

**NAME**

XOpenIM, XCloseIM, XSetIMValues, XGetIMValues, XDisplayOfIM, XLocaleOfIM, XRegisterIMInstantiateCallback, XUnregisterIMInstantiateCallback - open, close, and obtain input method information

**SYNTAX**

```
XIM XOpenIM(Display *display, XrmDatabase db, char *res_name, char *res_class);
```

```
Status XCloseIM(XIM im);
```

```
char *XSetIMValues(XIM im, ...);
```

```
char *XGetIMValues(XIM im, ...);
```

```
Display *XDisplayOfIM(XIM im);
```

```
char *XLocaleOfIM(XIM im);
```

```
Bool XRegisterIMInstantiateCallback(Display *display, XrmDatabase db, char *res_name, char  
*res_class, XIDProc callback, XPointer client_data);
```

```
Bool XUnregisterIMInstantiateCallback(Display *display, XrmDatabase db, char *res_name, char  
*res_class, XIDProc callback, XPointer client_data);
```

**ARGUMENTS**

- |                    |   |
|--------------------|---|
| <i>callback</i>    | Specifies a pointer to the input method instantiate callback.         |
| <i>client_data</i> | Specifies the additional client data.                                 |
| <i>db</i>          | Specifies a pointer to the resource database.                         |
| <i>display</i>     | Specifies the connection to the X server.                             |
| <i>im</i>          | Specifies the input method.   |
| <i>res_class</i>   | Specifies the full class name of the application.                     |
| <i>res_name</i>    | Specifies the full resource name of the application.                  |
| <i>...</i>         | Specifies the variable length argument list to set or get XIM values. |

**DESCRIPTION**

The **XOpenIM** function opens an input method, matching the current locale and modifiers specification. Current locale and modifiers are bound to the input method at opening time. The locale associated with an input method cannot be changed dynamically. This implies that the strings returned by **XmbLookupString** or **XwcLookupString**, for any input context affiliated with a given input method will be encoded in the locale current at the time the input method is opened.

The specific input method to which this call will be routed is identified on the basis of the current locale. **XOpenIM** will identify a default input method corresponding to the current locale. That default can be modified using **XSetLocaleModifiers** for the input method modifier.

The db argument is the resource database to be used by the input method for looking up resources that are private to the input method. It is not intended that this database be used to look up values that can be set as IC values in an input context. If db is NULL, no database is passed to the input method.

The res\_name and res\_class arguments specify the resource name and class of the application. They are intended to be used as prefixes by the input method when looking up resources that are common to all input contexts that may be created for this input method. The characters used for resource names and classes must be in the X Portable Character Set. The resources looked up are not fully specified if res\_name or res\_class is NULL.

The res\_name and res\_class arguments are not assumed to exist beyond the call to **XOpenIM**. The specified resource database is assumed to exist for the lifetime of the input method.

**XOpenIM** returns NULL if no input method could be opened.

The **XCloseIM** function closes the specified input method and frees the memory pointed to by the im argument.

The **XSetIMValues** function presents a variable argument list programming interface for setting attributes of the specified input method. It returns NULL if it succeeds; otherwise, it returns the name of the first argument that could not be set. Xlib does not attempt to set arguments from the supplied list that follow the failed argument; all arguments in the list preceding the failed argument have been set correctly.

The **XGetIMValues** function presents a variable argument list programming interface for querying properties or features of the specified input method. This function returns NULL if it succeeds; otherwise, it returns the name of the first argument that could not be obtained.

Each XIM value argument (following a name) must point to a location where the XIM value is to be

stored. That is, if the XIM value is of type T, the argument must be of type T\*. If T itself is a pointer type, then **XGetIMValues** allocates memory to store the actual data, and the client is responsible for freeing this data by calling **XFree** with the returned pointer.

The **XDisplayOfIM** function returns the display associated with the specified input method.

The **XLocaleOfIM** function returns the locale associated with the specified input method.

The **XRegisterIMInstantiateCallback** function registers a callback to be invoked whenever a new input method becomes available for the specified display that matches the current locale and modifiers.

The function returns **True** if it succeeds; otherwise, it returns **False**.

The **XUnregisterIMInstantiateCallback** function removes an input method instantiation callback previously registered. The function returns **True** if it succeeds; otherwise, it returns **False**.

#### SEE ALSO

XCreateIC(3), XSetICFocus(3), XSetICValues(3), XmbResetIC(3)

*Xlib - C Language X Interface*