Definition of the Porting Layer for the X v11 Sample Server

Susan Angebranndt, Digital Equipment Corporation

Raymond Drewry, Digital Equipment Corporation

Philip Karlton, Digital Equipment Corporation

Todd Newman, Digital Equipment Corporation

Bob Scheifler, Massachusetts Institute of Technology

Keith Packard, MIT X Consortium

David P. Wiggins, X Consortium

Jim Gettys, X.org Foundation and Hewlett Packard

X Version 11, Release 7.7

X Server Version 1.12.1

Copyright © 1994 X Consortium, Inc.

Copyright © 2004 X.org Foundation, Inc.

! "

$

%

#&' ()#*+,' - ,(.-/'/ + - *-#&(0# *+,,+1#2 ()
+12 3-1/ '4 ,' (, -5 6-'/ -1760/-18 90# 1(# 6-5-#'/ #(
#&' *+,,+1#-' () 5',7&+1#+9-6-#2)-#1')(, + +,#-70
6+, 0, (' +1/ 1(1-1),-18'5'1#% -1 1('.'1# &+66 #&' 4
7(1 (,#-05 9' 6-+96')(, +12 76+-5 /+5+8' (, (#&', 6-+9-6-
#2 *&'#&', -1 +1 +7#-(1 () 7(1#,+7# #(,# (, (#&',*- '
+,- -18),(5 (0# () (, -1 7(11'7#-(1 *-#& #&' ()#*+,' (,
#&' 0 ' (, (#&', /'+6-18 -1 #&' ()#*+,'%

63:;< /'7 = / '> 7 % 5
+ = + 7 - % =
+ - % ' = 4 ? 7 % 4 *
= 4%) - % 7 =
7 , - %

2012-04-13

Revision History

Revision 1.0	27 Oct 2004	sa
	Initial Version	
Revision 1.1	27 Oct 2004	bs
	Minor Revisions	
Revision 2.0	27 Oct 2004	kp
	Revised for Release 4 and 5	
Revision 3.0	27 Oct 2004	dpw
	Revised for Release 6	
Revision 3.1	27 Oct 2004	jg
	Revised for Release 6.8.2	
Revision 3.2	17 Dec 2006	efw
	DocBook conversion	
Revision 3.3	17 Feb 2008	aj
	Revised for backing store changes	
Revision 3.4	31 Mar 2008	efw
	Revised for devPrivates changes	
Revision 3.5	July 2010	ac
	Revised for Xorg 1.9 devPrivates changes and 1.8 CreateNewResourceType changes	

Abstract

? 4 * $
% -
4 / $ = % -
4 * @ 4
$<< $ %@ # "

<%, @ @ ($ $
%

:% = $ / %

A%= $ %

B% = / %

2 = "

C @# 4 * @ $ $ 4%

C @4 7 6 4 - @ $ %

C @4 * @
$ %

= 7
% 2 :/
% 2 $ = 4 * % #
$ =
$ = = % #
$ %

%
> % ($ % #
$

$ $ 6 ? D9 /
4 $ %

1 :;;E "
$ $:;;B % - , ? =
$ 4 %
5 = $ $
, ? 4
% # $
% *
% & %

Table of Contents

4 * %%% K
($ $ $ %%% K
/-4 6 %%%I
$, %%%I
7 = 5 %%% H
'? - %%% <<
5 (& %%% <<
(6 %%% <<
, > / $ %%% <:
1 7 7 %%% <I
, / 7 %%% <E
'$ ' + , # 7 %%%<H
) %% :;
5 5 %%% :;
7 %% :<
(( ) %%% ::
//4 6 %%% ::
- %%% :A
%%% :H
/ %%BX
8 7 ? . %%%KX
/ $ %%% I:
? %% IH
%%% XB
* $ %%%XB
* = F %%% XX
, %%% XX

The X Window System

4 * @4 @ $
$ $ %

4 % #
$ $ A:
? =% -
= ? % '$

$ % -
B %

4 = $
= % $
' = #7 - G @ @
% + G , :A: HI;;
> %

4 %)
% - @ $ %@ 9 =
$ $ %

) $ $ @# 4 * %@)
@4 * %@

Overview of the Server

$ >
% #
=% # $ "

C , > ? ?

C $ = = $?

C 5

$ J ! "

C / $ - /-4

C ((
$

C / $ / //4
$

C '? - 4 $

@ @ (//4 G
% # /-4

$ /-4
//4 % '?
$ G > $ //4%

% A
= /-4 % # (? B%
K $ //4 %
I ? ? %

DIX Layer

/-4 $ 4%
$ % #
/-4
? //4 (% #
$? %

- $ = = /-4
? % 3
= % * ? 4 $
$! /-4 % -
/-4 $ $
$ %

/-4 % '
$ $
> % #
J $ >
% 5 (//4
%

Server Resource System

4 7 $ % 7
! 4 % 7
-/ A:
$ =% * $! 7
%

Pre-Defined Resource Types

//4 $ = "

C *

C ?

C

C / $

C 7

C)

C 7

C 8 7 ?

$ @, @
@/ $, G@ @ @ @/ $ %@

% $
! %) %
% % % %
= G %
%

= "

C + $ =

C !

C + $ $ $ L*
$ M //4 $ %

/-4 = % #
//4 % 5

%) ? ? ,
?

```
(* pScreen->CreatePixmap)(pScreen, width, height, depth).
```

+ G
$ > %

$ "

```
void pScreen->MyScreenRoutine(arg, arg, ...)
```

"

```
void MyFreeRoutine(arg, arg, ...)
```

/-4G //4
%

Creating Resources and Resource Types

? -
$ > ? % '
$ $
% , $ <% 8

```
RESTYPE CreateNewResourceType(deleteFunc, char *name)
```

) %
4 , ?
? '6 ? = /# % N#
? $ <%E%O

, = < PP A<
$ % #

```
RESTYPE CreateNewResourceClass()
```

= % 7
? $ =
% ,7Q+12 %

1 $ 7
1 , # = %

- $ -/)=
7 -/%

```
XID FakeClientID(client)
    int client;
```

-/ $ $ %

$ -/ + , %

```
Bool AddResource(id, type, value)
    XID id;
    RESTYPE type;
    pointer value;
```

% - +
,)
) %

%

```
void FreeResource(id, skipDeleteFuncType)
    XID id;
    RESTYPE skipDeleteFuncType;

void FreeResourceByType(id, type, skipFree)
```

```
            XID id;
            RESTYPE type;
            Bool    skipFree;
```

) , $ G
) ? = /
) #)
,#Q1(1' %) , 9 #
$ G =)) %

Looking Up Resources

= %

```
        int dixLookupResourceByType(
            pointer *result,
            XID id,
            RESTYPE rtype,
            ClientPtr client,
            Mask access_mode);

        int dixLookupResourceByClass(
            pointer *result,
            XID id,
            RESTYPE rclass,
            ClientPtr client,
            Mask access_mode);
```

?6 = , 9 # $? %
?6 = , 9 7 $
% # Q $
? = $
> % # = $? %
> Q %

Callback Manager

>
= $ = ,I% +
= ! =
$ = % #
%

9 ?
=% 6 = 7 7 = ?
% % # =
% # = @ @
$ "

```
        CallbackListPtr ClientStateCallback;

*  $
              "


        if (ClientStateCallback) CallCallbacks(&ClientStateCallback, (pointer)clien

-               7    7   =      "


        AddCallback(&ClientStateCallback, func, data);

*  7 7   =   $ =                  "


        (*func)(&ClientStateCallback, data, client)

1         %



        Bool AddCallback(pcbl, callback, subscriber_data)
            CallbackListPtr *pcbl;
            CallbackProcPtr callback;
            pointer         subscriber_data;


+       =      Q          $     =   % 7
  =          ?  % ,    #,0'       %



        Bool DeleteCallback(pcbl, callback, subscriber_data)
            CallbackListPtr *pcbl;
            CallbackProcPtr callback;
            pointer         subscriber_data;


, $       =          $     =       % ,    #,0'
     =         %



        void CallCallbacks(pcbl, call_data)
            CallbackListPtr    *pcbl;
            pointer     call_data;


)       =              $     =           "



        (*callback)(pcbl, subscriber_data, call_data);
```

```
void DeleteCallbackList(pcbl)
    CallbackListPtr    *pcbl;
```

/ $ = %

Extension Interfaces

? -
- '? %

```
ExtensionEntry *AddExtension(name, NumEvents,NumErrors,
 MainProc, SwappedMainProc, CloseDownProc, MinorOpcodeProc)

 const char *name;  /*Null terminate string; case matters*/
 int NumEvents;
 int NumErrors;
 int (* MainProc)(ClientPtr);/*Called if client matches server order*/
 int (* SwappedMainProc)(ClientPtr);/*Called if client differs from server*
 void (* CloseDownProc)(ExtensionEntry *);
 unsigned short (*MinorOpcodeProc)(ClientPtr);
```

? % 1 '$
$? 1 '
? % 5 $!
? % 5 ?
? $?% 7 / $
$? % 5 (
/-4 $ % # /-4
5 (=
> % % ! ! %

