NAME

XSolarisOvlSelectPartner – returns the overlay/underlay visual that best meets the criteria

SYNOPSIS

XSolarisOvlSelectStatus
XSolarisOvlSelectPartner (Display **display,* **int** *screen,* **VisualID** *vid,* **XSolarisOvlSelectType** *seltype,* **int** *numCriteria,* **XSolarisOvlVisualCriteria** **pCriteria,* **XVisualInfo** **visinfoReturn,* **unsigned long** **unmetCriteriaReturn)*

Arguments

display Specifies the connection to the X server.

screen An integer specifying the screen for the visual *vid.*

vid The XID of the visual to find a partner for.

seltype The type of selection that is to be done.

numCriteria
The number of **XSolarisOvlVisualCriteria** structures in the *pCriteria array.*

pCriteria
An array of criteria structures in priority order from high to low specifying the criteria to be used in selecting the visual.

visinfoReturn
A pointer to a caller provided **XVisualInfo** structure. On successful return, this structure contains a description of the chosen visual.

unmetCriteriaReturn
A pointer to a bitmask that describes the criteria that were not satisfied. This return argument is only meaningful when the routine returns a value of **XSolarisOvlQualifiedSuccess,** or **XSolarisOvlCriteriaFailure.**

Argument Types

See the **XSolarisOvlSelectPartner** Description section for a full description of how these types should be used.

XSolarisOvlSelectType

An enumeration defining two types of selections that can be done in **XSolarisOvlSelectPartner.**

```
typedef enum {
        XSolarisOvlSelectBestOverlay,
        XSolarisOvlSelectBestUnderlay,
} XSolarisOvlSelectType;
```

XSolarisOvlVisualCriteria

A structure defining various criteria to be used during visual selection, along with indications of the stringency of the criteria.

```
typedef struct {
        unsigned long   hardCriteriaMask;
        unsigned long   softCriteriaMask
        int             c_class;
```

```
        unsigned int    depth;
        unsigned int    minColors;
        unsigned int    minRed;
        unsigned int    minGreen;
        unsigned int    minBlue;
        unsigned int    minBitsPerRGB;
        unsigned int    minBuffers;
} XSolarisOvlVisualCriteria;
```

hardCriteriaMask and *softCriteriaMask* are bitmasks whose values can be the logical OR of any of the following bitmasks:

```
#define XSolarisOvlVisualClass          (1L<<0)
#define XSolarisOvlDepth                (1L<<1)
#define XSolarisOvlMinColors            (1L<<2)
#define XSolarisOvlMinRed               (1L<<3)
#define XSolarisOvlMinGreen             (1L<<4)
#define XSolarisOvlMinBlue              (1L<<5)
#define XSolarisOvlMinBitsPerRGB        (1L<<6)
#define XSolarisOvlMinBuffers           (1L<<7)
#define XSolarisOvlUnsharedPixels       (1L<<8)
#define XSolarisOvlUnsharedColors       (1L<<9)
#define XSolarisOvlPreferredPartner     (1L<<10)
```

These are described in the **XSolarisOvlSelectPartner** Description documentation that follows.

Return Types

XSolarisOvlSelectStatus

A value that indicates whether the routine succeeded in finding a visual and, if it failed, the reason for the failure. The return value can be one of:

```
typedef enum {
        XSolarisOvlSuccess,
        XSolarisOvlQualifiedSuccess,
        XSolarisOvlCriteriaFailure,
        XSolarisOvlFailure,
} XSolarisOvlSelectStatus;
```

XSolarisOvlSuccess is returned if the search is completely successful in finding a visual that meets all hard and soft criteria of one of the **XSolarisOvlVisualCriteria** structure.

XSolarisOvlQualifiedSuccess is returned if the chosen visual satisfies all hard criteria of one of the **XSolarisOvlVisualCriteria** structure, but doesn't meet all soft criteria. In this case, *unmetCriteriaReturn* contains the logical OR of the soft criteria that were not met.

XSolarisOvlCriteriaFailure indicates that no visual could be found that meets all the hard criteria of any of the **XSolarisOvlVisualCriteria** structures. In this case, *unmetCriteriaReturn* contains the logical OR of the hard criteria that were not met for the **XSolarisOvlVisualCriteria** structure with the fewest hard criteria not met.

XSolarisOvlFailure is returned if some other error is encountered besides criteria match failure.

