

XtDisplayInitialize, XtOpenDisplay, XtDatabase, XtScreenDatabase, XtCloseDisplay – initialize, open, or close a display

```
void XtDisplayInitialize(app_context, display, application_name, application_class,  
                        options, num_options, argc, argv)
```

```
    XtAppContext app_context;  
    Display *display;  
    String application_name;  
    String application_class;  
    XrmOptionDescRec *options;  
    Cardinal num_options;  
    int *argc;  
    String *argv;
```

```
Display *XtOpenDisplay(app_context, display_string, application_name, application_class,  
                     options, num_options, argc, argv)
```

```
    XtAppContext app_context;  
    String display_string;  
    String application_name;  
    String application_class;  
    XrmOptionDescRec *options;  
    Cardinal num_options;  
    int *argc;  
    String *argv;
```

```
void XtCloseDisplay(display)
```

```
    Display *display;
```

```
XrmDatabase XtDatabase(display)
```

```
    Display *display;
```

```
XrmDatabase XtScreenDatabase(screen)
```

```
    Screen* screen;
```

argc Specifies a pointer to the number of command line parameters.

argv Specifies the command line parameters.

app_context Specifies the application context.

application_class Specifies the class name of this application, which usually is the generic name for all instances of this application.

application_name Specifies the name of the application instance.

display Specifies the display. Note that a display can be in at most one application context.

num_options Specifies the number of entries in the options list.

options Specifies how to parse the command line for any application-specific resources. The options argument is passed as a parameter to **XrmParseCommand**. For further information, see *Xlib – C Language X Interface*.

screen Specifies the screen whose resource database is to be returned.

The XtDisplayInitialize function builds the resource database, calls the Xlib **XrmParseCommand** function to parse the command line, and performs other per display initialization. After **XrmParseCommand** has been called, *argc* and *argv* contain only those parameters that were not in the standard option table or in the table specified by the options argument. If the modified *argc* is not zero, most applications simply print out the modified *argv* along with a message listing the allowable options. On UNIX-based systems, the application name is usually the final component of *argv*[0]. If the synchronize resource is **True** for the specified application, **XtDisplayInitialize** calls the Xlib **XSynchronize** function to put Xlib into synchronous mode for this display connection. If the reverseVideo resource is **True**, the Intrinsics exchange

XtDefaultForeground and **XtDefaultBackground** for widgets created on this display. (See Section 9.6.1).

The **XtOpenDisplay** function calls **XOpenDisplay** the specified display name. If `display_string` is `NULL`, **XtOpenDisplay** uses the current value of the `-display` option specified in `argv` and if no display is specified in `argv`, uses the user's default display (on UNIX-based systems, this is the value of the `DISPLAY` environment variable).

If this succeeds, it then calls **XtDisplayInitialize** and pass it the opened display and the value of the `-name` option specified in `argv` as the application name. If no name option is specified, it uses the application name passed to **XtOpenDisplay**. If the application name is `NULL`, it uses the last component of `argv[0]`.

XtOpenDisplay returns the newly opened display or `NULL` if it failed.

XtOpenDisplay is provided as a convenience to the application programmer.

The **XtCloseDisplay** function closes the specified display as soon as it is safe to do so. If called from within an event dispatch (for example, a callback procedure), **XtCloseDisplay** does not close the display until the dispatch is complete. Note that applications need only call **XtCloseDisplay** if they are to continue executing after closing the display; otherwise, they should call **XtDestroyApplicationContext** or just exit.

The **XtDatabase** function returns the fully merged resource database that was built by **XtDisplayInitialize** associated with the display that was passed in. If this display has not been initialized by **XtDisplayInitialize**, the results are not defined.

The **XtScreenDatabase** function returns the fully merged resource database associated with the specified screen. If the `screen` does not belong to a **Display** initialized by **XtDisplayInitialize**, the results are undefined.

XtAppCreateShell(3Xt), XtCreateApplicationContext(3Xt)

X Toolkit Intrinsic – C Language Interface

Xlib – C Language X Interface