Macros and Other Helpers

4 $?% ?
% (" ,'F0' # ,'F0' #Q -R'Q5+#7&
,'F0' #Q+#Q6'+ #Q -R' ,'F0' #Q)-4'/Q -R' 6'8+6Q1'*Q,' (0,7'
.+6-/+#'Q/,+*+96'Q+1/Q87% 0
4 $?% " * , #7 * / #7 G
4 $ % " 6 , 9 6 , 6 , 6
, , 6 %

OS Layer

$
% # (
= % # $
$ %

```
void OsInit()
```

(- J ($ = %)
> % # $ 4 $
% %

Scheduling and Request Delivery

/-4 =
$ % # =
= % >
$ % $ $
= = % # $ >
$ =
$ %

2 %

```
int WaitForSomething(pClientReady)
 int *pClientReady;
```

*)
$ % # = $
"

C # $ - 7 =

C # > =
7 ,

C +

9 *) =
=
> G /-4 * =F %

```
extern WorkQueuePtr workQueue;

if (workQueue)
 ProcessWorkQueue ();
```

- *) =
$ /-4 9 =
& %

```
void BlockHandler(pTimeout, pReadmask)
 pointer pTimeout;
 pointer pReadmask;
```

(//4
#
= , =
%

- $ # $ DD , =
= %

/-4 9 =& 9 =
& % + 9 =& "

```
void xxxBlockHandler(nscreen, pbdata, pptv, pReadmask)
 int nscreen;
 pointer pbdata;
 struct timeval ** pptv;
 pointer pReadmask;
```

? =/
/-4 9 =& %

- *) = $
= /-4 *= & %

```
void WakeupHandler(result, pReadmask)
 int result;
 pointer pReadmask;
```

(/-4% #
$ = , = =
$ % - $
> $, =
= %

/-4 *= & *= & % + *= &
"

```
void xxxWakeupHandler(nscreen, pbdata, err, pReadmask)
 int nscreen;
 pointer pbdata;
 unsigned long result;
 pointer pReadmask;
```

? =/
/-4 *= & %

- 9 =& =
= "

```
Bool RegisterBlockAndWakeupHandlers (blockHandler, wakeupHandler, blockData
 BlockHandlerProcPtr blockHandler;
 WakeupHandlerProcPtr wakeupHandler;
 pointer blockData;
```

+)+6 ' = % #
$ 9 = $
$ "

```
RemoveBlockAndWakeupHandlers (blockHandler, wakeupHandler, blockData)
 BlockHandlerProcPtr   blockHandler;
 WakeupHandlerProcPtr  wakeupHandler;
 pointer blockData;
```

+ $, 9 =+ *=
& %

= "

```
void (*BlockHandler) (blockData, pptv, pReadmask)
 pointer blockData;
 OsTimerPtr pptv;
 pointer pReadmask;
```

= ! (#
01-4 $
A: $ % + $
$ > $
$ A:

```
void AdjustWaitForDelay(pointer /*waitTime*, unsigned long /* newdelay */)
```

$ %

+ = , 9 =+ *= &
"

```
void (*WakeupHandler) (blockData, err, pReadmask)
 pointer blockData;
 int err;
 pointer pReadmask;
```

*) $ $
= % # $
4 $ *) % %

1 *) $ $
$ > > %)
$ $! > $
*) % - $ >
> /-4 $ $
? -# #2 % # *)
$ $ > > ?
%

+
%) =
$

= = $
> % # $
$ $ $ %

+ ? S ?
7 , ?
$ % # $ >
$ % # /-4
> , , >) 7 %

*) > $
$ % + /-4 "

```
ClientPtr NextAvailableClient(ospriv)
 pointer ospriv;
```

1066 % % ?
% # $ ($
7 S $ ($ % -
$ $ = %
@1 7 7 %@ 1 ? +$ 7 -) = , >
%

- $ = -
7 = > % # /-4
- '$ - 7 = > %

$ *) %
*) $ %
(G $ %

```
int AddEnabledDevice(fd)
 int fd;

int RemoveEnabledDevice(fd)
 int fd;
```

//4 J -
/ $, + -
/ $ % # $ + ' / $, $ '
/ $ 4 $ % %

Timer Facilities

4 $? %' 4
= =
$
% # # - #)
7 = # 7 #) 4 $ % % +
=
$ %

```
typedef struct _OsTimerRec *OsTimerPtr;

typedef CARD32 (*OsTimerCallback)(
 OsTimerPtr /* timer */,
 CARD32 /* time */,
 pointer /* arg */);

 OsTimerPtr TimerSet( OsTimerPtr /* timer */,
 int /* flags */,
 CARD32 /* millis */,
 OsTimerCallback /* func */,
 pointer /* arg */);
```

% # # + #) (%
(= % # # + = $
% #)
%

```
void TimerInit(void)

Bool TimerForce(OsTimerPtr /* pTimer */)

void TimerCheck(void);

void TimerCancel(OsTimerPtr /* pTimer */)

void TimerFree(OsTimerPtr /* pTimer */)
```

- ? % #) = ?
= % # 7 %
) # 7 % 7 # 7 =
$ = %

New Client Connections

$ $
% #
? #7 - $ %

$ % &
= = 4
% -) 4 =
7 4 %)
$ 7 4 $
%) 4 =
4 % 4 = $ $ /- 6+2 $
@ ";%;%@ #) = @ @ ;

4 $ %
4 4 %

# #7 7 #7
4 = @ %@ # #7
7 = #7 @ @
> #7 I;;; %

+ 4 $ #7 I;;; G =
@ = @ - %

$ $ > I;;; =
$ @ @ = % #
% # 4
$ %

+ % ' 4 $
@ %@ ' $
% @ "A%:@ : A
= % # >
= I;;A%

($
#7% # ?7 7 ? 4 %
$ $ 4
J % - $
= J $
%

- $
= % 2 % # $
=
!
% # %

2 $ J %#
$ J %#
/-4 "

```
char *
ClientAuthorized(client, proto_n, auth_proto, string_n, auth_string)
    ClientPtr client;
    unsigned int proto_n;
    char *auth_proto;
    unsigned int string_n;
    char *auth_string;
```

/-4
% - 7 + J
1066 %

+ *) % *
) /-4 1 ? +$ 7
! % /-4% 2 (

$ (
$ % /-4 7
% # $ $
7 =
J %

J $

```
void CreateWellKnownSockets()
```

$ %
= $ "

```
void ResetWellKnownSockets()
```

4 $
% %

) @7 @ 4
%

Reading Data from Clients

, > (% #
$ = $ > G >
= $ = > =% '
> % -
= > = %

```
int ReadRequestFromClient(who)
 ClientPtr who;
```

2 , , >) 7 >
@ %@ 2
<I
> % # A:
%# ? > :KI3% & $?
$ % # J
% # ? J 5+4Q,'F0' #Q -R'
4 $ % # > 1(#
/-4%

7 *) %
$ $
; = $ %

2 > >
G S > 9 % # G /-4

% (>
G %

$ 4 $ % %

Inserting Data for Clients

/-4 @ @
> %

```
Bool InsertFakeRequest(client, data, count)
    ClientPtr client;
    char *data;
    int count;

int ResetCurrentRequest(client)
    ClientPtr client;
```

-)= , >
% # >
> %) ? 1 ? +$ 7 >
% #
)+6 ' % , 7 , > @ = @
? > ? % /-4
$ $ % % $ > =
$ > % , 7 , >
-# #2 %

Sending Events, Errors And Replies To Clients

```
int WriteToClient(who, n, buf)
 ClientPtr who;
 int n;
 char *buf;
```

* #7 7 @ @% -

$ > < %# $
4 $ % %

```
void SendErrorToClient(client, majorCode, minorCode, resId, errorCode)
    ClientPtr client;
    unsigned int majorCode;
    unsigned int minorCode;
    XID resId;
    int errorCode;
```

' #7 =
> $

S . $ /-4
%

```
void FlushAllOutput()

void FlushIfCriticalOutputPending()

void SetCriticalOutputPending()
```

#
$? %) + (
G) -7 (
7 (%#
$ 4 $ % >
= >
> %

Font Support

- $ = $ %
> $ L ? M L M
%) = =
@ ? % @%) $ $ = 4) $
%# = = $
@ %@

- ,(5 ,+5 $ % 2
=% 2
= % 9) 6
? 4 $)
$ %

#
= fonts.dir 4
$ %#
4 $) $
%+
) ? ? %

Memory Management

5 7 % 4 4
4
7 % 7 7
%

% 5 =
$ % + 4 $
= ! = % - =
J = $
= %

Client Scheduling

4 $ =
%
4 $
> = $ =
%

```
Bool isItTimeToYield;
```

-# #2 $ /-4
@ @ ? % +
> /-4 $? %

- $, ,>) 7 -# #2 <;
> = %

$
% -
> > -# #2 >
$ = % (
> =
? %) ?

