

XrmPutResource, XrmQPutResource, XrmPutStringResource, XrmQPutStringResource, XrmPutLineResource – store database resources

```
void XrmPutResource(database, specifier, type, value)
    XrmDatabase *database;
    char *specifier;
    char *type;
    XrmValue *value;

void XrmQPutResource(database, bindings, quarks, type, value)
    XrmDatabase *database;
    XrmBindingList bindings;
    XrmQuarkList quarks;
    XrmRepresentation type;
    XrmValue *value;

void XrmPutStringResource(database, specifier, value)
    XrmDatabase *database;
    char *specifier;
    char *value;

void XrmQPutStringResource(database, bindings, quarks, value)
    XrmDatabase *database;
    XrmBindingList bindings;
    XrmQuarkList quarks;
    char *value;

void XrmPutLineResource(database, line)
    XrmDatabase *database;
    char *line;
```

<i>bindings</i>	Specifies a list of bindings.
<i>database</i>	Specifies the resource database.
<i>line</i>	Specifies the resource name and value pair as a single string.
<i>quarks</i>	Specifies the complete or partial name or the class list of the resource.
<i>specifier</i>	Specifies a complete or partial specification of the resource.
<i>type</i>	Specifies the type of the resource.
<i>value</i>	Specifies the value of the resource, which is specified as a string.

If database contains NULL, **XrmPutResource** creates a new database and returns a pointer to it. **XrmPutResource** is a convenience function that calls **XrmStringToBindingQuarkList** followed by:

```
XrmQPutResource(database, bindings, quarks, XrmStringToQuark(type), value)
```

If the specifier and type are not in the Host Portable Character Encoding, the result is implementation-dependent. The value is stored in the database without modification.

If database contains NULL, **XrmQPutResource** creates a new database and returns a pointer to it. If a resource entry with the identical bindings and quarks already exists in the database, the previous type and value are replaced by the new specified type and value. The value is stored in the database without modification.

If database contains NULL, **XrmPutStringResource** creates a new database and returns a pointer to it.

XrmPutStringResource adds a resource with the specified value to the specified database.

XrmPutStringResource is a convenience function that first calls **XrmStringToBindingQuarkList** on the specifier and then calls **XrmQPutResource**, using a “String” representation type. If the specifier is not in the Host Portable Character Encoding, the result is implementation-dependent. The value is stored in the

database without modification.

If database contains NULL, **XrmQPutStringResource** creates a new database and returns a pointer to it. **XrmQPutStringResource** is a convenience routine that constructs an **XrmValue** for the value string (by calling **strlen** to compute the size) and then calls **XrmQPutResource**, using a “String” representation type. The value is stored in the database without modification.

If database contains NULL, **XrmPutLineResource** creates a new database and returns a pointer to it. **XrmPutLineResource** adds a single resource entry to the specified database. The line should be in valid ResourceLine format (see section 15.1) terminated by a newline or null character; the database that results from using a string with incorrect syntax is implementation-dependent. The string is parsed in the locale of the database. If the **ResourceName** is not in the Host Portable Character Encoding, the result is implementation-dependent. Note that comment lines are not stored.

XrmGetResource(3X11), XrmInitialize(3X11), XrmMergeDatabases(3X11), XrmUniqueQuark(3X11)
Xlib – C Language X Interface