrnInfoPtr pScrn);
```

7D "(
" # " A ChipFreeScreen()

```
Bool vgaHWSetRegCounts(ScrnInfoPtr pScrn, int numCRTC,
                       int numSequencer, int numGraphics, int numAttribut
```

$ #(! !6 % ; 6
* " (("

#- ("" ("
" "$ vgaHWSave() " vgaHWRestore()
"(" 7D
("$ vgaHWGetHWRec() "
" " vgaHW.h $B

```
#define VGA_NUM_CRTC 25
#define VGA_NUM_SEQ   5
#define VGA_NUM_GFX   9
#define VGA_NUM_ATTR 21
```

```
Bool vgaHWCopyReg(vgaRegPtr dst, vgaRegPtr src);
```

,* "
src *dst* / A (#
&" $ memcpy() 0 # 1
TRUE (#
CRTC " " " *dst*

```
void vgaHWSetStdFuncs(vgaHWPtr hwp);
```

" *hwp* $ " ",*
"(& vgaHWGetHWRec()6
& " &
" ("($ "
$" #) 0 #
!) 16 ("
"$" *hwp->PIOOffset* "
$ offset6 " $& $
" ",*) """
"
E vgaHWGetHWRec() 0/ B
& 8 9:? @ " 1

```
void vgaHWSetMmioFuncs(vgaHWPtr hwp, CARD8 *base, int offset);
```

"
" $ $
hwp->MMIOBase "$ *base*6$ # (
""
" *hwp->MMIOOffset* "$ *off-*
*set*6$ # (" $& $
" ",*) """
#
& ($,*)
#" (" #
6 " $ "

" $
" ("($

```
Bool vgaHWMapMem(ScrnInfoPtr pScrn);
```

,* ## & $ "$
; 7D (" "
; " *MapPhys* *MapSize* 0
& " E ,* ## & $ "$1
" 7D # ("
($ 6
" 0xA0000 *MapPhys*
" (64 * 1024) *MapSize* # (
"(# ,*
" " A &6 vgaH-
WSave() " vgaHWRestore() #&
& " ,* ## & 0 $
& $ vgaHWUnmapMem() # ,*
&(1

```
void vgaHWUnmapMem(ScrnInfoPtr pScrn);
```

,* ## &$ "$
&(" ## & (# "
Base " 7D " NULL "
& # "

```
void vgaHWGetIOBase(vgaHWPtr hwp);
```

IOBase " 7
D # ("(&
" "$

VGAHW_GET_IOBASE() (
vgaHW.h)(6 " #&("
$ $#" ""0 # 6
ChipProbe() 1

```
void vgaHWUnlock(vgaHWPtr hwp);
```

- ,* CRTC[0-7] 6
"# ("(# $

```
void vgaHWLock(vgaHWPtr hwp);
```

- ,* CRTC[0-7]

```
void vgaHWEnable(vgaHWPtr hwp);
```

(,* (& # 0/ 6
8 9: ? @ " 1

```
void vgaHWDisable(vgaHWPtr hwp);
```

" (,* (& # 0/ 6
8 9: ? @ " 1

```
void vgaHWSave(ScrnInfoPtr pScrn, vgaRegPtr save, int flags);
```

,* $
" (& *save flags*
$ " B

VGA_SR_MODE #" "

VGA_SR_FONTS #") " "

VGA_SR_CMAP # 0+. 1 "

VGA_SR_ALL ("

7D " *IOBase* " # ("
(" VGA_SR_FONTS
*flags*6 ,* ## & $ "$ # (# "
A vgaHWMapMem() $ (" #
6 " vgaHWUnmapMem() $ (" #
$ " vgaHWSave() ($
" vgaHWSaveColormap()6vgaHWSaveMode()6
vgaHWSaveFonts() & "
" "" $ & $
$ #" A (

```
void vgaHWSaveMode(ScrnInfoPtr pScrn, vgaRegPtr save);
```

,* #" &
" " (& *save*
" B

! !K@ @ 9L
(K@ @ ?L
* K@ 9L
%; K@ ?L

#(& "#&(#"
"(& vgaHWSetRegCounts()

void vgaHWSaveFonts(ScrnInfoPtr pScrn, vgaRegPtr save);

#" " "
" " ## & "$
#"6 " ,* ## &$ "$#
(# "$ vgaHWMapMem() (

#6 #)
#&($
#A ,!" "& -

void vgaHWSaveColormap(ScrnInfoPtr pScrn, vgaRegPtr save);

,* # 0+. 1 3
6 # & #
"($ A "(6 "
" "

void vgaHWRestore(ScrnInfoPtr pScrn, vgaRegPtr restore, int flags);

,*
##" " "
(& *restore flags* # " (" (
vgaHWSave()

7D " *IOBase* "# ("
(" VGA_SR_FONTS
*flags*6 ,* ## &$ "$# (# "
A vgaHWMapMem() $ (" #
6 " vgaHWUnmapMem() $ (" #
$ " vgaHWRestore()
($ " vgaHWRestoreFonts()6 vgaHWRe-
storeMode()6 vgaHWRestoreColormap() &
" "" ""
$ & $ # $
#" A (

void vgaHWRestoreMode(ScrnInfoPtr pScrn, vgaRegPtr restore);

,* #" &
" # " "
(& *restore* " B

! !K@ @ 9L
(K@ @ ?L
* K@ 9L
%; K@ ?L

#(& "#&(#"
"(& vgaHWSetRegCounts()

