Extended Visual Information Extension

X Project Team Standard

Peter Daifuku, Silicon Graphics, Inc.

Extended Visual Information Extension: X Project Team Standard

by Peter Daifuku

Version 1.0
Copyright © 1986-97 The Open Group

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the Software), to use the Software without restriction, including, without limitation, the rights to copy, modify, merge, publish, distribute and sublicense the Software, to make, have made, license and distribute derivative works thereof, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and the following permission notice shall be included in all copies of the Software:

THE SOFTWARE IS PROVIDED AS IS, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON- INFRINGEMENT. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER USEABILITIY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF, OR IN CONNNECTION WITH THE SOFTWARE OR THE USE OF OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of The Open Group shall not be used in advertising or otherwise to promote the use or other dealings in this Software without prior written authorization from The Open Group.

X Window System is a trademark of The Open Group.

Table of Contents

Chapter 1. Introduction

EVI (Extended Visual Information extension) allows a client to determine information about core X visuals beyond what the core protocol provides.

Chapter 2. Goals

As the X Window System has evolved, it has become clear that the information returned by the core X protocol regarding Visuals is often insufficient for a client to determine which is the most appropriate visual for its needs. This extension allows clients to query the X server for additional visual information, specifically as regards colormaps and framebuffer levels.

This extension is meant to address the needs of pure X clients only. It is specifically and purposefully not designed to address the needs of X extensions. Extensions that have an impact on visual information should provide their own mechanisms for delivering that information. For example, the Double Buffering Extension (DBE) provides its own mechanism for determining which visuals support double-buffering.

Chapter 3. Requests

`GetVersion`

client_major_version: CARD8
client_minor_version: CARD8
=>
server_major_version: CARD8
server_minor_version: CARD8

If supplied, the client_major_version and client_minor_version indicate what version of the protocol the client wants the server to implement. The server version numbers returned indicate the protocol this extension actually supports. This might not equal the version sent by the client. An implementation can (but need not) support more than one version simultaneously. The server_major_version and the server_minor_version are a mechanism to support future revisions of the EVI protocol that may be necessary. In general, the major version would increment for incompatible changes, and the minor version would increment for small upward-compatible changes. Servers that support the protocol defined in this document will return a server_major_version of one (1), and a server_minor_version of zero (0).

`GetVisualInfo`

visual_list: LISTofVISUALID
=>
per_visual_info: LISTofVISUALINFO

where:

VISUALINFO: [core_visual_id: VISUALID
screen: CARD8
level: INT8
transparency_type: CARD8
unused: CARD8
transparency_value: CARD32
min_hw_colormaps: CARD8
max_hw_colormaps: CARD8
num_colormap_conflicts: CARD16
colormap_conflicts: LISTofVISUALID]

- level is 0 for normal planes, > 0 for overlays, < 0 for underlays.
- transparency_type is 0 for none, 1 for transparent pixel, 2 for transparent mask.
- transparency_value: value to get transparent pixel if transparency supported.
- min_hw_colormaps: minimum number of hardware colormaps backing up the visual.

- max_hw_colormaps: maximum number of hardware colormaps backing up the visual.

  (architectures with static colormap allocation/reallocation would have min = max)

- num_colormap_conflicts: number of elements in colormap_conflicts.

- colormap_conflicts: list of visuals that may conflict with this one. For example, if a 12-bit colormap is overloaded to support 8-bit visuals, the 8-bit visuals would conflict with the 12-bit visuals.

Chapter 4. Events and Errors

No new events or errors are defined by this extension.

Chapter 5. Changes to existing protocol.

None.

Chapter 6. Encoding

The name of this extension is "Extended-Visual-Information".

The conventions used here are the same as those for the core X11 Protocol Encoding.

```
GetVersion
     1     CARD8               opcode
     1     0                   EVI opcode
     2     2                   request length
     2     CARD16              client_major_version
     2     CARD16              client_minor_version
=>
     1     1                   reply
     1                         unused
     2     CARD16              sequence number
     4     0                   length
     2     CARD16              server_major_version
     2     CARD16              server_minor_version
     20                        unused
```

```
GetVisualInfo
     1     CARD8               opcode
     1     1                   EVI opcode
     2     2+n                 request length
     4     CARD32              n_visual
     4n    CARD32              visual_ids
=>
     1     1                   reply
     1                         unused
     2     CARD16              sequence number
     4     n                   length
     4     CARD32              n_info
     4     CARD32              n_conflicts
     16                        unused
     16n   LISTofVISUALINFO    items
```

```
VISUALINFO
     4     VisualID            core_visual_id
     1     INT8                screen
     1     INT8                level
     1     CARD8               transparency_type
     1     CARD8               unused
     4     CARD32              transparency_value
     1     CARD8               min_hw_colormaps
     1     CARD8               max_hw_colormaps
     2     CARD16              num_colormap_conflicts
```

Chapter 7. C Language Binding

The C functions provide direct access to the protocol and add no additional semantics. For complete details on the effects of these functions, refer to the appropriate protocol request, which can be derived by deleting Xevi at the start of the function. All functions that have return type Status will return nonzero for success and zero for failure.

The include file for this extension is: `< X11/extensions/XEVI.h>`.

```
Bool XeviQueryVersion( *display, *major_version_return, *minor_version_return);
```

display	Specifies the connection to the X server.
major_version_return	Returns the major version supported by the server.
minor_version_return	Returns the minor version supported by the server.

XeviQueryVersion sets major_version_return and minor_version_return to the major and minor EVI protocol version supported by the server. If the EVI library is compatible with the version returned by the server, it returns nonzero. If dpy does not support the EVI extension, or if there was an error during communication with the server, or if the server and library protocol versions are incompatible, it returns zero. No other Xevi functions may be called before this function. If a client violates this rule, the effects of all subsequent Xevi calls that it makes are undefined.

To get the extended information for any subset of visuals use XeviGetVisualInfo.

```
int XeviGetVisualInfo( *display, *visual, n_visual, **evi_return, *n_info_return);
```

display	Specifies the connection to the X server.
visual	If NULL, then information for all visuals of all screens is returned. Otherwise, a pointer to a list of visuals for which extended visual information is desired.
n_visual	If 0, then information for all visuals of all screens is returned. Otherwise, the number of elements in the array *visual*.
evi_return	Returns a pointer to a list of *ExtendedVisualInfo*. When done, the client should free the list using *XFree*.
n_info_return	Returns the number of elements in the list of *ExtendedVisualInfo*.

XeviGetVisualInfo returns a list of ExtendedVisualInfo structures that describe visual information beyond that supported by the core protocol. This includes layer information relevant for systems supporting overlays and/or underlay planes, and information that allows applications better to determine the level of hardware support for multiple colormaps. XeviGetVisualInfo returns Success if successful, or an X error otherwise.