%

-# #2 %

$ >
"

```
int IgnoreClient (who)
 ClientPtr who;

int AttendClient (who)
 ClientPtr who;
```

- $? % *
)
% + 7 $ - 7
%

@ @ 4 "

```
Bool ClientSleep (client, function, closure)
 ClientPtr client;
 Bool  (*function)();
 pointer  closure;
```

=
= / % 1 4 >
7 *= %

```
Bool ClientSignal (client)
 ClientPtr client;
```

D 7
> => %# @ = @
"

```
ClientWakeup (client)
 ClientPtr client;
```

4 > %

Other OS Functions

```
void
ErrorF(char *f, ...)

void
FatalError(char *f, ...)
```

2 $?
?
$ % ')) ' =

% 1 ')) '
%

+) '
$ + //4 $?
$ G %

$ 4 $ %
%

DDX Layer

/-4 //4% *
($ //4 $
//4 %

//4 =
% + $ $
% 5

% #
$ %

Input

- @ @ =
% 4 $ $ " = $
% # $? $
4 - '? G ?
% # $
$ 9 / $. / $) =/ $
= 3 / $) / $ 3) =/ $ % '
$ % #
$ $ /-4%

2 //4 %
(/-4
//4 % %

Input Device Data Structures

/-4 = $ - - %
) $ $ / $, % /-4
/ $, - - % -
$ = %

/ $, 4 $ % $
$ % + / $ / $, %

+ ?'$ $ $ % / 4 %
$ = $ % +
$ $ $ %

Processing Events

//4 "

```
void ProcessInputEvents()
```

2 $ $ % /-4
? $ % 2
$ $ $ $ /-4% # $
$ /-4 //4 "

```
void DevicePtr->processInputProc(pEvent, device, count)
     xEventPtr events;
     DeviceIntPtr device;
     int count;
```

@ @ $ /-4 % /-4
- '$

% 7 $ $
$ % /-4 % 7
$ = $ $
< $ 4 - '? $
S < %

) ? - '$ =
= % - = $ = = > !
= - > %

$?'$ % *
$ $
%

2 $ $ %

) = $?'$ $ = 9 %
) ?'$ "

C " 3 3 , 9 9 ,
5 1

C 3 3 , =
+ 7-- G

C $ A:

C 4 ?

C 2

/-4%

//4
$ %

? $
$ = $ %

$ $ % -
5 1 $ $ $
$ % - $ $
8 5 '$
%

>
$ %

Telling DIX When Input is Pending

- $ /-4 = $
$ *) % - = /-4
//4 - '$ %

= $ > =G %
= -) /-4

$ % - $
- '$ /-4%

2 /-4 $ % #
"

```
void SetInputCheck(p1, p2)
 long *p1, *p2;
```

2 J /-4
=% 2 J
%

J /-4 $
% - J
- '$
$ %

< : > % +
$;
= < $ %

//4 $ - 7 = $
$ > J % - > %
4 $ >% %

```
int TimeSinceLastInputEvent()
```

//4 $ % 9 /-4 =
(=
$ $ =% # 6 - '$
%

Controlling Input Devices

2 $ $ =
$ % # $
/-4 %

```
int pInternalDevice->valuator->GetMotionProc(pdevice, coords, start, stop,
 DeviceIntPtr pdevice;
 xTimecoord * coords;
 unsigned long start;
 unsigned long stop;
 ScreenPtr pScreen;
```

2 //4 $ $
A: % -
$ % - $
J % # ?
- / $ %

* $
% 1
$ $ $
$ % 5 $ $ > G
> % + $
$ $ % + =
> $ $
8 5 %

```
void pInternalDevice->bell->BellProc(percent, pDevice, ctrl, unknown)
 int percent;
 DeviceIntPtr pDevice;
 pointer ctrl;
 int class;
```

2 = %
; <;; <;; % 7 9) =7
3) =7 4-% %

```
void pInternalDevice->somedevice->CtrlProc(device, ctrl)
 DevicePtr device;
 SomethingCtrl *ctrl;
```

2 $ =
$ % /-4 //4 >
$ %) =
%) $
% % %

Input Initialization

- J % - - -
+ - / $ $
= %

* + $ $
J % 2 $ J - 3 / $
- / $ % -
= = %

```
void InitInput(argc, argv)
    int argc;
    char **argv;
```

- - //4 J //4%
- + - / $ $ $ %

```
DevicePtr AddInputDevice(deviceProc, autoStart)
 DeviceProc deviceProc;
 Bool autoStart;
```

+ - / $ /-4 $! % $
//4 /-4 $ % +
#,0' $ J
% #
1066 $ %

1 ? = $?
$ $ % %

/-4 / $ %
//4 %

```
DevicePtr LookupKeyboardDevice()
```

6 = 3 / $ = $ %

```
DevicePtr LookupPointerDevice()
```

6 = / $ $ %

+ / $ = + - / $ "

```
Bool pInternalDevice->DeviceProc(device, action);
 DeviceIntPtr device;
 int action;
```

2 / $ $ % $ $ %
= G
% "

C /'.-7'Q-1-# + /'.-7'Q-1-# $ J
- / $ - 3 / $
@ @ $ % - J % % TU
$ /'.-7'Q-1-# $ $
% - = $ J
/-4 %

C /'.-7'Q(1 - / $ /'.-7'Q(1
$ -
$ %

C /'.-7'Q()) - / $ /'.-7'Q()) $
$ $ /-4 % # $
%

C /'.-7'Q76(' + /'.-7'Q76(' $
%

```
void InitPointerDeviceStruct(device, map, mapLength,
  GetMotionEvents, ControlProc, numMotionEvents)
 DevicePtr device;
 CARD8 *map;
 int mapLength;
 ValuatorMotionProcPtr ControlProc;
 PtrCtrlProcPtr GetMotionEvents;
 int numMotionEvents;
```

- / $ /-4 /'.-7'Q-1-#
$ %
6 4 * % 7
8 5 '$ //4 $ %

5 '$ J 8 5 '$ > %
+ <;; $ % + $
5 '$ J %

```
void InitKeyboardDeviceStruct(device, pKeySyms, pModifiers, Bell, ControlPr
 DevicePtr device;
 KeySymsPtr pKeySyms;
 CARD8 *pModifiers;
 BellProcPtr Bell;
 KbdCtrlProcPtr ControlProc;
```

2 /-4 = $ J $
/'.-7'Q-1-#% # =
4 $ % % # = =
% ?
= % 7 9 //4
$ %

Keyboard Mapping and Keycodes

* = $ $ =
% # = =
= G = =
/-4% & $
= $ E :KK $ %

= "

C + ? =

C + = = ?= =
V <% '
= % # = %

(/-4 = %

4
= % 4<< = %

6 = % *
= = @+@ @7 @ = /-4
#,0' =
$ $ "

```
Bool LegalModifier(key, pDev)
    unsigned int key;
    DevicePtr pDev;
```

Screens

/ $ %
!
% (? =
% $ =
$ $ %

Screen Hardware Requirements

>
% + $ A:
4 ? % +
. / %

4 ? % #
$? %

$ > ? % *2 -*28
!
% 7 = $ $
> ? //4 $ >
%

Data Structures

4 $ % #

, 4 $ % % #
,
%

* %
@ ? @ % #
/ % +
= $
%

? $ //4% -
$ % ?

? $ %
+ $?
%

? $ " 42 ?
R % 42 ?
% R ?
?
%

+ $ $? ?
) ? % %

? % ? ?
$ % #
%

- %
, $ % #
$ % '
? ; < : %%% % - - $
?
% # 4
$ %

Output Initialization

```
InitOutput(pScreenInfo, argc, argv)
 ScreenInfo *pScreenInfo;
 int argc;
 char **argv;
```

0 J //4 - (/-4% -
- J % - $
$ % #
$ %)
$ @ /@
$ %

2 - (J
+ J ?
$ - % 2
-
? $ %

```
int AddScreen(scrInitProc, argc, argv)
 Bool (*scrInitProc)();
 int argc;
 char **argv;
```

2 + /-4 - (
- %# J

% # $
% - <
D -)+6 '%

- "

```
Bool scrInitProc(iScreen, pScreen, argc, argv)
 int iScreen;
 ScreenPtr pScreen;
 int argc;
 char **argv;
```

? G ; J <
% , % $
% 2 J #,0')+6 '
J
? %

$ J % -
$ % - $
$
J - (J
% - $ %

J
%
%

Region Routines in the ScreenRec

+
42 ? % 2 = ?
+1/ (,%

+ >
? % , @ @
= % # $ 24 % 0
$ =
% #
, % #
4 $ % % # 4 $ % %
//4 $
%

J
= % +
=
%) $, ,
%
? 4
, , %

5 "
% # $ =

% 5 = %

```
RegionPtr pScreen->RegionCreate( rect, size)
 BoxPtr rect;
 int size;

macro: RegionPtr RegionCreate(rect, size)
```

, 7 (1' % # $
?
% 2
! J % #
J G $
? J $ > % 2
% 2
J %

```
void pScreen->RegionInit (pRegion, rect, size)
 RegionPtr pRegion;
 BoxPtr  rect;
 int  size;

macro: RegionInit(pRegion, rect, size)
```

