NAME

XrmUniqueQuark, XrmStringToQuark, XrmPermStringToQuark, XrmQuarkToString, XrmStringToQuarkList, XrmStringToBindingQuarkList - manipulate resource quarks

SYNOPSIS

```
#include <X11/Xresource.h>

XrmQuark XrmUniqueQuark(void);

#define XrmStringToName(string) XrmStringToQuark(string)
#define XrmStringToClass(string) XrmStringToQuark(string)
#define XrmStringToRepresentation(string) XrmStringToQuark(string)

XrmQuark XrmStringToQuark(char *string);

XrmQuark XrmPermStringToQuark(_Xconst char *string);

#define XrmStringToName(string) XrmStringToQuark(string)
#define XrmStringToClass(string) XrmStringToQuark(string)
#define XrmStringToRepresentation(string) XrmStringToQuark(string)

XrmQuark XrmStringToQuark(char *string);

XrmQuark XrmPermStringToQuark(_Xconst char *string);

#define XrmNameToString(name) XrmQuarkToString(name)
#define XrmClassToString(class) XrmQuarkToString(class)
#define XrmRepresentationToString(type) XrmQuarkToString(type)

char *XrmQuarkToString(XrmQuark quark);

#define XrmStringToNameList(str, name)  XrmStringToQuarkList((str), (name))
#define XrmStringToClassList(str,class) XrmStringToQuarkList((str), (class))


void XrmStringToQuarkList(char *string, XrmQuarkList quarks_return);

void XrmStringToBindingQuarkList(_Xconst char *string, XrmBindingList bindings_return,
     XrmQuarkList quarks_return);
```

ARGUMENTS

bindings_return
Returns the binding list.

quark Specifies the quark for which the equivalent string is desired.

quarks_return Returns the list of quarks.

string Specifies the string for which a quark or quark list is to be allocated.

DESCRIPTION

The **XrmUniqueQuark** function allocates a quark that is guaranteed not to represent any string that is known to the resource manager.

These functions can be used to convert from string to quark representation. If the string is not in the Host Portable Character Encoding, the conversion is implementation-dependent. The string argument to **XrmStringToQuark** need not be permanently allocated storage. **XrmPermStringToQuark** is just like **XrmStringToQuark**, except that Xlib is permitted to assume the string argument is permanently allocated, and, hence, that it can be used as the value to be returned by **XrmQuarkToString**.

For any given quark, if **XrmStringToQuark** returns a non-NULL value, all future calls will return the same value (identical address).

These functions can be used to convert from quark representation to string. The string pointed to by the return value must not be modified or freed. The returned string is byte-for-byte equal to the original string passed to one of the string-to-quark routines. If no string exists for that quark, **XrmQuarkToString** returns NULL. For any given quark, if **XrmQuarkToString** returns a non-NULL value, all future calls will return the same value (identical address).

These functions can be used to convert from string to quark representation. If the string is not in the Host Portable Character Encoding, the conversion is implementation-dependent. The string argument to **XrmStringToQuark** need not be permanently allocated storage. **XrmPermStringToQuark** is just like **XrmStringToQuark**, except that Xlib is permitted to assume the string argument is permanently allocated, and, hence, that it can be used as the value to be returned by **XrmQuarkToString**.

For any given quark, if **XrmStringToQuark** returns a non-NULL value, all future calls will return the same value (identical address).

The **XrmStringToQuarkList** function converts the null-terminated string (generally a fully qualified name) to a list of quarks. The caller must allocate sufficient space for the quarks list before calling

XrmStringToQuarkList. Note that the string must be in the valid ResourceName format (see section 15.1). If the string is not in the Host Portable Character Encoding, the conversion is implementation-dependent.

A binding list is a list of type **XrmBindingList** and indicates if components of name or class lists are bound tightly or loosely (that is, if wildcarding of intermediate components is specified).

typedef enum {XrmBindTightly, XrmBindLoosely} XrmBinding, *XrmBindingList;

XrmBindTightly indicates that a period separates the components, and **XrmBindLoosely** indicates that an asterisk separates the components.

The **XrmStringToBindingQuarkList** function converts the specified string to a binding list and a quark list. The caller must allocate sufficient space for the quarks list and the binding list before calling **XrmStringToBindingQuarkList**. If the string is not in the Host Portable Character Encoding the conversion is implementation-dependent. Component names in the list are separated by a period or an asterisk character. If the string does not start with period or asterisk, a period is assumed. For example, "*a.b*c" becomes:

quarks a b c
bindingsloosetightloose

SEE ALSO

XrmGetResource(3), XrmInitialize(3), XrmMergeDatabases(3), XrmPutResource(3)

Xlib - C Language X Interface