```
void vgaHWRestoreFonts(ScrnInfoPtr pScrn, vgaRegPtr restore);
```

#" " "
" ## & ,* ## & $ "$ #
(# "$ vgaHWMapMem() (

#6 #)
#& ($
#A ,!" "& -

```
void vgaHWRestoreColormap(ScrnInfoPtr pScrn, vgaRegPtr restore);
```

,* # 0+. 1

```
void vgaHWInit(ScrnInfoPtr pScrn, DisplayModePtr mode);
```

7D A *ModeReg* "
$ ##
" #" ; % A *depth*
" "6 $ " # $
##"

```
void vgaHWSeqReset(vgaHWPtr hwp, Bool start);
```

,* ; TRUE6
" FALSE6 ""

```
void vgaHWProtect(ScrnInfoPtr pScrn, Bool on);
```

,* "## & #
" " & & "$
TRUE(## 6 "$ FALSE
##

```
Bool vgaHWSaveScreen(ScreenPtr pScreen, int mode);
```

(- " (-
(-" $ *mode* SCREEN_SAVER_ON
SCREEN_SAVER_CYCLE6 " (-" $ *mode*
SCREEN_SAVER_OFF SCREEN_SAVER_FORCER

```
void vgaHWBlankScreen(ScrnInfoPtr pScrn, Bool on);
```

(- " (-
(-"$ *on* FALSE6 " (-"$ *on*
TRUE "" $
% A (" " # %
0$ (-6 # 1

VGA Colormap Functions

$#" " " # 0 ! # 7 "
"$ $ B

```
Bool vgaHWHandleColormaps(ScreenPtr pScreen);
```

VGA Register Access Functions

$#" (" ",* (&
" 7D D 7D "
"$ " ",*)
"" 6 $#" " (
6 " 7D (" (&
vgaHWSetMmioFuncs() " (" (%# ") # #&
; " ,* "
("

```
void writeCrtc(vgaHWPtr hwp, CARD8 index, CARD8 value);
```

D *value* ! ! *index*

```
CARD8 readCrtc(vgaHWPtr hwp, CARD8 index);
```

" #! ! *index*

```
void writeGr(vgaHWPtr hwp, CARD8 index, CARD8 value);
```

D *value* * ! *index*

```
CARD8 readGR(vgaHWPtr hwp, CARD8 index);
```

" #* !
index

```
void writeSeq(vgaHWPtr hwp, CARD8 index, CARD8, value);
```

D *value* %; *index*

```
CARD8 readSeq(vgaHWPtr hwp, CARD8 index);
```

" #%; *index*

```
void writeAttr(vgaHWPtr hwp, CARD8 index, CARD8, value);
```

D *value* (! *index*
D $ " "
(G0 0x201 " *hwp->palet-*
teEnabled "
"("$ *hwp->paletteEnabled*
TRUE " $ FALSE

```
CARD8 readAttr(vgaHWPtr hwp, CARD8 index);
```

" # (!
index D $ "
" (G0 0x201 "
hwp->paletteEnabled "
"("$ *hwp->palet-*
teEnabled TRUE " $ FALSE

```
void writeMiscOut(vgaHWPtr hwp, CARD8 value);
```

D 4*value*5

```
CARD8 readMiscOut(vgwHWPtr hwp);
```

" #

```
void enablePalette(vgaHWPtr hwp);
```

! "" ((
! " " hwp->paletteEn-
abled TRUE

```
void disablePalette(vgaHWPtr hwp);
```

% "" ((!
" " hwp->paletteEnabled
FALSE

```
void writeDacMask(vgaHWPtr hwp, CARD8 value);
```

D *value* ! -

```
CARD8 readDacMask(vgaHWptr hwp);
```

" # ! -

```
void writeDacReadAddress(vgaHWPtr hwp, CARD8 value);
```

D *value* ! " ""

```
void writeDacWriteAddress(vgaHWPtr hwp, CARD8 value);
```

D *value* !D ""

```
void writeDacData(vgaHWPtr hwp, CARD8 value);
```

D *value* !

```
CARD8 readDacData(vgaHWptr hwp);
```

" # !