8 $? $
= ,
7 % # $ $
%

```
Bool pScreen->RegionCopy(dstrgn, srcrgn)
 RegionPtr dstrgn, srcrgn;

macro: Bool RegionCopy(dstrgn, srcrgn)
```

, 7
G #,0')+6 ' %

```
void pScreen->RegionDestroy( pRegion)
 RegionPtr pRegion;

macro: RegionDestroy(pRegion)
```

, / %

```
void pScreen->RegionUninit (pRegion)
 RegionPtr pRegion;

macro: RegionUninit(pRegion)
```

) $?
, - $, 7 % *
, , - %

```
Bool pScreen->Intersect(newReg, reg1, reg2)
 RegionPtr newReg, reg1, reg2;

macro: Bool RegionIntersect(newReg, reg1, reg2)

Bool  pScreen->Union(newReg, reg1, reg2)
 RegionPtr newReg, reg1, reg2;

macro: Bool RegionUnion(newReg, reg1, reg2)

Bool  pScreen->Subtract(newReg, regMinuend, regSubtrahend)
 RegionPtr newReg, regMinuend, regSubtrahend;

macro: Bool RegionUnion(newReg, regMinuend, regSubtrahend)

Bool pScreen->Inverse(newReg, pReg,  pBox)
 RegionPtr newReg, pReg;
 BoxPtr pBox;

macro: Bool RegionInverse(newReg, pReg,  pBox)
```

$ %
?
$ % 2
,
%

$ 5
, %

$ 9 ? %
+ @ $ @ = ?
% - ?
?%

' $ #,0' %

```
void pScreen->RegionReset(pRegion, pBox)
 RegionPtr pRegion;
 BoxPtr pBox;

macro: RegionReset(pRegion, pBox)
```

, , J
%

```
void  pScreen->TranslateRegion(pRegion, x, y)
 RegionPtr pRegion;
 int x, y;

macro: RegionTranslate(pRegion, x, y)
```

, $ V? ? V
%

```
int  pScreen->RectIn(pRegion, pBox)
 RegionPtr pRegion;
 BoxPtr pBox;

macro: int RegionContainsRect(pRegion, pBox)
```

, - -1 (0# +,#
?
% # 4 $
% %

```
Bool pScreen->PointInRegion(pRegion, x, y, pBox)
 RegionPtr pRegion;
 int x, y;
 BoxPtr pBox;

macro: Bool RegionContainsPoint(pRegion, x, y, pBox)
```

- , ? % -
9 ? ,
% - % /
$ %

/-4 = $
$ %) /-4
$ % / $

= % + - , $ $
$? $ % # 9 ? = = ?G /-4
- , ?%

```
Bool pScreen->RegionNotEmpty(pRegion)
 RegionPtr pRegion;

macro: Bool RegionNotEmpty(pRegion)
```

, 1 '
? %

```
void pScreen->RegionEmpty(pRegion)
 RegionPtr pRegion;

macro: RegionEmpty(pRegion)
```

, ' %

```
BoxPtr pScreen->RegionExtents(pRegion)
 RegionPtr pRegion;

macro: RegionExtents(pRegion)
```

, '?
% #
%

```
Bool pScreen->RegionAppend (pDstRgn, pRegion)
 RegionPtr pDstRgn;
 RegionPtr pRegion;

macro: Bool RegionAppend(pDstRgn, pRegion)

Bool pScreen->RegionValidate (pRegion, pOverlap)
 RegionPtr pRegion;
 Bool *pOverlap;

macro: Bool RegionValidate(pRegion, pOverlap)
```

$ J
! % #
, + $, .
= $

% ($ #,0'
$ G)+6 ' %

```
RegionPtr pScreen->BitmapToRegion (pPixmap)
 PixmapPtr pPixmap;

macro: RegionPtr BitmapToRegion(pScreen, pPixmap)
```

8 $ < ? $
? < ? ; % #
1066 %

```
RegionPtr pScreen->RectsToRegion (nrects, pRects, ordering)
 int nrects;
 xRectangle *pRects;
 int ordering;

macro: RegionPtr RegionFromRects(nrects, pRects, ordering)
```

8 $
% (
% + $ >

%

```
void pScreen->SendGraphicsExpose(client,pRegion,drawable,major,minor)
 ClientPtr client;
 RegionPtr pRegion;
 XID drawable;
 int major;
 int minor;
```

8 '? 8 '? $
% - 1066 1 '?
$ %

Cursor Routines for a Screen

+ $ $ % #
@4@ $ % *
%

) 7 , %
7 9
=
7 , =

% # 7 9 7 ,
=%
= <
= $
% (J
= % # $
> % 4 J
? 1 ? 5 ? % #
J %

7 9 4 $ % $
$ =% #
%

* @ J @ % + J
$
$
$ $ 7 , 7 9
$ %-
$ $ J 7 9 $
= J
%) $ $
"

```
pCursor->devPriv[pScreen->myNum]
pCursor->bits->devPriv[pScreen->myNum]
```

#
$ G $
%

2 "

```
Bool pScreen->RealizeCursor( pScr, pCurs)
 ScreenPtr pScr;
 CursorPtr pCurs;

Bool pScreen->UnrealizeCursor( pScr, pCurs)
 ScreenPtr pScr;
 CursorPtr pCurs;
```

, J 7 0 J 7 J
J $
/-4 % # $ $
% # = 7
% # 9
$ % 2 0 J
7 G %

```
Bool pScreen->DisplayCursor( pScr, pCurs)
 ScreenPtr pScr;
 CursorPtr pCurs;
```

/ 7 $ %
- /-4 $ $
% #
%

```
void pScreen->RecolorCursor( pScr, pCurs, displayed)
 ScreenPtr pScr;
 CursorPtr pCurs;
 Bool displayed;
```

, 7 //4 7 $
% -
$ % * , J
7 $ % + $, 7
J J % G
$ 0 J 7 , J 7 #,0'
%

```
void pScreen->ConstrainCursor( pScr, pBox)
 ScreenPtr pScr;
 BoxPtr pBox;
```

7 7 9 ?%
/-4 $
7 7
$ $ % # $
? ! = /-4 =%

```
void pScreen->PointerNonInterestBox( pScr, pBox)
 ScreenPtr pScr;
 BoxPtr pBox;
```

1 - 9 ? /-4 $
$ $ $ % -
% #
$ $
% # % - $ /-4
> $ % #
% 1 /-4 1 -
9 ?%

```
void pScreen->CursorLimits( pScr, pCurs, pHotBox, pTopLeftBox)
 ScreenPtr pScr;
 CursorPtr pCurs;
 BoxPtr pHotBox;
 BoxPtr pTopLeftBox; /* return value */
```

7 6 ?
$ $
7 ? /-4
& 9 ?% # /-4 G //4%

```
Bool pScreen->SetCursorPosition( pScr, newx, newy, generateEvent)
 ScreenPtr pScr;
 int newx;
 int newy;
 Bool generateEvent;
```

7 $! =
$ $ > = % # *
> % - '$ # $
- '$ / $
S - % - $
$ %

```
void NewCurrentScreen(newScreen, x, y)
    ScreenPtr newScreen;
    int x,y;
```

- ? $ $
/-4 % 2
1 7 ?
%

Visuals, Depths and Pixmap Formats for Screens

@ @ ? %

@ ? @ ? $
B E <I :B A:% - ?
R %) ? ? R
E ? %

+ @ ? @ @ ? @
? $ %) $ $
? % # ? $
? %) ?
<I A: W

) % 2
%
? ! $
G %

+ @ @
= = = %

+ @$ @ ?
$ % ' $ =
? $
%
% ' $ -/
% 1 Q G
$ E ? :KB
$ %

' $ A: $ -/
$ $ % 7 $
%

= %
%

C # ? $ = Q
%

C # # ? $ =
= =
Q %

C 8 # ? $ = Q
$ %

- = ? ?
$ = % #
? "

C 8 # ? $ $ $
? %

C 8 # ? $ = $ $
$? %

C 7 # ? $? =
? %

C 7 # ? $ =

? $ %

C # 7 ' ? $
? %

C / 7 ' ? $ %
' $ $ =
%) ? $ =

$ $
%

& ? "

C + < ? 8 %

C + : ? =
$ 7 %

C + ?
7 %

C - = ? ? $
7 %

+ E ? % # E :KI
= :B E %
:KI <I %
% # =
%

- ? % 2 $
$ =
= %) $
= ? %) =
% 9
:KI =
=
$ % 7
= % + > 7
4 7 %

2 $ = $ $ %
* J $. ,
= %

/ , . , , %
- (G 4 =
$ J $ %

Colormaps for Screens

+ $? $
%

/ $
% + $
$ % (
% +
%

J J % -
? % -
; $ % 5
% /
%

Colormap Routines

2 $
% 2
, % # $
% %

```
Bool pScreen->CreateColormap(pColormap)
 ColormapPtr pColormap;
```