DESCRIPTION

Portable applications using overlays may wish to search for an appropriate overlay visual to use for a given underlay visual, or vice-versa. Each X screen supporting the overlay extension defines a set of overlay visuals whose windows are best for use as children of underlay windows. For each underlay visual, there is a set of *optimal* overlay visuals. Together, all combinations of underlay visuals and their optimal overlay visuals form the set of optimal overlay/underlay pairs for that screen. The overlay and underlay visuals of an optimal pair are said to be *partners* of each other.

XSolarisOvlSelectPartner allows the client to select, given an underlay visual, an *optimal* overlay that meets certain criteria. Inversely, it also allows the client to select an optimal underlay visual given an overlay visual.

The client is assured that, short of X errors not related to overlays, it can successfully create a window with the returned visual.

This routine searches through the optimal partners of the given visual, applying the criteria specified in *pCriteria.* It returns a success or failure status depending on whether it finds a visual that meets the criteria.

A criterion can be one of two types:

1. Hard Criterion

   A criterion that must be satisfied. Only visuals that meet hard criteria are candidates for successful matches.

2. Soft Criterion

   A desirable criterion, but one which is not required. The visual that matches all hard criteria and the most soft criteria is chosen. Its attributes are returned in *visinfoReturn.* If two or more visuals are found that meet all of the hard criteria and the same number of soft criteria, one of them will be chosen and returned. It is implementation dependent which one is chosen.

XSolarisOvlSelectPartner supports a degradation sequence of criteria sets. This means that multiple criteria sets can be specified in a single call. First, an attempt is made to find a visual matching the first criteria set. If a visual is found which meets all of the hard criteria of the first set, this visual is chosen. If no visual met all hard criteria of the first set, a search is performed using the second criteria set. This process continues until either a visual is found that meets the hard criteria of some criteria set, or all sets have been used to search. This degradation sequence allows clients to specify the criteria for the most preferred visual as the first criteria set. Visuals that are acceptable but which are less desirable can be specified in criteria sets following the first. This allows the search to proceed through a progressive relaxation in the client’s requirements for the visual with a single subroutine call.

Any of the possible criteria can be specified either as a hard or soft criteria for a particular criteria set. For a given set, *hardCriteriaMask* is the logical OR of the criteria bitmasks that are to be applied as hard criteria during the search. Likewise, *softCriteriaMask* is the logical OR of the soft criteria bitmasks.

Some criteria have values associated with them. These values are provided by other data members in the **XSolarisOvlVisualCriteria** structure. In the criteria descriptions which follow, these data members are mentioned where applicable.

XSolarisOvlVisualClass specifies that the client desires the selected visual to have a specific visual class. The required class is specified in *c_class.*

The following criteria interact within one another: **XSolarisOvlDepth, XSolarisOvlMinColors, XSolarisOvlMinRed, XSolarisOvlMinGreen,** and **XSolarisOvlMinBlue.** Typically, only some subset of these should be specified. **XSolarisOvlDepth** specifies that the depth of the selected visual is to be equal to depth. **XSolarisOvlMinColors** specifies that the selected visual is to have at least minColors number of total displayable colors. **XSolarisOvlMinRed, XSolarisOvlMinGreen,** and **XSolarisOvlMinBlue** can be used to indicate more specific color requirements for *DirectColor* or *TrueColor* visuals. Their corresponding values are specified in *minRed, minGreen,* and *minBlue,* respectively. These indicate that the selected visual must have at least the specified number of reds, greens, and/or blues.

XSolarisOvlMinBitsPerRGB specifies that the selected visual is to have at least *minBitsPerRGB* of color channel output from colormaps created on that visual.

XSolarisOvlMinBuffers specifies that the client desires the selected visual to be able to be assigned at least *minBuffers* number of accelerated MBX image buffers.

XSolarisOvlUnsharedPixels selects partner visuals whose window pixels don't lie in the same drawing plane groups as the window pixels of the argument visual *vid.* If a visual uses the same drawing plane group as the argument visual it is not matched by this criterion.

XSolarisOvlUnsharedColors selects partner visuals whose window pixel colors can be displayed simultaneously when the overlay/underlay window pair has the colormap focus. If a visual shares the same color LUT pool and that pool has only one color LUT in it as the argument visual it is not matched by this criterion.

If either *hardCriteriaMask* of a criteria set is to 0, any visual will match that criteria set with a hard match. Likewise, setting the *softCriteriaMask* of a criteria set to 0, is sufficient to guarantee at least a soft match for that criteria set.