```
CARD8 readEnable(vgaHWptr hwp);
```

" # ,* ' (
0/ B 8 9: ? @ "
1

```
void writeEnable(vgaHWPtr hwp, CARD8 value);
```

D *value* ,* ' (0/ B
8 9: ? @ " 1

Some notes about writing a driver

/ 'B # "

$ $ ("" ! "
" $ # " #(6 " $,* "
" 0 $ " A
(" 1

("
" 4 $ " 5 " # # # #
(" 6 " / # &
" $ ("

' " # ; " #6 " "
& (&#(" # $ "
# " # 4UUU5 " 6 $ 4UUU5
4EEE5 & (&#(

Include files

" # & " $ " B

```
"xf86.h"
"xf86_OSproc.h"
"xf86_ansic.h"
"xf86Resources.h"
```

D () (0 " " 1 " $ "(""B

```
"compiler.h"
```

/ B " 6 # ("" "xf86_ansic.h"

" ! ")" " " B

```
"xf86PciInfo.h"
```

" ! " B

```
"xf86Pci.h"
```

(- $ "B

```
"mibank.h"
```

%D " B

"mipointer.h"

" # # (- " B

"mibstore.h"

" # # " " B

"micmap.h"

" $ # " 6 " B

"vgaHW.h"

,* 7 # # "B

"xf1bpp.h"

,* '*! : "B

"xf4bpp.h"

("B

```
#define PSZ 8
#include "cfb.h"
#undef PSZ
```

(:6 <? =< $ (" # B

```
"cfb16.h"
"cfb24.h"
"cfb32.h"
```

" A $ " B

"zzz.h"

/ " $ B

```
  "xf86Priv.h"
  "xf86Privstr.h"
  "xf86_libc.h"
  "xf86_OSlib.h"
  "Xos.h"
 & %  "
```

Data structures and initialisation

C $ # "(" "B

```
#define VERSION <version-as-an-int>
#define ZZZ_NAME "ZZZ"          /* the name used to prefix messages */
#define ZZZ_DRIVER_NAME "zzz"  /* the driver name as used in config file */
#define ZZZ_MAJOR_VERSION <int>
#define ZZZ_MINOR_VERSION <int>
#define ZZZ_PATCHLEVEL    <int>
```

```
/ 'B   ZZZ_DRIVER_NAME  " #   #   "  #" $
   -  F (F  6  FN" F      #
```

C # (" "6 $ " ; "
(# # (
UUUN ,' N/ ' 0 - 1

```
DriverRec ZZZ = {
    VERSION,
    ZZZ_DRIVER_NAME,
    ZZZIdentify,
    ZZZProbe,
    ZZZAvailableOptions,
    NULL,
    0
};
```

C " " # B

```
static SymTabRec ZZZChipsets[] = {
    { PCI_CHIP_ZZZ1234, "zzz1234a" },
    { PCI_CHIP_ZZZ5678, "zzz5678a" },
    { -1,               NULL }
};
```

```
  -  " # & (  &  "  # &  ; &
" &  "  "  & ! "
 ! "  # (  6 (  A # "  &  "
   ( ! "  "  0 - % 16 #  " (( &
```

" D " - ## "" #
(" " # &

C " xf86MatchPciInstances() 0 ## "" "
! "1 # ! " "
(" "B

```
static PciChipsets ZZZPciChipsets[] = {
    { PCI_CHIP_ZZZ1234, PCI_CHIP_ZZZ1234, RES_SHARED_VGA },
    { PCI_CHIP_ZZZ5678, PCI_CHIP_ZZZ5678, RES_SHARED_VGA },
    { -1,               -1,               RES_UNDEFINED }
}
```

C 89: " , " ; "
"& # & "" B

```
static XF86ModuleVersionInfo zzzVersRec =
{
    "zzz",
    MODULEVENDORSTRING,
    MODINFOSTRING1,
    MODINFOSTRING2,
    XF86_VERSION_CURRENT,
    ZZZ_MAJOR_VERSION, ZZZ_MINOR_VERSION, ZZZ_PATCHLEVEL,
    ABI_CLASS_VIDEODRV,
    ABI_VIDEODRV_VERSION,
    MOD_CLASS_VIDEODRV,
    {0,0,0,0}
};
```

C " " " A " # ("
" (($ "(" " "zzz.h" 6 #
-B

```
typedef struct {
    type1  field1;
    type2  field2;
    int    fooHack;
    Bool   pciRetry;
    Bool   noAccel;
    Bool   hwCursor;
    CloseScreenProcPtr CloseScreen;
    OptionInfoPtr Options;
    ...
} ZZZRec, *ZZZPtr;
```

C " &(
$ " 4 " "5 "("$
($