/-4 7 7
! % -
//4 J %
J % #
)+6 ' % 1
$ $
%
$ %

```
void pScreen->DestroyColormap(pColormap)
 ColormapPtr pColormap;
```

/-4) 7

% - //4 %

```
void pScreen->InstallColormap(pColormap)
 ColormapPtr pColormap;
```

- 7 =
7 % - =
%

- =

%
%

```
void pScreen->UninstallColormap(pColormap)
 ColormapPtr pColormap;
```

0 7 $ 7 7 S %

7 % - 7 % -
> 7 1 $ //4 % #
* =# % # /-4 # 1 5
# 1 5 # 8 5 $
* =# % # 4 $? % %

```
int pScreen->ListInstalledColormaps(pScreen, pCmapList)
 ScreenPtr pScreen;
 XID *pCmapList;
```

6 - 7 7 6
% 7 6 J
5 ?- 5 %

```
void pScreen->StoreColors (pmap, ndef, pdefs)
 ColormapPtr pmap;
 int ndef;
 xColorItem *pdefs;
```

7 % #

% 1 % (
?7 - %
& $ $
$ % -
% # $ G
% ?7 - 4 % %

```
void pScreen->ResolveColor(pRed, pGreen, pBlue, pVisual)
 unsigned short *pRed, *pGreen, *pBlue;
 VisualPtr pVisual;
```

8 $ > , $ 7
$ % - $
% ,
% # $
? $ Q Q . %

' $ $; IKKAK% # =
%

) ? ?
= I :KI3
$ I %

$ I ? $
IKKAK % # = $
I =
E %

Initializing a Colormap

* > $
7 7 /-4 $ = % #

? % - 9 7
7 , //4 7 7 % #
J % - =

? % -
= - / 9 = ?
* ? $ % (
= $ %

2 + 7 = ? ?
% -
J S7 7 + 7
? $ $ = ?
? % - J
$ > + 7 ? $
% # ? $ %

% # /-4
+ 7 %

```
int AllocColor (pmap, pred, pgreen, pblue, pPix, client)
    ColormapPtr         pmap;
    unsigned short      *pred, *pgreen, *pblue;
    Pixel               *pPix;
    int                 client;
```

= <I $?
$ % # ? $?
% -
= $
? % # $ = $ D ?
+ 7 % - ? =
> ; D ? ;% # ?
%

+ 7 = == 7
$ J % # $
%

- + 7 =
= 7 % #
$ %

Fonts for Screens

+ % '
$ J !
% $ $ J $
@ @ @ %@ 4 E <I
$:KI % 8
$ %

+ 7-- ?
$ % # ?
9/) ? + :%< 9 /
) + - % %

+ $
9/)% -
%

2
$ %
# $ %

- 9/) $
% -
% (
%

+ $ % # $
% -
4<< % - $
$ %

Portable Compiled Format

+ $ % -
$ 9/) 7)
7)% #
7) 7)
% 9 >
7) $
% # > %

%
%

Font Realization

' $ @
J @ % # $
%

+) , 4 $? % $
$ % *
@ J @ % + $
% # //4
$ $ %

```
Bool pScreen->RealizeFont(pScr, pFont)
 ScreenPtr pScr;
 FontPtr pFont;

Bool pScreen->UnrealizeFont(pScr, pFont)
 ScreenPtr pScr;
 FontPtr pFont;
```

, J) 0 J) ?
% #
( ) 7) > % # $
% $ %

Other Screen Routines

2 $ 4 $
% "

C 8 - / $ %

C 8 ? %

C $? *
/ %

C # 7 87 8 7 ? %

```
void pScreen->QueryBestSize(kind, pWidth, pHeight)
 int kind;
 unsigned short *pWidth, *pHeight;
 ScreenPtr pScreen;
```

F 9 J J
> % = 7 #
4% %) 7 ?
%) #
$ * &
$ > $ % # $
4 $ % %

```
pScreen->SourceValidate(pDrawable, x, y, width, height)
 DrawablePtr pDrawable;
 int x, y, width, height;
 unsigned int subWindowMode;
```

. 7 + 7 $
. % - =

$. $ =% 7 .
$
$ % ?
$ % * 5
87 %

```
Bool pScreen->SaveScreen(pScreen, on)
 ScreenPtr pScreen;
 int on;
```

$ $ *) %
$ %

```
Bool pScreen->CloseScreen(pScreen)
    ScreenPtr pScreen;
```

* $ %

```
Bool pScreen->CreateScreenResources(pScreen)
    ScreenPtr pScreen;
```

- 1066 $ J
$ > $
L* $ M % 2

+ > $
% * $ $
$ % # #,0' %

Drawables

+
? %

'
J % #
/,+*+96'Q -45+ /,+*+96'Q*-1/(* 01/,+*+96'Q*-1/(*%
+ - (% #
> $
87% #
%
G ? ; ;
? % J
J % #
? $ %

9 ? , * ,
% ? $ $ $
? $
//4 % 9 ? $ $ $
//4 L* $ M
% # >
G
$ %

? 4 $
? % % # 4 $
% %

Pixmaps

+ ?
$ % -
% #
? $? $ % #
? % +
J ? G ? %
? G
? $ %

1 ? %

$?
% 5 $
$ % 5
%

- $ $ L* $ M
? $ $ %
+ $ %
$3 $?
% % - $?
$3 ? % $ $
? %

+ ? %

```
PixmapPtr pScreen->CreatePixmap(pScreen, width, height, depth)
 ScreenPtr pScreen;
 int width, height, depth;
```

, ? J > % -
? , % #
<% - J ?
$ $?
1066% - ? G
1066% 4 $? % $ %

```
Bool pScreen->DestroyPixmap(pPixmap)
 PixmapPtr pPixmap;
```

, @ @ ? % -
J ? , $
$ = % - #,0'% 4 $? %
$ %

```
Bool
pScreen->ModifyPixmapHeader(pPixmap, width, height, depth, bitsPerPixel, de
 PixmapPtr pPixmap;
 int width;
 int height;
 int depth;
 int bitsPerPixel;
 int devKind;
 pointer pPixData;
```

= ? J ? ,
% ? $ $
S7 ? J $?
% ?/ ? G
? S $ $ % %# #,0'
% 4 $ % $ %

```
PixmapPtr
GetScratchPixmapHeader(pScreen, width, height, depth, bitsPerPixel, devKind
 ScreenPtr pScreen;
 int width;
 int height;
 int depth;
 int bitsPerPixel;
 int devKind;
 pointer pPixData;

void FreeScratchPixmapHeader(pPixmap)
 PixmapPtr pPixmap;
```

//4 /-4
?
$ $ % #
?/
? S $ $ % %#
? , % - 8 ? & $?
$? ? 1066%
?
) ? & %

Windows

+ $ $ % /-4

% *
%
%
= G
% ' %

//4 /-4 >
% # >
$ %

* ? $ % *
? /-4 ? $
%

+ $ $
G @ = @% *
? > = $?
? $ = %
'$ =
! % +
$ $ = =
= % - =
$ %

(= > ?

% 1 $
% # =
=
? $ %

- 4 = $
% # $ = @ @
7 ? % + * 5
+ > $ % & $ $?
%

* >
$ % / /-4 //4

$? % # $ =
= % - //4 ? /-4 ?
$ % -
? %

- J ? $
% # > @8 $ @ =
$ J %)
$
J ? $ $ $
% - =

$?
% 8 $ %

$ "

C 6 # !
% 6
$ % #
- -
1 7 9 7 * %

C 7 7 *
$
$ %

5 /-4 %

Window Procedures in the ScreenRec

2
%

$ $ @ @ % & $
$ = % #
, $
% #
//4 G /-4%

```
Bool pScreen->CreateWindow(pWin)
 WindowPtr pWin;
```

= /-4 % - @*
* , @ $ $ = %

4 $ % $ %

```
Bool pScreen->DestroyWindow(pWin);
 WindowPtr pWin;
```

= /-4 % -
$ $ = =
%

4 $ % $ %

```
Bool pScreen->PositionWindow(pWin, x, y);
 WindowPtr pWin;
 int x, y;
```

= /-4 $ J %- $
$ $ J %)
//4 = ? = =

%# $ $ $
? 7 *
%

4 $ % $ %

```
Bool pScreen->RealizeWindow(pWin);
 WindowPtr pWin;

Bool  pScreen->UnrealizeWindow(pWin);
 WindowPtr pWin;
```

= /-4 = $ =
$ %- $ $

$ % , J * 1(# =
G 0 J * 1(# ?
$ G = /-4% /-4 $
* 9 = * 9 %

```
Bool pScreen->ChangeWindowAttributes(pWin, vmask)
 WindowPtr pWin;
 unsigned long vmask;
```

7 * + $ /-4
J = $
%# $ %-
> = = %
* , %

```
int pScreen->ValidateTree(pParent,  pChild, kind)
 WindowPtr pParent, pChild;
 VTKind kind;
```

. #
%# $ %# $
$ 4 $ $ %%# $ %

```
void pScreen->PostValidateTree(pParent,  pChild, kind)
 WindowPtr pParent, pChild;
 VTKind kind;
```

