NAME
XLookupKeysym, XRefreshKeyboardMapping, XLookupString, XRebindKeysym - handle keyboard
input events in Latin-1

SYNTAX
KeySym XLookupKeysym(XKeyEvent *_key_event_, int _index_);

int XRefreshKeyboardMapping(XMappingEvent *_event_map_);

int XLookupString(XKeyEvent *_event_struct_, char *_buffer_return_, int _bytes_buffer_, KeySym
     *_keysym_return_, XComposeStatus *_status_in_out_);

int XRebindKeysym(Display *_display_, KeySym _keysym_, KeySym _list_[], int _mod_count_, _Xconst
     unsigned char *_string_, int _num_bytes_);

ARGUMENTS
_buffer_return_   Returns the translated characters.

_bytes_buffer_   Specifies the length of the buffer. No more than bytes_buffer of translation are
                  returned.

_num_bytes_   Specifies the number of bytes in the string argument.

display   Specifies the connection to the X server.

_event_map_   Specifies the mapping event that is to be used.

_event_struct_   Specifies the key event structure to be used. You can pass **XKeyPressedEvent** or
                  XKeyReleasedEvent.

index   Specifies the index into the KeySyms list for the event's KeyCode.

_key_event_   Specifies the **KeyPress** or **KeyRelease** event.

keysym   Specifies the KeySym that is to be returned.

_keysym_return_ Returns the KeySym computed from the event if this argument is not NULL.

list   Specifies the KeySyms to be used as modifiers.

mod_count   Specifies the number of modifiers in the modifier list.

status_in_out   Specifies or returns the **XComposeStatus** structure or NULL.

string   Specifies the string that is copied and will be returned by **XLookupString**.

DESCRIPTION

The **XLookupKeysym** function uses a given keyboard event and the index you specified to return the KeySym from the list that corresponds to the KeyCode member in the **XKeyPressedEvent** or **XKeyReleasedEvent** structure. If no KeySym is defined for the KeyCode of the event, **XLookupKeysym** returns **NoSymbol**.

The **XRefreshKeyboardMapping** function refreshes the stored modifier and keymap information. You usually call this function when a **MappingNotify** event with a request member of **MappingKeyboard** or **MappingModifier** occurs. The result is to update Xlib's knowledge of the keyboard.

The **XLookupString** function translates a key event to a KeySym and a string. The KeySym is obtained by using the standard interpretation of the **Shift**, **Lock**, group, and numlock modifiers as defined in the X Protocol specification. If the KeySym has been rebound (see **XRebindKeysym**), the bound string will be stored in the buffer. Otherwise, the KeySym is mapped, if possible, to an ISO Latin-1 character or (if the Control modifier is on) to an ASCII control character, and that character is stored in the buffer. **XLookupString** returns the number of characters that are stored in the buffer.

If present (non-NULL), the **XComposeStatus** structure records the state, which is private to Xlib, that needs preservation across calls to **XLookupString** to implement compose processing. The creation of **XComposeStatus** structures is implementation-dependent; a portable program must pass NULL for this argument.

The **XRebindKeysym** function can be used to rebind the meaning of a KeySym for the client. It does not redefine any key in the X server but merely provides an easy way for long strings to be attached to keys. **XLookupString** returns this string when the appropriate set of modifier keys are pressed and when the KeySym would have been used for the translation. No text conversions are performed; the client is responsible for supplying appropriately encoded strings. Note that you can rebind a KeySym that may not exist.

SEE ALSO

XButtonEvent(3), XMapEvent(3), XStringToKeysym(3), XkbLookupKeySym(3), XkbRefreshKeyboardMapping(3), XmbLookupString(3), XwcLookupString(3), Xutf8LookupString(3), Compose(5)
Xlib - C Language X Interface