

**NAME**

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

**SYNTAX**

```
void XtDisplayInitialize(XtAppContext app_context, Display *display, String application_name, String application_class, XrmOptionDescRec *options, Cardinal num_options, int *argc, String *argv);
```

```
Display *XtOpenDisplay(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);
```

```
XrmDatabase XtDatabase(Display *display);
```

```
XrmDatabase XtScreenDatabase(Screen* screen);
```

**ARGUMENTS**

<i>argc</i>	Specifies a pointer to the number of command line parameters.
<i>argv</i>	Specifies the command line parameters.
<i>app_context</i>	Specifies the application context.
<i>application_class</i>	Specifies the class name of this application, which usually is the generic name for all instances of this application.
<i>application_name</i>	Specifies the name of the application instance.
<i>display</i>	Specifies the display. Note that a display can be in at most one application context.
<i>num_options</i>	Specifies the number of entries in the options list.
<i>options</i>	Specifies how to parse the command line for any application-specific resources. The options argument is passed as a parameter to <b>XrmParseCommand</b> . For further information, see <i>Xlib – C Language X Interface</i> .
<i>screen</i>	Specifies the screen whose resource database is to be returned.

**DESCRIPTION**

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.

#### SEE ALSO

**XtAppCreateShell(3)**, **XtCreateApplicationContext(3)**

*X Toolkit Intrinsics – C Language Interface*

*Xlib – C Language X Interface*