- 1066 /-4 . #
% # % #
$ 1066%

```
void pScreen->WindowExposures(pWin, pRegion, pBSRegion)
 WindowPtr pWin;
 RegionPtr pRegion;
 RegionPtr pBSRegion;
```

* '? ? $
% , 9 ,
= % ? $
? $ =
? $ % # ?
$ % #
$ % # $ $
4 $? % %

```
void pScreen->ClipNotify (pWin, dx, dy)
 WindowPtr pWin;
 int dx, dy;
```

* $
$ % *
$ % ?
$ %

Window Painting Procedures

- $
% - $
= =
E <I A: ? > %

```
void pScreen->ClearToBackground(pWin, x, y, w, h, generateExposures);
 WindowPtr pWin;
 int x, y, w, h;
 Bool generateExposures;
```

7 #9 = >
% # >
'? %

- '? $
% # $

? /-4 &
'? * '?
%

- '?
= % 7 #9 = =
> * 9 = % -
J %

$ 4 $ % %

```
void pScreen->CopyWindow(pWin, oldpt, oldRegion);
 WindowPtr pWin;
 DDXPointRec oldpt;
 RegionPtr oldRegion;
```

7 * $ $?
% - //4
* $ %

% ,
% # * , % ,
%

7 + ? %)
% #

%
= %

? ,
* S 7 % * S 7
%

$ 4 $ % %

Screen Operations for Multi-Layered Framebuffers

$
% % $ @ @
% - $
$ % % #
- %

```
   void pScreen->MarkWindow(pWin)
WindowPtr pWin;
```

? 5 =* $? $
% #

$ * S$. #
$ %

Bool pScreen->MarkOverlappedWindows(parent, firstChild, ppLayerWin)
WindowPtr parent;
WindowPtr firstChild;
WindowPtr * ppLayerWin;

? 5 =* $? $
%- " 6 *
$ = .
# %- 6 * UU * %

Bool pScreen->ChangeSaveUnder(pLayerWin, firstChild)
WindowPtr pLayerWin;
WindowPtr firstChild;

? 7 $0 7 = $0 $ $?
$ % 6 *
6 * 5 =($ * %#
= $ =
%- #,0' 7 $0
= %

void pScreen->PostChangeSaveUnder(pLayerWin, firstChild)
WindowPtr pLayerWin;
WindowPtr firstChild;

? /7 $0 $? $
%# ! =
7 $0 %

void pScreen->MoveWindow(pWin, x, y, pSib, kind)
WindowPtr pWin;
int x;
int y;
WindowPtr pSib;
VTKind kind;

? 5$* $? $
%# $? % = .#5$
$.#(%

```
   void pScreen->ResizeWindow(pWin, x, y, w, h, pSib)
WindowPtr pWin;
int x;
int y;
unsigned int w;
unsigned int h;
WindowPtr pSib;
```

? + J * $?
$ % # $? % # J $
%

```
   WindowPtr pScreen->GetLayerWindow(pWin)
WindowPtr pWin
```

* %

```
   void pScreen->HandleExposures(pWin)
WindowPtr pWin;
```

? & '? $? $
% # . #
$? %

```
   void pScreen->ReparentWindow(pWin, pPriorParent)
WindowPtr pWin;
WindowPtr pPriorParent;
```

% +
*
$ % # 1066%

```
   void pScreen->SetShape(pWin)
WindowPtr pWin;
```

? $? $
%# $
%

```
   void pScreen->ChangeBorderWidth(pWin, width)
```

```
        WindowPtr pWin;
        unsigned int width;
```

? 7 9 * $?
$ % # $ %

```
           void pScreen->MarkUnrealizedWindow(pChild, pWin, fromConfigure)
        WindowPtr pChild;
        WindowPtr pWin;
        Bool fromConfigure;
```

J 0
J # % 7 J *
7 $ 0 J # %

Graphics Contexts and Validation

? 87 $ =
? $
?
%

- > $?
% # =
$
? %
+ = = % + 87 = =%

0 = @ %@ '$
%

87 $ 87
87 % # 87

? % # //4
$ = $
$!
% # 87
@ @ 87 $ $ 87
% + $ $ 87
87 %
% %

//4 /-4 87%)
? //4
87 ! % -
? =
$
%

+ 4 $ % %

+ 87 $ $ //4
L* $ M %

/-4 $ 87 7 87 7 87
7 874-/ 7 87 7 , /) 87%

```
GCPtr CreateGC(pDrawable, mask, pval, pStatus)
    DrawablePtr pDrawable;
    BITS32 mask;
    XID *pval;
    int *pStatus;

int ChangeGC(client, pGC, mask, pUnion)
    ClientPtr client;
    GCPtr pGC;
    BITS32 mask;
    ChangeGCValPtr pUnion;

int ChangeGCXIDs(client, pGC, mask, pC32)
    ClientPtr client;
    GCPtr pGC;
    BITS32 mask;
    CARD32 *pC32;

int CopyGC(pgcSrc, pgcDst, mask)
    GCPtr pgcSrc;
    GCPtr pgcDst;
    BITS32 mask;

int SetClipRects(pGC, xOrigin, yOrigin, nrects, prects, ordering)
    GCPtr pGC;
    int xOrigin, yOrigin;
    int nrects;
    xRectangle *prects;
    int ordering;

SetDashes(pGC, offset, ndash, pdash)
    GCPtr pGC;
    unsigned offset;
    unsigned ndash;
    unsigned char *pdash;

int FreeGC(pGC, gid)
    GCPtr pGC;
    GContext gid;
```

+ $ 87
% #
% # /-4 87 "

```
GCPtr GetScratchGC(depth, pScreen)
    int depth;
    ScreenPtr pScreen;

FreeScratchGC(pGC)
    GCPtr pGC;
```

+ ? 87
%

- 87 $ $
= 8 87
7 87 7 87% # $
$ %

```
GCPtr CreateScratchGC(pScreen, depth)
    ScreenPtr pScreen;
    int depth;
```

1066 87 % # 87
) 87%

Details of Operation

+ J 87 % + 87
87 87 $ 4 $
% %) 87 $
$ $ % # $ * =
%

* > 87
$ 87 % #
% # 7 87 87 % #
$ 7 0% //4
= 87 % #
87 > $ 87%

. 87 87
% /-4 . 87 87 % //4
% //4 " <
$ $ = 87G : G A 87 %

87
87 % #
6 7 % #
87 /-4
$
87%

- $
;% # //4
< 87 $? 87 % /-4

> 87 87
/ 7 > %

GC Handling Routines

, 7 87 %

```
Bool pScreen->CreateGC(pGC)
 GCPtr pGC;
```

87 % -
1(# 87 G /-4 %

87 G
87 $
87 % %

87 $ X $ $ "

```
pGC->funcs->ChangeGC(pGC, changes)
 GCPtr pGC;
 unsigned long changes;
```

87 87 %
= 87 > %

7 87 $
87 $
%

```
pGC->funcs->ValidateGC(pGC, changes, pDraw)
 GCPtr pGC;
 unsigned long changes;
 DrawablePtr pDraw;
```

. 87 /-4 ! 87
87 % - $
$ 87 $
$ 87% - $
87 %

- . 87 =

= $ 87% - $
$ %
2 7
1 7 9 7 /-4% + ?
4 $ % %

= ?
G =
%

$ %

```
pGC->funcs->CopyGC(pGCSrc, mask, pGCDst)
 GCPtr pGCSrc;
 unsigned long mask;
 GCPtr pGCDst;
```

/-4 87 87%
= 87
$ $ 87%

```
pGC->funcs->DestroyGC(pGC)
 GCPtr pGC;
```

87 87
% # ?
%

GC Clip Region Routines

87 7 > %
87 $ % # ?=
?
4 $ % -
$? 4 87 >
= $ $ = $
= % # $ 4 $ 7
87 > %

```
void pGC->funcs->ChangeClip(pGC, type, pValue, nrects)
 GCPtr pGC;
 int type;
 char *pValue;
 int nrects;
```

$ % # 87
87 $ $ % -
7#Q1(1' % - 7#Q01 (,#'/ 7#Q29+1/'/
7#Q249+1/'/ . % -
7#Q,'8-(1 . , , % -
7#Q -45+ . ? % # 7#Q1(1' %
4 $ % % #

% % ? = =
87 5 = % #
4 , /
? / ?
% /-4 5- ?
$ = 87% #
= %

- $
$
= % # $ =
$ %

```
void pGC->funcs->DestroyClip(pGC)
 GCPtr pGC;
```

$ % # 87
87 $ $ % # # 87
7#Q1(1'% - $ 1066
7
7 $ J %

```
void pGC->funcs->CopyClip(pgcDst, pgcSrc)
 GCPtr pgcDst, pgcSrc;
```

