

# **Guile-GNOME: Clutter-GLX**

---

version 1.10.0, updated 8 May 2012

**Matthew Allum and OpenedHand LTD  
Intel Corporation**

---

This manual is for (**gnome clutter-glx**) (version 1.10.0, updated 8 May 2012)

Copyright 2006,2007,2008,2012 OpenedHand LTD

Copyright 2009,2010,2011,2012 Intel Corporation

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. You may obtain a copy of the GNU Free Documentation License from the Free Software Foundation by visiting their Web site or by writing to:

The Free Software Foundation, Inc., 59 Temple Place - Suite 330,  
Boston, MA 02111-1307, USA

## Short Contents

1	Overview .....	1
2	X11 Specific Support .....	2
3	ClutterX11TexturePixmap .....	5
4	GLX Specific Support .....	6
5	Undocumented .....	7
	Type Index .....	8
	Function Index .....	9

# 1 Overview

`(gnome clutter)` wraps the Clutter graphical canvas toolkit for Guile. It is a part of Guile-GNOME.

See the documentation for `(gnome gobject)` for more information on Guile-GNOME.

## 2 X11 Specific Support

X11 specific API

### 2.1 Overview

The X11 backend for Clutter provides some specific API, allowing integration with the Xlibs API for embedding and manipulating the stage window, or for trapping X errors.

The ClutterX11 API is available since Clutter 0.6

### 2.2 Usage

`clutter-x11-has-event-retrieval`  $\Rightarrow$  (*ret* `bool`) [Function]

Queries the X11 backend to check if event collection has been disabled.

*ret* TRUE if event retrieval has been disabled. FALSE otherwise.

Since 0.8

`clutter-x11-get-stage-from-window` (*win* `unsigned-int32`) [Function]

$\Rightarrow$  (*ret* `<clutter-stage>`)

Gets the stage for a particular X window.

*win* an X Window ID

*ret* A `<clutter-stage>`, or % NULL if a stage does not exist for the window.

Since 0.8

`clutter-x11-get-default-screen`  $\Rightarrow$  (*ret* `int`) [Function]

Gets the number of the default X Screen object.

*ret* the number of the default screen

Since 0.6

`clutter-x11-get-root-window`  $\Rightarrow$  (*ret* `unsigned-int32`) [Function]

Retrieves the root window.

*ret* the id of the root window

Since 0.6

`clutter-x11-get-stage-window` (*stage* `<clutter-stage>`) [Function]

$\Rightarrow$  (*ret* `unsigned-int32`)

Gets the stages X Window.

*stage* a `<clutter-stage>`

*ret* An XID for the stage window.

Since 0.4

**clutter-x11-set-stage-foreign** (*stage* <clutter-stage>) [Function]  
 (*xwindow* unsigned-int32) ⇒ (*ret* bool)

Target the <clutter-stage> to use an existing external X Window

*stage* a <clutter-stage>

*xwindow* an existing X Window id

*ret* '#t' if foreign window is valid

Since 0.4

**clutter-x11-trap-x-errors** [Function]

Traps every X error until **clutter-x11-untrap-x-errors** is called.

Since 0.6

**clutter-x11-untrap-x-errors** ⇒ (*ret* int) [Function]

Removes the X error trap and returns the current status.

*ret* the trapped error code, or 0 for success

Since 0.4

**clutter-x11-has-composite-extension** ⇒ (*ret* bool) [Function]

Retrieves whether Clutter is running on an X11 server with the XComposite extension

*ret* '#t' if the XComposite extension is available

**clutter-x11-set-use-argb-visual** (*use\_argb* bool) [Function]

Sets whether the Clutter X11 backend should request ARGB visuals by default or not.

By default, Clutter requests RGB visuals.

If no ARGB visuals are found, the X11 backend will fall back to requesting a RGB visual instead.

ARGB visuals are required for the <"use-alpha"> property to work.

This function can only be called once, and before **clutter-init** is called.

*use-argb* '#t' if ARGB visuals should be requested by default

Since 1.2

**clutter-x11-get-use-argb-visual** ⇒ (*ret* bool) [Function]

Retrieves whether the Clutter X11 backend is using ARGB visuals by default

*ret* '#t' if ARGB visuals are queried by default

Since 1.2

`clutter-x11-has-xinput`  $\Rightarrow$  (*ret* bool) [Function]

Gets whether Clutter has XInput support.

