XtOpenApplication, XtVaOpenApplication – initialize, open, or close a display

**Widget XtOpenApplication**

```c
XtOpenApplication(app_context_return, application_class, options, num_options, argc_in_out, argv_in_out, fallback_resources, widget_class, args, num_args)
```

```c
XtAppContext app_context_return;
String application_class;
XrmOptionDescRec* options;
Cardinal num_options;
int* argc_in_out;
String* argv_in_out;
String* fallback_resources;
WidgetClass widget_class;
ArgList args;
Cardinal num_args;
```

**Widget XtVaOpenApplication**

```c
XtVaOpenApplication(app_context_return, application_class, options, num_options, argc_in_out, argv_in_out, fallback_resources, widget_class, args, num_args, ...
```

```c
XtAppContext app_context_return;
String application_class;
XrmOptionDescRec* options;
Cardinal num_options;
int* argc_in_out;
String* argv_in_out;
String* fallback_resources;
WidgetClass widget_class;
```

**app_context_return**
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.

**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*.

**num_options**
Specifies the number of entries in the options list.

**argc_in_out**
Specifies a pointer to the number of command line parameters.

**argv_in_out**
Specifies the command line parameters.

**fallback_resources**
Specifies resource values to be used if the application class resource file cannot be opened or read, or NULL.

**widget_class**
Specifies the widget class of the shell to be created.

**args**
Specifies the argument list to override any other resource specification for the created shell widget.

**num_args**
Specifies the number of entries in the argument list.

**...**
Specifies the variable argument list to override any other resource specification for the created shell widget.

The **XtOpenApplication** function calls **XtToolkitInitialize** followed by **XtCreateApplicationContext**, then calls **XtOpenDisplay** with `display_string` NULL and `application_name` NULL, and finally calls **XtAppCreateShell** with `application_name` NULL, `widget_class` applicationShellWidgetClass, and the specified `args` and `num_args` and returns the created shell. The modified `argc` and `argv` returned by **XtDisplayInitialize** are returned in `argc_in_out` and `argv_in_out`. If `app_context_return` is not NULL, the created application context is also returned. If the display specified by the command line cannot be opened, an error message is issued and **XtOpenApplication** terminates the application. If `fallback_resources` is
non-NULL, \texttt{XtAppSetFallbackResources} is called with the value prior to calling \texttt{XtOpenDisplay}.

\texttt{XtAppInitialize} and \texttt{XtVaAppInitialize} have been superceded by \texttt{XtOpenApplication} and \texttt{XtVaOpenApplication} respectively.

\texttt{XtOpenApplication(3Xt)}, \texttt{XtVaOpenApplication(3Xt)}  
\textit{X Toolkit Intrinsics – C Language Interface}

\textit{Xlib – C Language X Interface}