= 5 = # / %
- $ 5 = / % # =
=
= % # >
? = = %

Drawing Primitives

4 $
$ / $ % #
? % 2
<I % # <I %

5 <I 87 $ % + $
+66 50 # $

% + $ $
<I J

87 %

$ =
$ $ % 5 $
= %

= ?
= @ ? , @ ? % #
$?
$ % - $
=
$ %

GC Components

87 $ %
$ 8 - 87
% $
? 4 > 4
%

+66 50 # 76- %
. 87
> =
% # ? $ % #
?
%

+ =% # =
$ G < =
%

+ ? - # ? % #
7 <I %

+ ? 7 +
= ? $ % ' ? $ A: % #
= $
? $ % / !
? $ %

6 , +
! % 6 ? % #
= % #
, 9 % # ! ! !
5 , 9 $ % *
% 4 %

R 1(# J G
J
! % # @ @ ? % #
> %

6 J
$! $
% # $?
$
? %

)) ,) +

$ % #) +
%

#? @
%@ # >
% <
; ?
= > % + ?
87 % #
J
J E?E A:?A:% #
? %

4 % # 7 87
> $ %

The Primitives

/ $ "

```
RegionPtr pGC->ops->CopyArea(src, dst, pGC, srcx, srcy, w, h, dstx, dsty)
 DrawablePtr dst, src;
 GCPtr pGC;
 int srcx, srcy, w, h, dstx, dsty;
```

7 + ?
% #
$ % 1 > J
J % & $ $
%

- 87 S '? #
$ $

1066 %) $

= % # & '?
$ $
S * 9 = %

)
% + ? ?
$
? $ %
(=
? $ %

+ ? 7 + 4 $ % %

```
RegionPtr pGC->ops->CopyPlane(src, dst, pGC, srcx, srcy, w, h, dstx, dsty,
```

```
DrawablePtr dst, src;
GCPtr pGC;
int srcx, srcy, w, h, dstx, dsty;
unsigned long plane;
```

7
% 9 ?
% #
? ?
= % + 7 +
7 %

+ ? 7 4 $ % %

```
void pGC->ops->PolyPoint(dst, pGC, mode, n, pPoint)
DrawablePtr dst;
GCPtr pGC;
int mode;
int n;
DDXPointPtr pPoint;
```

? $
% (
$ $ % 1
= %

'? 4 $ %
4 $ % %

```
void pGC->ops->Polylines(dst, pGC, mode, n, pPoint)
DrawablePtr dst;
GCPtr pGC;
int mode;
int n;
DDXPointPtr pPoint;
```

$ %
R 1(# J G
? J
! % $ (
$ %

'? * 6 * / %
R 6 J % %

```
void pGC->ops->PolySegment(dst, pGC, n, pPoint)
```

```
DrawablePtr dst;
GCPtr pGC;
int n;
xSegment *pSegments;
```

G
$ J G %

+ ? % %

```
void pGC->ops->PolyRectangle(dst, pGC, n, pRect)
 DrawablePtr dst;
 GCPtr pGC;
 int n;
 xRectangle *pRect;
```

, %

+ ? , 4 $ % %

```
void pGC->ops->PolyArc(dst, pGC, n, pArc)
 DrawablePtr dst;
 GCPtr pGC;
 int n;
 xArc*pArc;
```

+ %
%

'? R + 4 $ J %
+ 4 $ % %

```
void pGC->ops->FillPolygon(dst, pGC, shape, mode, count, pPoint)
 DrawablePtr dst;
 GCPtr pGC;
 int shape;
 int mode;
 int count;
 DDXPointPtr pPoint;
```

)
% - ?
% # J G
$? <E; $?
<E; ? % 2
%

$ (
$ %

+ ?) 4 $ % %

```
void pGC->ops->PolyFillRect(dst, pGC, n, pRect)
 DrawablePtr dst;
 GCPtr pGC;
 int n;
 xRectangle *pRect;
```

) , %

'?) , 4 $ %
) , 4 $ % %

```
void pGC->ops->PolyFillArc(dst, pGC, n, pArc)
 DrawablePtr dst;
 GCPtr pGC;
 int n;
 xArc *pArc;
```

) +
%

+ ?) + 4 $ % %

```
void pGC->ops->PutImage(dst, pGC, depth, x, y, w, h, leftPad, format, pBinI
 DrawablePtr dst;
 GCPtr pGC;
 int x, y, w, h;
 int format;
 char *pBinImage;
```

- ? % # ?
4 9 42 ? R ?
% 4 % 2

%' ? $
? 9 %- 9 =
%

+ ? - 4 $ % %

```
void pScreen->GetImage(src, x, y, w, h, format, planeMask, pBinImage)
```

```
DrawablePtr src;
int x, y, w, h;
unsigned int format;
unsigned long planeMask;
char *pBinImage;
```

8 - %
$?
% 9 -
%

? 4 8 - >
/-4 = J > %-
R ? R) G
; 5 = ? $
> ;% - 42 ? = $
G 9
42 ? %

+ ? 8 - 4 $ % %

```
void pGC->ops->ImageText8(pDraw, pGC, x, y, count, chars)
 DrawablePtr pDraw;
 GCPtr pGC;
 int x, y;
 int count;
 char *chars;
```

- #?E ?%# ? G =
%#
?%

+ ? - #?E 4 $?%%

```
int pGC->ops->PolyText8(pDraw, pGC, x, y, count, chars)
 DrawablePtr pDraw;
 GCPtr pGC;
 int x, y;
 int count;
 char *chars;
```

#?E = = - #?E ?
?% 4 %

+ ? #?E 4 $?%%

```
int pGC->ops->PolyText16(pDraw, pGC, x, y, count, chars)
```

```
DrawablePtr pDraw;
GCPtr pGC;
int x, y;
int count;
unsigned short *chars;

void pGC->ops->ImageText16(pDraw, pGC, x, y, count, chars)
DrawablePtr pDraw;
GCPtr pGC;
int x, y;
int count;
unsigned short *chars;
```

@E@ $? <I
%

= %
E <I $ $ $ = $
= G = $ 7 -
8 % 8 $
7 - $
%

<I $:
> ? 7 - %
' ? $ G 7 -
?%

'? #? <I - #? <I 4 $
? % %

4 %

= 87 6 &
% - $
$? Y 1"/Q6-1'&'6 ',
%

Pixblit Procedures

/ $ $ % (
$ % -
$ T ? % #
? $ %

? $ @ @ % +
%

#
< A: ?
% - ?
% - ? $ J
/ $ % - =
? $ $ %

- $ @ @
% - $ $
? %
- $ / $
/ $ % '$
/ $ $
%

/-4
% # ?
% # %

$ $
$ %)
/-4 =
% -
?
%

=
. 87 % 2 =
? % # $ %

+) % #

% , (87% #
; ; %

& $ $
$ % - # 87
7 87 . 87
% - $ %
# #,0'
% % $
G $
> %
)+6 ' $
? $?
% 7 $ 8
? % # #
87 ? ? $ =
? ? =
$ > %

+ 87
? % - J
? $ %

? = %
? % # 8 % # ?
%

- 8 ? $
% -
$ > %

Span Routines

) $ @ %@ +
J ? % -
? %

' = 87

%

```
void pGC->ops->FillSpans(dst, pGC, nSpans, pPoints, pWidths, sorted)
 DrawablePtr dst;
 GCPtr pGC;
 int nSpans;
 DDXPointPtr pPoints;
 int *pWidths;
 int sorted;
```

) J ?
% $ 87% # G
* % -
= J %

87 " (7 %

87 " ? G (
G (? G (?
? (> %

```
void pGC->ops->SetSpans(pDrawable, pGC, pSrc, ppt, pWidths, nSpans, sorted)
 DrawablePtr pDrawable;
 GCPtr pGC;
 char *pSrc;
 DDXPointPtr pPoints;
 int *pWidths;
 int nSpans;
 int sorted;
```

) * /
87% -
% # ?
% # ? ?
$ % - %

87 " (7

$? 87%
? $ > =
$
%

```
void pScreen->GetSpans(pDrawable, wMax, pPoints, pWidths, nSpans)
 DrawablePtr pDrawable;
 int wMax;
 DDXPointPtr pPoints;
 int *pWidths;
 int nSpans;
 char *pDst;
```

) 8
* % ' $ % #
1066 %

8 $
$ % 7 "
4 % % $
> $ 8
% 7 $ "
; ;
> $
% 7 $) $
# ? %

#
8 ?
%

Glyph Routines

8 $? > %

2 $ % 2 $ G
% + ? =
$ %

```
void pGC->ops->PolyGlyphBlt(pDrawable, pGC, x, y, nglyph, ppci, pglyphBase)
 DrawablePtr pDrawable;
 GCPtr pGC;
 int x , y;
 unsigned int nglyph;
 CharInfoRec **ppci;  /* array of character info */
 pointer unused;   /* unused since R5 */
```

87 " (7 %

87 " ? G (
G (? G (?
? (> %

```
void pGC->ops->ImageGlyphBlt(pDrawable, pGC, x, y, nglyph, ppci, pglyphBase
 DrawablePtr pDrawable;
 GCPtr pGC;
 int x , y;
 unsigned int nglyph;
 CharInfoRec **ppci; /* array of character info */
 pointer unused;  /* unused since R5 */
```

87 " (7 ? ?

