

**NAME**

XtCallCallbacks, XtCallCallbackList, XtHasCallbacks – process callbacks

**SYNTAX**

```
void XtCallCallbacks(Widget w, String callback_name, XtPointer call_data);  
void XtCallCallbackList(Widget w, XtCallbackList callbacks, XtPointer call_data);  
typedef enum { XtCallbackNoList, XtCallbackHasNone, XtCallbackHasSome } XtCallbackStatus;  
XtCallbackStatus XtHasCallbacks(Widget w, String callback_name);
```

**ARGUMENTS**

<i>callback_name</i>	Specifies the callback list to be executed or checked.
<i>call_data</i>	Specifies a callback-list specific data value to pass to each of the callback procedure in the list.
<i>callbacks</i>	Specifies the callback list to be executed.
<i>w</i>	Specifies the widget.

**DESCRIPTION**

The **XtCallCallbacks** function calls each procedure that is registered in the specified widget's callback list.

If *callbacks* is not NULL, **XtCallCallbackList** calls each of the callback procedures in the list, passing client data and *call\_data*. The *callbacks* parameter must specify the contents of a widget or object resource declared with representation type **XtRCallback**.

The **XtHasCallbacks** function first checks to see if the widget has a callback list identified by *callback\_name*. If the callback list does not exist, **XtHasCallbacks** returns **XtCallbackNoList**. If the callback list exists but is empty, it returns **XtCallbackHasNone**. If the callback list exists and has at least one callback registered, it returns **XtCallbackHasSome**.

**SEE ALSO**

XtAddCallback(3)  
*X Toolkit Intrinsics – C Language Interface*  
*Xlib – C Language X Interface*