```
typedef enum {
    OPTION_FOO_HACK,
    OPTION_PCI_RETRY,
    OPTION_HW_CURSOR,
    OPTION_NOACCEL
} ZZZOpts;

static const OptionInfoRec ZZZOptions[] = {
  { OPTION_FOO_HACK,  "FooHack",   OPTV_INTEGER, {0}, FALSE },
  { OPTION_PCI_RETRY, "PciRetry",  OPTV_BOOLEAN, {0}, FALSE },
  { OPTION_HW_CURSOR, "HWcursor",  OPTV_BOOLEAN, {0}, FALSE },
  { OPTION_NOACCEL,   "NoAccel",   OPTV_BOOLEAN, {0}, FALSE },
  { -1,               NULL,        OPTV_NONE,    {0}, FALSE }
};
```

Functions

SetupProc

```
  "& #  &  " " # "  6              ModuleData   (    ;  "      "
(    #    "      " "  F "   F                          Setup()
 ;  "6 $             xf86AddDriver()  ""  "      #     "


static MODULESETUPPROTO(zzzSetup);

XF86ModuleData zzzModuleData = { &zzzVersRec, zzzSetup, NULL };

static pointer
zzzSetup(pointer module, pointer opts, int *errmaj, int *errmin)
{
    static Bool setupDone = FALSE;

    /* This module should be loaded only once, but check to be sure. */

    if (!setupDone) {
        /*
         * Modules that this driver always requires may be loaded
         * here  by calling LoadSubModule().
         */

        setupDone = TRUE;
        xf86AddDriver(&MGA, module, 0);

        /*
         * The return value must be non-NULL on success even though
         * there is no TearDownProc.
         */
        return (pointer)1;
    } else {
        if (errmaj) *errmaj = LDR_ONCEONLY;
```

```
        return NULL;
    }
}
```

GetRec, FreeRec

& ; " " A "
" - % A *driverPrivate* " % A *driver-Private* " NULL6 & - "&
(" 6 " 3& xnfcalloc() "
E "6 "

/ 'B D # " " $ " "
E E &
(

```
static Bool
ZZZGetRec(ScrnInfoPtr pScrn)
{
    if (pScrn->driverPrivate != NULL)
        return TRUE;
    pScrn->driverPrivate = xnfcalloc(sizeof(ZZZRec), 1);
    /* Initialise as required */
    ...
    return TRUE;
}
```

"zzz.h" $ UUU $ *pScrn*B

```
#define ZZZPTR(p) ((ZZZPtr)((p)->driverPrivate))
```

(6 NULL ("B

```
static void
ZZZFreeRec(ScrnInfoPtr pScrn)
{
    if (pScrn->driverPrivate == NULL)
        return;
    xfree(pScrn->driverPrivate);
    pScrn->driverPrivate = NULL;
}
```

Identify

Identify() ((6 " & &
" & # 6$ # "
" &B

```
static void
ZZZIdentify(int flags)
{
    xf86PrintChipsets(ZZZ_NAME, "driver for ZZZ Tech chipsets",
                      ZZZChipsets);
}
```

Probe

```
          Probe()                    "           "
$      "      6 "          "&   #"(&     "
 6   #    6 "      %          "(  # #   (6 "
   " "     #               "$     " 3
 "      "6" A    #     $  (   "   A "  &
      ;  "%              A       &  $"


       # "  &

/ 'B        xf86DrvMsg()        (   "  #    (


static Bool
ZZZProbe(DriverPtr drv, int flags)
{
    Bool foundScreen = FALSE;
    int numDevSections, numUsed;
    GDevPtr *devSections;
    int *usedChips;
    int i;

    /*
     * Find the config file Device sections that match this
     * driver, and return if there are none.
     */
    if ((numDevSections = xf86MatchDevice(ZZZ_DRIVER_NAME,
                                          &devSections)) <= 0) {
        return FALSE;
    }

    /*
     * Since this is a PCI card, "probing" just amounts to checking
     * the PCI data that the server has already collected.  If there
     * is none, return.
     *
     * Although the config file is allowed to override things, it
     * is reasonable to not allow it to override the detection
     * of no PCI video cards.
     *
     * The provided xf86MatchPciInstances() helper takes care of
     * the details.
     */
```

```
    /* test if PCI bus present */
    if (xf86GetPciVideoInfo()) {

        numUsed = xf86MatchPciInstances(ZZZ_NAME, PCI_VENDOR_ZZZ,
                            ZZZChipsets, ZZZPciChipsets, devSections,
                            numDevSections, drv, &usedChips);

        for (i = 0; i < numUsed; i++) {
            ScrnInfoPtr pScrn = NULL;
            if ((pScrn = xf86ConfigPciEntity(pScrn, flags, usedChips[i],
                                             ZZZPciChipsets, NULL, NULL,
                                             NULL, NULL, NULL))) {
               /* Allocate a ScrnInfoRec */
               pScrn->driverVersion = VERSION;
               pScrn->driverName    = ZZZ_DRIVER_NAME;
               pScrn->name          = ZZZ_NAME;
               pScrn->Probe         = ZZZProbe;
               pScrn->PreInit       = ZZZPreInit;
               pScrn->ScreenInit    = ZZZScreenInit;
               pScrn->SwitchMode    = ZZZSwitchMode;
               pScrn->AdjustFrame   = ZZZAdjustFrame;
               pScrn->EnterVT       = ZZZEnterVT;
               pScrn->LeaveVT       = ZZZLeaveVT;
               pScrn->FreeScreen    = ZZZFreeScreen;
               pScrn->ValidMode     = ZZZValidMode;
               foundScreen = TRUE;
               /* add screen to entity */
            }
        }
        xfree(usedChips);
    }