9
/ / % #
% #
= %

5 - =
% # ?
% - 8
? %

?
% # $ % #?
$ G
*2 -*28 %

9 $ $ %

'? 8 9 - 8 9
4 $ % %

PushPixels routine

?
% ? > $ % 2
% ?
%

" ;;<<<<;; " <;<;<;<;
? " ;;<;<;;;

2 $ % 2 $
) % + = %

```
void pGC->ops->PushPixels(pGC, pBitMap, pDrawable, dx, dy, xOrg, yOrg)
 GCPtr pGC;
 PixmapPtr pBitMap;
 DrawablePtr pDrawable;
 int dx, dy, xOrg, yOrg;
```

87 " (7 %

87 " ? G (
G (? G (?
? (> %

? 87
/ % 9 5 ?
$?((% * <
% *
; %

$ %

+ '? ? 4 $? % %

Shutdown Procedures

```
void AbortDDX(enum ExitCode error)
void ddxGiveUp(enum ExitCode error)
```

> = $?
% + ? ( ) '
+ //4 ! $ % - ?8 $ 0
! $ @ @ % * + //4
?8 $ 0 ? ? % -
%

Command Line Procedures

```
int ddxProcessArgument(argc, argv, i)
    int argc;
    char *argv[];
    int i;

void
ddxUseMsg()
```

2 $
% # ? +
? $G J $
$
%) % % @ @
$ % # =
$ = $
$ % 2 $
?0 5 0 5
$ %

Wrappers and Privates

? $ $ B *
$ $ % # ,A 87- >
? $ =

% $ $ $ $ 4%(4 $
<%K <%H %

devPrivates

$ $ $ $ $ $
%+ *devPrivates* $,
% $
! $ >
! % Xserver/include/privates.h
devPrivates%

> $

Bool dixRegisterPrivateKey(DevPrivateKey key, DevPrivateType type, unsigned

/$ $ 3 , $
> $ % # /
$ $ 3 , % # !
= % # J >
0 % - >
! % 1
dixSetPrivate $
$ % #
TRUE % -
= $ J $ %

> $!

Bool dixRegisterScreenPrivateKey(DevScreenPrivateKey key, ScreenPtr pScreen

type *size* dixRegisterPrivateKey
$ *key* ? !
$ *pScreen*% # = $ $
> $ $ %

$! "

void dixSetPrivate(PrivateRec **privates, const DevPrivateKey key, pointer

devPrivates %
$ /-4
% # /$ $ 3 ,
dixRegisterPrivateKey dixCreatePrivateKey%(
$ = !
= ! % + ?
,-.+#'Q4 '6-104 ! % #
$ %

- $ $ = ! dixSetPri-
vate $ $ %

= $ "

```
        pointer dixLookupPrivate(PrivateRec **privates, const DevPrivateKey key)
        pointer *dixLookupPrivateAddr(PrivateRec **privates, const DevPrivateKey ke

#                                    devPrivates              %
#            =    =  % -    J    J    $
=             $          $  =
   !            dixLookupPrivate            $                    dixLookup-
PrivateAddr                  %

*            $       !          $  $
            "+     $      /$  $  #                                    Xserv-
er/include/privates.h              $  ,  D         G
  J               NULL           !  G            !
        dixFreePrivates   dixFreeObjectWithPrivates    %
```

Wrappers

```
*                          % #
 >     =              ?         >   % #
                 $              =
    G                                % #
               87-     >
      ?        G
 $      $  % #                    "


       privateWrapperFunction (object, ...)
        ObjectPtr object;
       {
        pre-wrapped-function-stuff ...

        object->functionVector = dixLookupPrivate(&object->devPrivates, privateKey)
        (*object->functionVector) (object, ...);
        /*
         * this next line is occasionally required by the rules governing
         * wrapper functions.  Always using it will not cause problems.
         * Not using it when necessary can cause severe troubles.
         */
        dixSetPrivate(&object->devPrivates, privateKey, object->functionVector);
        object->functionVector = privateWrapperFunction;

        post-wrapped-function-stuff ...
       }

       privateInitialize (object)
        ObjectPtr object;
       {
        dixSetPrivate(&object->devPrivates, privateKey, object->functionVector);
        object->functionVector = privateWrapperFunction;
       }

#      $  *   )      $    =          =
                         G
  .        @      @
```

@ @% -
87- > % -
$! $ 87
$ %

* > G $ $ %
J
%

* 87 % 87 $
$ S S
7 87 % *

87 % 2
87 87
%

87 " 7 87
% -
% 1
$ % #
! %

Work Queue

> = ? % % !
*) "

```
Bool QueueWorkProc(function,client,closure)
 Bool  (*function)();
 ClientPtr client;
 pointer  closure;
```

* $ $? "

```
(*function) (client, closure)
```

1 => %

Summary of Routines

%
% # G =
% #
$ $ % 1
$ %

C ? G 4 $?

C $ 4 $ 4 $ %

C ? ? 4 $

C $ 4 $

C ? 4 $

C

ALLOCATE_LOCAL	os	
AbortDDX	hd	
AddCallback	dix	
AddEnabledDevice	os	
AddInputDevice	dix	
AddScreen	dix	
AdjustWaitForDelay	os	
Bell	hd	/ $
ChangeClip	mi	87
ChangeGC		87
ChangeWindowAttributes	ddx	
ClearToBackground	ddx	*
ClientAuthorized	os	
ClientSignal	dix	
ClientSleep	dix	
ClientWakeup	dix	
ClipNotify	ddx	
CloseScreen	hd	
ConstrainCursor	hd	
CopyArea	mi	87
CopyGCDest	ddx	87
CopyGCSource	none	87
CopyPlane	mi	87
CopyWindow	ddx	*
CreateGC	ddx	
CreateCallbackList	dix	
CreatePixmap	ddx	
CreateScreenResources	ddx	
CreateWellKnowSockets	os	
CreateWindow	ddx	
CursorLimits	hd	
DEALLOCATE_LOCAL	os	
DeleteCallback	dix	

DeleteCallbackList	dix	
DestroyClip	ddx	87
DestroyGC	ddx	87
DestroyPixmap	ddx	
DestroyWindow	ddx	
DisplayCursor	hd	
Error	os	
ErrorF	os	
FatalError	os	
FillPolygon	mi	87
FillSpans	ddx	87
FlushAllOutput	os	
FlushIfCriticalOutput-Pending	os	
FreeScratchPixmapHead-er	dix	
GetImage	mi	
GetMotionEvents	hd	/$
GetScratchPixmapHeader	dix	
GetSpans	ddx	
GetStaticColormap	ddx	

ImageGlyphBlt	mi	87
ImageText16	mi	87
ImageText8	mi	87
InitInput	hd	
InitKeyboardDe-viceStruct	dix	
InitOutput	hd	
InitPointerDe-viceStruct	dix	
InsertFakeRequest	os	
InstallColormap	ddx	
Intersect	mi	
Inverse	mi	
LegalModifier	hd	
LineHelper	mi	87

ListInstalledColormaps	ddx	
LookupKeyboardDevice	dix	
LookupPointerDevice	dix	
ModifyPixmapHeader	mi	
NextAvailableClient	dix	
OsInit	os	
PaintWindowBackground	mi	*
PaintWindowBorder	mi	*
PointerNonInterestBox	hd	
PointInRegion	mi	
PolyArc	mi	87
PolyFillArc	mi	87
PolyFillRect	mi	87
PolyGlyphBlt	mi	87
Polylines	mi	87
PolyPoint	mi	87
PolyRectangle	mi	87
PolySegment	mi	87
PolyText16	mi	87
PolyText8	mi	87
PositionWindow	ddx	
ProcessInputEvents	hd	
PushPixels	mi	87
PutImage	mi	87
QueryBestSize	hd	
ReadRequestFromClient	os	
RealizeCursor	hd	
RealizeFont	ddx	
RealizeWindow	ddx	
RecolorCursor	hd	
RectIn	mi	
RegionCopy	mi	
RegionCreate	mi	
RegionDestroy	mi	
RegionEmpty	mi	
RegionExtents	mi	
RegionNotEmpty	mi	
RegionReset	mi	

ResolveColor	ddx	

RemoveEnabledDevice	os	
ResetCurrentRequest	os	
SaveScreen	ddx	
SetCriticalOutputPend-ing	os	
SetCursorPosition	hd	
SetInputCheck	dix	
SetSpans	ddx	87
StoreColors	ddx	
Subtract	mi	
TimerCancel	os	
TimerCheck	os	
TimerForce	os	
TimerFree	os	
TimerInit	os	
TimerSet	os	
TimeSinceLastIn-putEvent	hd	
TranslateRegion	mi	
UninstallColormap	ddx	
Union	mi	
UnrealizeCursor	hd	
UnrealizeFont	ddx	
UnrealizeWindow	ddx	
ValidateGC	ddx	87
ValidateTree	mi	
WaitForSomething	os	
WindowExposures	mi	*
WriteToClient	os	
Xalloc	os	
Xfree	os	
Xrealloc	os	