Release Notes

OSF/Motif Release 1.2

The release notes in this document are based on Chapter 3 of the *OSF/Motif Release Notes* for Release 1.2 and include a few other notes relevant to programmers developing OSF/Motif applications. Most of the notes describe changes made for OSF/Motif Release 1.2. The first two notes discuss performance improvements and information about backward compatibility.

The notes support the OSF/Motif software provided with the DEC OSF/1 software, DECwindows Motif for OpenVMS, the eXcursion Software Development Kit, and ULTRIX Worksystem Software.

For the OSF/Motif Release 1.2.3 patch, approximately 425 problems were fixed. The DEC OSF/1 system includes a version of the OSF/Motif Open Bug List in `/usr/doc/motif/OPENBUGS`.

1.1 Performance Improvements

The Open Software Foundation set as its goal for OSF/Motif Release 1.2 to improve performance where possible and, at a minimum, not allow it to degrade below the performance of OSF/Motif Release 1.1.

Performance testing was done in three areas:

- Aspects obvious to end users
- Data space usage
- Memory leakage

The performance of user-perceptible events such as posting and unposting dialog boxes and pop-up menus are comparable or better than the performance for OSF/Motif Release 1.1.4. A significant improvement was made for scrolling inside a Scrolled Text region that contains a large amount of test.

Data space usage has improved throughout the OSF/Motif Toolkit, particularly for the Text widget. In some cases, data space usage has been reduced by as much as 40%. Also, the memory used for the text in a Text widget is now correctly reduced when that text is replaced by a smaller amount of text.

Memory leakage has been reduced to a minimum for multiple creates and destroys of all Toolkit widgets. Although the Motif tests showed small memory leaks, the OSF believes that this amount of memory is required as part of the startup overhead and is not a true memory leak.

The OSF did find some memory leaks that they plan to fix in a future release. Three widgets – File Selection Box, Command, and Drawn Button – leak approximately 500 bytes of memory per instance.

1.2 Backward Compatibility

The OSF tested OSF/Motif Release 1.2 for both binary compatibility as well as visual and behavioral compatibility.

1.2.1 Binary Compatibility

Binary compatibility was checked throughout the development of OSF/Motif Release 1.2 by compiling the VTS test directories or certain automated test directories into the object code using the Release 1.1.4 header files. The object files were then linked with the OSF/Motif Release 1.2 libraries and the resulting executables were run. Other testing involved relinking certain demo and X-client programs and running their executables. All these test completed satisfactorily.

Note, however, that no tests were conducted to ensure that applications using the OSF/Motif Release 1.1 shared libraries could run without any modifications using the OSF/Motif Release 1.2 shared libraries.

Custom widgets that are a subclass of any Motif widget, except the Text widget, should be binary compatible with OSF/Motif Release 1.2.

1.2.2 Visual and Behavioral Compatibility

The OSF ran automated tests that compared current visuals with those recorded using OSF/Motif Release 1.1.4 libraries. Once all differences between the Release 1.2 and Release 1.1.4 versions were accounted for, the visuals were rerecorded using Release 1.2 visuals. These new recorded visuals were used in all subsequent tests.

The OSF believes that Motif Release 1.2 is visually and behaviorally compatible with Release 1.1.4. However, they have made extensive improvements to the Traversal and Geometry Management algorithms that result in some differences between those versions. These differences reflect efforts to fix defects in the earlier release.

For example, one such modification involves the new policy in which an initial size set for a manager widget in an application is now honored by the Toolkit. In OSF/Motif Release 1.1, applications set the initial size for a manager widget, but did not, in fact, use that size. In Release 1.2, Motif now uses that size setting and the initial layout is changed accordingly.

1.3 Changes and New Features for OSF/Motif Release 1.2

This section summarizes changes and new features that the OSF has made to OSF/Motif Release 1.2. Detailed information about these modifications is contained in the Motif reference pages and the Motif Release 1.2 revisions of following books:

- *OSF/Motif Style Guide*
- *OSF/Motif Programmer’s Guide*

The following sections discuss the OSF/Motif Release 1.2 enhancements.