#ifdef HAS_ISA_DEVS
    /*
     * If the driver supports ISA hardware, the following block
     * can be included too.
     */
    numUsed = xf86MatchIsaInstances(ZZZ_NAME, ZZZChipsets,
                            ZZZIsaChipsets, drv, ZZZFindIsaDevice,
                            devSections, numDevSections, &usedChips);
    for (i = 0; i < numUsed; i++) {
        ScrnInfoPtr pScrn = NULL;
 if ((pScrn = xf86ConfigIsaEntity(pScrn, flags, usedChips[i],
      ZZZIsaChipsets, NULL, NULL, NULL,
      NULL, NULL))) {
            pScrn->driverVersion = VERSION;
            pScrn->driverName    = ZZZ_DRIVER_NAME;
            pScrn->name          = ZZZ_NAME;
            pScrn->Probe         = ZZZProbe;
            pScrn->PreInit       = ZZZPreInit;
            pScrn->ScreenInit    = ZZZScreenInit;
            pScrn->SwitchMode    = ZZZSwitchMode;
            pScrn->AdjustFrame   = ZZZAdjustFrame;
            pScrn->EnterVT       = ZZZEnterVT;
```

```
            pScrn->LeaveVT       = ZZZLeaveVT;
            pScrn->FreeScreen    = ZZZFreeScreen;
            pScrn->ValidMode     = ZZZValidMode;
            foundScreen = TRUE;
        }
    }
    xfree(usedChips);
#endif /* HAS_ISA_DEVS */

    xfree(devSections);
    return foundScreen;
```

AvailableOptions

```
          AvailableOptions()
( "       ( -           6              (
(   "        $          (     #
```

PreInit

```
          PreInit()                    "       #
 ; "  "  #              ( 6  "
  %         (        (
    #     "(   "           $ &   (

       # "  &

/ '%B

      PreInit()       &   "   "          8    0
                1

<       "   #  (     &                 8
    #     "      -     %     A                        privates  "   "(
      "   6( "      -     %    A                        devPrivates  "   "
    (     "                 driverPrivate  "   "   (     "

=        %        (      "(              "6
  %        (      "    #         ;        (
    "

? , &        %        (       " $             "
    #          "   "


static Bool
ZZZPreInit(ScrnInfoPtr pScrn, int flags)
{
    /* Fill in the monitor field */
    pScrn->monitor = pScrn->confScreen->monitor;

    /*
     * If using the vgahw module, it will typically be loaded
     * here by calling xf86LoadSubModule(pScrn, "vgahw");
```

```
 */

/*
 * Set the depth/bpp.  Use the globally preferred depth/bpp.  If the
 * driver has special default depth/bpp requirements, the defaults should
 * be specified here explicitly.
 * We support both 24bpp and 32bpp framebuffer layouts.
 * This sets pScrn->display also.
 */
if (!xf86SetDepthBpp(pScrn, 0, 0, 0,
                     Support24bppFb | Support32bppFb)) {
    return FALSE;
} else {
    if (depth/bpp isn't one we support) {
        print error message;
        return FALSE;
    }
}
/* Print out the depth/bpp that was set */
xf86PrintDepthBpp(pScrn);

/* Set bits per RGB for 8bpp */
if (pScrn->depth <= 8) {
    /* Take into account a dac_6_bit option here */
    pScrn->rgbBits = 6 or 8;
}

/*
 * xf86SetWeight() and xf86SetDefaultVisual() must be called
 * after pScrn->display is initialised.
 */

/* Set weight/mask/offset for depth > 8 */
if (pScrn->depth > 8) {
    if (!xf86SetWeight(pScrn, defaultWeight, defaultMask)) {
        return FALSE;
    } else {
        if (weight isn't one we support) {
            print error message;
            return FALSE;
        }
    }
}

/* Set the default visual. */
if (!xf86SetDefaultVisual(pScrn, -1)) {
    return FALSE;
} else {
    if (visual isn't one we support) {
        print error message;
        return FALSE;
    }
}
```

```
/* If the driver supports gamma correction, set the gamma. */
if (!xf86SetGamma(pScrn, default_gamma)) {
    return FALSE;
}

/* This driver uses a programmable clock */
pScrn->progClock = TRUE;

/* Allocate the ZZZRec driverPrivate */
if (!ZZZGetRec(pScrn)) {
    return FALSE;
}

pZzz = ZZZPTR(pScrn);

/* Collect all of the option flags (fill in pScrn->options) */
xf86CollectOptions(pScrn, NULL);