*ret*        ‘#t’ if Clutter was compiled with XInput support and XInput support is available at run time.

Since 0.8

`clutter-x11-enable-xinput` [Function]

Enables the use of the XInput extension if present on connected XServer and support built into Clutter. XInput allows for multiple pointing devices to be used.

This function must be called before `clutter-init`.

Since XInput might not be supported by the X server, you might want to use `clutter-x11-has-xinput` to see if support was enabled.

Since 0.8

`clutter-x11-event-get-key-group` (*event* <clutter-event>) [Function]

$\Rightarrow$  (*ret* int)

Retrieves the group for the modifiers set in *event*

*event*       a    <clutter-event>    of    type    ‘CLUTTER\_KEY\_PRESS’    or  
              ‘CLUTTER\_KEY\_RELEASE’

*ret*        the group id

Since 1.4

## 3 ClutterX11TexturePixmap

A texture which displays the content of an X Pixmap.

### 3.1 Overview

`<clutter-x11-texture-pixmap>` is a class for displaying the content of an X Pixmap as a ClutterActor. Used together with the X Composite extension, it allows to display the content of X Windows inside Clutter.

The class uses the `GLX_EXT_texture_from_pixmap` OpenGL extension ([http://people.freedesktop.org/~davidr/GLX\\_EXT\\_texture\\_from\\_pixmap.txt](http://people.freedesktop.org/~davidr/GLX_EXT_texture_from_pixmap.txt)) if available

### 3.2 Usage

`clutter-x11-texture-pixmap-new`  $\Rightarrow$  (*ret* `<clutter-actor>`) [Function]

Creates a new `<clutter-x11-texture-pixmap>` which can be used to display the contents of an X11 Pixmap inside a Clutter scene graph

*ret*            A new `<clutter-x11-texture-pixmap>`

Since 0.8



## 4 GLX Specific Support

GLX specific API

### 4.1 Overview

The GLX backend for Clutter provides some specific API for GLX related calls.

The ClutterGLX API is available since Clutter 0.4

### 4.2 Usage

`clutter-glx-texture-pixmap-new`  $\Rightarrow$  (*ret* <clutter-actor>) [Function]  
'`clutter_glx_texture_pixmap_new`' has been deprecated since version 1.6 and should not be used in newly-written code. Use `clutter-x11-texture-pixmap-new` instead  
Creates a new, empty <clutter-glx-texture-pixmap>  
*ret*            A new <clutter-glx-texture-pixmap>  
Since 0.8

## 5 Undocumented

The following symbols, if any, have not been properly documented.

### 5.1 (gnome gw clutter-glx)

<code>clutter-x11-texture-pixmap-new-with-pixmap</code>	[Variable]
<code>clutter-x11-texture-pixmap-new-with-window</code>	[Variable]
<code>clutter-x11-texture-pixmap-set-automatic</code>	[Variable]
<code>clutter-x11-texture-pixmap-set-pixmap</code>	[Variable]
<code>clutter-x11-texture-pixmap-set-window</code>	[Variable]
<code>clutter-x11-texture-pixmap-sync-window</code>	[Variable]
<code>clutter-x11-texture-pixmap-update-area</code>	[Variable]

## Type Index

(Index is nonexistent)

## Function Index

clutter-glx-texture-pixmap-new.....	6	clutter-x11-has-composite-extension.....	3
clutter-x11-enable-xinput .....	4	clutter-x11-has-event-retrieval.....	2
clutter-x11-event-get-key-group.....	4	clutter-x11-has-xinput .....	4
clutter-x11-get-default-screen.....	2	clutter-x11-set-stage-foreign.....	3
clutter-x11-get-root-window.....	2	clutter-x11-set-use-argb-visual.....	3
clutter-x11-get-stage-from-window.....	2	clutter-x11-texture-pixmap-new.....	5
clutter-x11-get-stage-window.....	2	clutter-x11-trap-x-errors .....	3
clutter-x11-get-use-argb-visual.....	3	clutter-x11-untrap-x-errors .....	3