screen-root-depth *screen* Function

Returns:
root-window-depth — Type **image-depth**.

Returns the depth of the root window for the specified *screen*. Other depths can also be supported on this *screen*.

screen — A **screen**.

screen-root-visual *screen* Function

Returns:
root-window-visual — Type **card29**.

Returns the default visual type for the root window for the specified *screen*.

screen — A **screen**.

screen-save-unders-p *screen* Function

Returns:
save-unders-p — Type **boolean**.

If true, the server can support the save-under mode in **create-window** and in changing window attributes.

screen — A screen.

screen-white-pixel *screen* Function

Returns:
white-pixel — Type **pixel**.

Returns the white pixel value for the specified *screen*.

screen — A screen.

screen-width *screen* Function

Returns:
width — Type **card16**.

Returns the width of the specified *screen* in pixel units.

screen — A screen.

screen-width-in-millimeters *screen* Function

Returns:
width-in-millimeters — Type **card16**.

Returns the width of the specified *screen* in millimeters. The returned width can be used with the height in millimeters to determine the physical size and the aspect ratio of the screen.

screen — A screen.

screen-height *screen* Function

Returns:
height — Type **card16**.

Returns the *height* of the specified *screen* in pixel units.

screen — A **screen**.

screen-height-in-millimeters *screen* Function

Returns:
height-in-millimeters — Type **card16**.

Returns the height of the specified *screen* in millimeters. The returned height can be used with the width in millimeters to determine the physical size and the aspect ratio of the screen.

screen — A **screen**.

screen-max-installed-maps *screen* Function

Returns:
max-installed-colormaps — Type **card16**.

Returns the maximum number of colormaps that can be installed simultaneously with **install-colormap**.

screen — A **screen**.

screen-min-installed-maps *screen* Function

Returns:
min-installed-colormaps — Type **card16**.

Returns the minimum number of colormaps that can be guaranteed to be installed simultaneously.

screen — A **screen**.

screen-p *screen* Function

Returns:
screen-p — Type **boolean**.

Returns non-**nil** if the *screen* argument is a **screen** structure; otherwise, returns **nil**.

screen-plist *screen* Function

Returns:
plist — Type **list**.

Returns and (with **setf**) sets the property list for the specified *screen*. This function provides a hook where extensions can add data.

screen — A **screen**.

screen-root *screen* Function

Returns:
root-window — Type **window** or **null**.

Returns the *root-window* for the specified *screen*. This function is useful with functions that take a parent window as an argument. The class of the root window is always **:input-output**.

screen — A **screen**.

screen-backing-stores *screen* Function

Returns:
backing-stores-type — One of **:always**, **:never**, or **:when-mapped**.

Returns a value indicating when the *screen* supports backing stores, although it may be storage limited in the number of windows it can support at once. The value returned can be one of **:always**, **:never**, or **:when-mapped**.

screen — A **screen**.

screen-black-pixel *screen* Function

Returns:
black-pixel — Type **pixel**.

Returns the black pixel value for the specified *screen*.

screen — A **screen**.

screen-default-colormap *screen* Function

Returns:
default-colormap — Type **colormap**.

Returns the *default-colormap* for the specified *screen*. The *default-colormap* is initially associated with the root window. Clients with minimal color requirements creating windows of the same depth as the root may want to allocate from this map by default. Most routine allocations of color should be made out of this colormap.

screen — A **screen**.

screen-depths *screen* Function

Returns:
depths — Type **alist**.

Returns an association list that specifies what drawable depths are supported on the specified *screen*. Elements of the returned association list have the form (depth *visual**), where each *visual* is a **visual-info** structure. Pixmaps are supported for each depth listed, and windows of that depth are supported if at least one visual type is listed for the depth. A pixmap depth of one is always supported and listed, but windows of depth one might not be supported. A depth of zero is never listed, but zero-depth **:input-only** windows are always supported.

screen — A **screen**.

screen-event-mask-at-open *screen* Function

Returns:
event-mask-at-open — Type **mask32**.

Returns the initial root event mask for the specified *screen*.

screen — A **screen**.

3 SCREENS

Screens and Visuals

3.1 An X display supports graphical output to one or more *screens*. Each screen has its own root window and window hierarchy. Each window belongs to exactly one screen and cannot simultaneously appear on another screen.

The kinds of graphics hardware used by X screens can vary greatly in their support for color and in their methods for accessing raster memory. X uses the concept of a *visual type* (usually referred to simply as a *visual*) which uniquely identifies the hardware capabilities of a display screen. Fundamentally, a visual is represented by a **card29** integer ID, which uniquely identifies the visual type relative to a single display. CLX also represents a visual with a **visual-info** structure that contains other attributes associated with a visual (see paragraph 1.6, Data Types). A screen can support more than one depth (that is, pixel size), and for each supported depth, a screen may support more than one visual. However, it is more typical for a screen to have only a single depth and a single visual type.

A visual represents various aspects of the screen hardware, as follows:

- A screen can be color or gray-scale.
- A screen can have a colormap that is either writable or read-only.
- A screen can have a single colormap or separate colormaps for each of the red, green, and blue components. With separate colormaps, a pixel value is decomposed into three parts to determine indexes into each of the red, green, and blue colormaps.

CLX supports the following classes of visual types: **:direct-color**, **:gray-scale**, **:pseudo-color**, **:static-color**, **:static-gray**, and **:true-color**. The following tables show how the characteristics of a screen determine the class of its visual type.

For screens with a single colormap:

	Color	Gray-Scale
Read-only	**:static-color**	**:static-gray**
Writable	**:pseudo-color**	**:gray-scale**

For screens with red, green, and blue colormaps:

Read-only	**:true-color**	
Writable	**:direct-color**	**:gray-scale**

The visual class also indicates how screen colormaps are handled. See paragraph 9.1, Colormaps and Colors.

Screen Attributes

3.2 In CLX, each display screen is represented by a **screen** structure. The **display-roots** function returns the list of **screen** structures for the display. The following paragraphs discuss the attributes of CLX **screen** structures.