/*
 * Process the options based on the information in ZZZOptions.
 * The results are written to pZzz->Options.  If all of the options
 * processing is done within this function a local variable "options"
 * can be used instead of pZzz->Options.
 */
if (!(pZzz->Options = xalloc(sizeof(ZZZOptions))))
    return FALSE;
(void)memcpy(pZzz->Options, ZZZOptions, sizeof(ZZZOptions));
xf86ProcessOptions(pScrn->scrnIndex, pScrn->options, pZzz->Options);

/*
 * Set various fields of ScrnInfoRec and/or ZZZRec based on
 * the options found.
 */
from = X_DEFAULT;
pZzz->hwCursor = FALSE;
if (xf86IsOptionSet(pZzz->Options, OPTION_HW_CURSOR)) {
    from = X_CONFIG;
    pZzz->hwCursor = TRUE;
}
xf86DrvMsg(pScrn->scrnIndex, from, "Using %s cursor\n",
           pZzz->hwCursor ? "HW" : "SW");
if (xf86IsOptionSet(pZzz->Options, OPTION_NOACCEL)) {
    pZzz->noAccel = TRUE;
    xf86DrvMsg(pScrn->scrnIndex, X_CONFIG,
               "Acceleration disabled\n");
} else {
    pZzz->noAccel = FALSE;
}
if (xf86IsOptionSet(pZzz->Options, OPTION_PCI_RETRY)) {
    pZzz->UsePCIRetry = TRUE;
    xf86DrvMsg(pScrn->scrnIndex, X_CONFIG, "PCI retry enabled\n");
}
pZzz->fooHack = 0;
if (xf86GetOptValInteger(pZzz->Options, OPTION_FOO_HACK,
```

```
                        &pZzz->fooHack)) {
    xf86DrvMsg(pScrn->scrnIndex, X_CONFIG, "Foo Hack set to %d\n",
               pZzz->fooHack);
}

/*
 * Find the PCI slot(s) that this screen claimed in the probe.
 * In this case, exactly one is expected, so complain otherwise.
 * Note in this case we're not interested in the card types so
 * that parameter is set to NULL.
 */
if ((i = xf86GetPciInfoForScreen(pScrn->scrnIndex, &pciList, NULL))
    != 1) {
    print error message;
    ZZZFreeRec(pScrn);
    if (i > 0)
        xfree(pciList);
    return FALSE;
}
/* Note that pciList should be freed below when no longer needed */

/*
 * Determine the chipset, allowing config file chipset and
 * chipid values to override the probed information.  The config
 * chipset value has precedence over its chipid value if both
 * are present.
 *
 * It isn't necessary to fill in pScrn->chipset if the driver
 * keeps track of the chipset in its ZZZRec.
 */

...

/*
 * Determine video memory, fb base address, I/O addresses, etc,
 * allowing the config file to override probed values.
 *
 * Set the appropriate pScrn fields (videoRam is probably the
 * most important one that other code might require), and
 * print out the settings.
 */

...

/* Initialise a clockRanges list. */

...

/* Set any other chipset specific things in the ZZZRec */

...

/* Select valid modes from those available */
```

```
i = xf86ValidateModes(pScrn, pScrn->monitor->Modes,
                      pScrn->display->modes, clockRanges,
                      NULL, minPitch, maxPitch, rounding,
                      minHeight, maxHeight,
                      pScrn->display->virtualX,
                      pScrn->display->virtualY,
                      pScrn->videoRam * 1024,
                      LOOKUP_BEST_REFRESH);
if (i == -1) {
    ZZZFreeRec(pScrn);
    return FALSE;
}

/* Prune the modes marked as invalid */

xf86PruneDriverModes(pScrn);

/* If no valid modes, return */

if (i == 0 || pScrn->modes == NULL) {
    print error message;
    ZZZFreeRec(pScrn);
    return FALSE;
}

/*
 * Initialise the CRTC fields for the modes.  This driver expects
 * vertical values to be halved for interlaced modes.
 */
xf86SetCrtcForModes(pScrn, INTERLACE_HALVE_V);

/* Set the current mode to the first in the list. */
pScrn->currentMode = pScrn->modes;

/* Print the list of modes being used. */
xf86PrintModes(pScrn);

/* Set the DPI */
xf86SetDpi(pScrn, 0, 0);

/* Load bpp-specific modules */
switch (pScrn->bitsPerPixel) {
case 1:
    mod = "xf1bpp";
    break;
case 4:
    mod = "xf4bpp";
    break;
case 8:
    mod = "cfb";
    break;
case 16:
    mod = "cfb16";
    break;
```

```
    case 24:
        mod = "cfb24";
        break;
    case 32:
        mod = "cfb32";
        break;
    }
    if (mod && !xf86LoadSubModule(pScrn, mod))
        ZZZFreeRec(pScrn);
        return FALSE;

