XInternAtom, XInternAtoms, XGetAtomName, XGetAtomNames – create or return atom names

Atom XInternAtom(*display***,** *atom_name***,** *only_if_exists***)**
Display ***display*;**
char ***atom_name*;**
Bool *only_if_exists***;**

Status XInternAtoms(*display*, *names*, *count*, *only_if_exists*, *atoms_return*)
Display **display*;
char ***names*;
int *count*;
Bool *only_if_exists*;
Atom **atoms_return*;

char *XGetAtomName(*display*, *atom*)
Display **display*;
Atom *atom*;

Status XGetAtomNames(*display*, *atoms*, *count*, *names_return*)
Display **display*;
Atom **atoms*;
int *count*;
char ***names_return*;

atom	Specifies the atom for the property name you want returned.
atoms	Specifies the array of atoms.
atom_name	Specifies the name associated with the atom you want returned.
atoms_return	Returns the atoms.
count	Specifies the number of atom names in the array.
count	Specifies the number of atoms in the array.
display	Specifies the connection to the X server.
names	Specifies the array of atom names.
names_return	Returns the atom names.
only_if_exists	Specifies a Boolean value that indicates whether the atom must be created.

The XInternAtom function returns the atom identifier associated with the specified atom_name string. If only_if_exists is **False**, the atom is created if it does not exist. Therefore, **XInternAtom** can return **None**. If the atom name is not in the Host Portable Character Encoding, the result is implementation-dependent. Uppercase and lowercase matter; the strings ‘‘thing’’, ‘‘Thing’’, and ‘‘thinG’’ all designate different atoms. The atom will remain defined even after the client’s connection closes. It will become undefined only when the last connection to the X server closes.

XInternAtom can generate **BadAlloc** and **BadValue** errors.

The **XInternAtoms** function returns the atom identifiers associated with the specified names. The atoms are stored in the atoms_return array supplied by the caller. Calling this function is equivalent to calling **XInternAtom** for each of the names in turn with the specified value of only_if_exists, but this function minimizes the number of round-trip protocol exchanges between the client and the X server.

This function returns a nonzero status if atoms are returned for all of the names; otherwise, it returns zero.

XInternAtoms can generate **BadAlloc** and **BadValue** errors.

The **XGetAtomName** function returns the name associated with the specified atom. If the data returned by the server is in the Latin Portable Character Encoding, then the returned string is in the Host Portable Character Encoding. Otherwise, the result is implementation-dependent. To free the resulting string, call **XFree**.

XGetAtomName can generate a **BadAtom** error.

The **XGetAtomNames** function returns the names associated with the specified atoms. The names are stored in the names_return array supplied by the caller. Calling this function is equivalent to calling **XGetAtomName** for each of the atoms in turn, but this function minimizes the number of round-trip protocol exchanges between the client and the X server.

This function returns a nonzero status if names are returned for all of the atoms; otherwise, it returns zero.

XGetAtomNames can generate a **BadAtom** error.

BadAlloc The server failed to allocate the requested resource or server memory. **BadAtom** A value for an Atom argument does not name a defined Atom. **BadValue** Some numeric value falls outside the range of values accepted by the request. Unless a specific range is specified for an argument, the full range defined by the argument's type is accepted. Any argument defined as a set of alternatives can generate this error.

XFree(3X11), XGetWindowProperty(3X11)
Xlib – C Language X Interface