    /* Load XAA if needed */
    if (!pZzz->noAccel || pZzz->hwCursor)
        if (!xf86LoadSubModule(pScrn, "xaa")) {
            ZZZFreeRec(pScrn);
            return FALSE;
        }

    /* Done */
    return TRUE;
}
```

MapMem, UnmapMem

```
        #  " #    " ## & " &   ## &
    ; "            # "  &6 (
```

```
static Bool
ZZZMapMem(ScrnInfoPtr pScrn)
{
    /* Call xf86MapPciMem() to map each PCI memory area */
    ...
    return TRUE or FALSE;
}

static Bool
ZZZUnmapMem(ScrnInfoPtr pScrn)
{
    /* Call xf86UnMapVidMem() to unmap each memory area */
    ...
    return TRUE or FALSE;
}
```

Save, Restore

```
          "          "
  # "  &6 (
```

```
static void
```

```
ZZZSave(ScrnInfoPtr pScrn)
{
    /*
     * Save state into per-screen data structures.
     * If using the vgahw module, vgaHWSave will typically be
     * called here.
     */
    ...
}

static void
ZZZRestore(ScrnInfoPtr pScrn)
{
    /*
     * Restore state from per-screen data structures.
     * If using the vgahw module, vgaHWRestore will typically be
     * called here.
     */
    ...
}
```

ModeInit

$ " #" A # " &6 (

```
static Bool
ZZZModeInit(ScrnInfoPtr pScrn, DisplayModePtr mode)
{
    /*
     * Program a video mode.  If using the vgahw module,
     * vgaHWInit and vgaRestore will typically be called here.
     * Once up to the point where there can't be a failure
     * set pScrn->vtSema to TRUE.
     */
    ...
}
```

ScreenInit

ScreenInit() "
6 " " # % ($ &
" " " #(&
6 " " #"

" &

/ 'B % 0 *pScreen*1 " " 6 (" ScrnInfoRecs & -" # 6 "
6 (" #

```
static Bool
ZZZScreenInit(int scrnIndex, ScreenPtr pScreen, int argc, char **argv)
{
    /* Get the ScrnInfoRec */
    pScrn = xf86Screens[pScreen->myNum];

    /*
     * If using the vgahw module, its data structures and related
     * things are typically initialised/mapped here.
     */

    /* Save the current video state */
    ZZZSave(pScrn);

    /* Initialise the first mode */
    ZZZModeInit(pScrn, pScrn->currentMode);

    /* Set the viewport if supported */

    ZZZAdjustFrame(scrnIndex, pScrn->frameX0, pScrn->frameY0, 0);

    /*
     * Setup the screen's visuals, and initialise the framebuffer
     * code.
     */

    /* Reset the visual list */
    miClearVisualTypes();

    /*
     * Setup the visuals supported.  This driver only supports
     * TrueColor for bpp > 8, so the default set of visuals isn't
     * acceptable.  To deal with this, call miSetVisualTypes with
     * the appropriate visual mask.
     */

    if (pScrn->bitsPerPixel > 8) {
        if (!miSetVisualTypes(pScrn->depth, TrueColorMask,
                              pScrn->rgbBits, pScrn->defaultVisual))
            return FALSE;
    } else {
        if (!miSetVisualTypes(pScrn->depth,
                              miGetDefaultVisualMask(pScrn->depth),
                              pScrn->rgbBits, pScrn->defaultVisual))
            return FALSE;
    }

    /*
     * Initialise the framebuffer.
     */

    switch (pScrn->bitsPerPixel) {
    case 1:
        ret = xf1bppScreenInit(pScreen, FbBase,
```

```
                            pScrn->virtualX, pScrn->virtualY,
                            pScrn->xDpi, pScrn->yDpi,
                            pScrn->displayWidth);
    break;
case 4:
    ret = xf4bppScreenInit(pScreen, FbBase,
                            pScrn->virtualX, pScrn->virtualY,
                            pScrn->xDpi, pScrn->yDpi,
                            pScrn->displayWidth);
    break;
case 8:
    ret = cfbScreenInit(pScreen, FbBase,
                        pScrn->virtualX, pScrn->virtualY,
                        pScrn->xDpi, pScrn->yDpi,
                        pScrn->displayWidth);
    break;
case 16:
    ret = cfb16ScreenInit(pScreen, FbBase,
                          pScrn->virtualX, pScrn->virtualY,
                          pScrn->xDpi, pScrn->yDpi,
                          pScrn->displayWidth);
    break;
case 24:
    ret = cfb24ScreenInit(pScreen, FbBase,
                          pScrn->virtualX, pScrn->virtualY,
                          pScrn->xDpi, pScrn->yDpi,
                          pScrn->displayWidth);
    break;
case 32:
    ret = cfb32ScreenInit(pScreen, FbBase,
                          pScrn->virtualX, pScrn->virtualY,
                          pScrn->xDpi, pScrn->yDpi,
                          pScrn->displayWidth);
    break;
default:
    print a message about an internal error;
    ret = FALSE;
    break;
}

if (!ret)
    return FALSE;

/* Override the default mask/offset settings */
if (pScrn->bitsPerPixel > 8) {
    for (i = 0, visual = pScreen->visuals;
         i < pScreen->numVisuals; i++, visual++) {
        if ((visual->class | DynamicClass) == DirectColor) {
            visual->offsetRed = pScrn->offset.red;
            visual->offsetGreen = pScrn->offset.green;
            visual->offsetBlue = pScrn->offset.blue;
            visual->redMask = pScrn->mask.red;
            visual->greenMask = pScrn->mask.green;
            visual->blueMask = pScrn->mask.blue;
```

```
        }
    }
}

/*
 * If banking is needed, initialise an miBankInfoRec (defined in
 * "mibank.h"), and call miInitializeBanking().
 */
if (!miInitializeBanking(pScreen, pScrn->virtualX, pScrn->virtualY,
                                  pScrn->displayWidth, pBankInfo))
    return FALSE;

/*
 * If backing store is to be supported (as is usually the case),
 * initialise it.
 */
miInitializeBackingStore(pScreen);

/*
 * Set initial black & white colourmap indices.
 */
xf86SetBlackWhitePixels(pScreen);

/*
 * Install colourmap functions.  If using the vgahw module,
 * vgaHandleColormaps would usually be called here.
 */

...

/*
 * Initialise cursor functions.  This example is for the mi
 * software cursor.
 */
miDCInitialize(pScreen, xf86GetPointerScreenFuncs());

/* Initialise the default colourmap */
switch (pScrn->depth) {
case 1:
    if (!xf1bppCreateDefColormap(pScreen))
        return FALSE;
    break;
case 4:
    if (!xf4bppCreateDefColormap(pScreen))
        return FALSE;
    break;
default:
    if (!cfbCreateDefColormap(pScreen))
        return FALSE;
    break;
}

/*
 * Wrap the CloseScreen vector and set SaveScreen.
```

```
     */
    ZZZPTR(pScrn)->CloseScreen = pScreen->CloseScreen;
    pScreen->CloseScreen = ZZZCloseScreen;
    pScreen->SaveScreen = ZZZSaveScreen;

    /* Report any unused options (only for the first generation) */
    if (serverGeneration == 1) {
        xf86ShowUnusedOptions(pScrn->scrnIndex, pScrn->options);
    }

    /* Done */
    return TRUE;
}
```

SwitchMode

SwitchMode() #" $ "(& "

```
static Bool
ZZZSwitchMode(int scrnIndex, DisplayModePtr mode, int flags)
{
    return ZZZModeInit(xf86Screens[scrnIndex], mode);
}
```

AdjustFrame

AdjustFrame() "

```
static void
ZZZAdjustFrame(int scrnIndex, int x, int y, int flags)
{
    /* Adjust the viewport */
}
```

EnterVT, LeaveVT

EnterVT() " LeaveVT()

#" &

```
static Bool
ZZZEnterVT(int scrnIndex, int flags)
{
    ScrnInfoPtr pScrn = xf86Screens[scrnIndex];
    return ZZZModeInit(pScrn, pScrn->currentMode);
}

static void
```

```
ZZZLeaveVT(int scrnIndex, int flags)
{
    ScrnInfoPtr pScrn = xf86Screens[scrnIndex];
    ZZZRestore(pScrn);
}
```

CloseScreen

CloseScreen() B

" & / $ & $ " *pScreen->CloseScreen*6 " (&

```
static Bool
ZZZCloseScreen(int scrnIndex, ScreenPtr pScreen)
{
    ScrnInfoPtr pScrn = xf86Screens[scrnIndex];
    if (pScrn->vtSema) {
        ZZZRestore(pScrn);
        ZZZUnmapMem(pScrn);
    }
    pScrn->vtSema = FALSE;
    pScreen->CloseScreen = ZZZPTR(pScrn)->CloseScreen;
    return (*pScreen->CloseScreen)(scrnIndex, pScreen);
}
```

SaveScreen

SaveScreen() 0 (- 1 D
$#" 6 $ & &(B

```
static Bool
ZZZSaveScreen(ScreenPtr pScreen, int mode)
{
    return vgaHWSaveScreen(pScreen, mode);
}
```

" & 3 #" & & "$ " &
" #- 8 (& - *pScrn*
/.++ " pScrn->vtSema == TRUE

FreeScreen

FreeScreen() "(" "
% *driverPrivate* " " ("$
" "(& ## & (&(& " " A
" " # 0 1
" "(& CloseScreen()

```
static void
ZZZFreeScreen(int scrnIndex, int flags)
{
    /*
     * If the vgahw module is used vgaHWFreeHWRec() would be called
     * here.
     */
    ZZZFreeRec(xf86Screens[scrnIndex]);
}
```