1.3.1 General Toolkit Changes

This section discusses the changes made to the overall OSF/Motif Release 1.2 Toolkit.

1.3.1.1 Include File Changes

In OSF/Motif Release 1.1, links to the `AtomMgr.h`, `Protocols.h`, `ProtocolsP.h`, and `MwmUtil.h` header files in `Xm` were placed in the X11 directory in both the `make includes` and `make install` procedures. This change resulted in backwards compatibility with OSF/Motif Release 1.0. Now, for OSF/Motif Release 1.2, the `make includes` and `make install` procedures no longer create these links. You must include the files as `<Xm/...>` instead of `<X11/...>`.

The following header files that were in OSF/Motif Release 1.1 are obsolete in Release 1.2:

- `ExtObject.h`
- `Traversal.h`
- `VaSimple.h`
- `VendorE.h`
- `VendorEP.h`

A new public header file, `XmAll.h`, has been added to OSF/Motif Release 1.2. This header file includes all the documented header files.

1.3.1.2 Change in XT Translations

As a result of fixing a problem in the `Xt` translation code (Patch 25 for X11 R5), the translations in `Xt` are now handled strictly and no longer accept any possible match as they did before. This change has caused a change in the behavior of the QATS and Motif VTS test suites which now make incorrect assumptions for certain keyboards, such as those that have the arrow keys defined in the keypad. Other Motif applications might be affected as well.

You can avoid this problem by creating a file to redefine the bindings for the keys in question and then passing that file to the `xmodmap` utility.

The following example changes the bindings for the keypad keys that match the arrow keys so that only the arrow keys are produced, not the keypad numbers. The new definitions allow the application to use modifiers with the arrow keys.

```
!
! Always force:
!   KP_2 = Down
!   KP_8 = Up
!   KP_4 = Left
!   KP_6 = Right
keycode 120 = Down
keycode 76  = Up
keycode 98  = Left
keycode 100 = Right
```

1.3.2 ANSI C Compliance

All references to `caddr_t` have been changed to `XtPointer`. This change affects all callback routines and any other routines that reference `caddr_t`. The OSF made this change so that OSF/Motif Release 1.2 would be compliant with the ANSI C specification.

1.3.3 Display and Screen Specific Data

Motif now has an `XmDisplay` object that supports per-display data and resources. An `XmScreen` object has been added that supports per-screen data and resources.

1.3.4 Drag and Drop

OSF/Motif Release 1.2 supports the drag and drop metaphor for data interchange. The Drag and and Drop specification has been fully implemented. See the Release 1.2 version of the *OSF/Motif Programmer's Guide* for information on the drag and drop interface.

Note

If you want to use `Btn2` to have pop-up menus pop up, Drag and Drop will not function properly. You need to disable Drag and Drop in such instances.

1.3.5 Tear-Off Menus

With tear-off menus, the user can retain menus on the display area for subsequent selections. Each tearable menupane is a tear-off button. When the mouse drag button is pressed on the tear-off button, the pane tears off and can be dragged and then placed by releasing the mouse drag button. The window manager surrounds the tear-off menupane with a menu button and a title. Shifting focus to a torn-off menu's windowpane follows the standard window manager policy.

Tear-off behavior is enabled by setting the `XmNtearOffModel` resource to `XmTEAR_OFF_ENABLED`. (The default is `XmTEAR_OFF_DISABLED`).

Note that there is no resource converter preregistered by `XmNtearOffModel`. To allow the tear-off function to be enabled through the resource database, an application must register its own resource converter for the `XmNtearOffModel` resource using the `XmRepTypeInstallTearOffModelConverter` function.

The converter is not automatically installed because many applications use map or cascading callbacks to dynamically set the sensitivity of items within their menus. However, if a tear-off menu is mapped, the sensitivity of its menu items must be changed immediately to reflect changes in other application states. Existing applications are unlikely to change menu item sensitivity in this manner. Thus, allowing their menus to be torn off could result in operations being enabled at unexpected times. If a user activates one of these menu items, the application can crash or the persistent data can be corrupted.

1.3.6 Insensitive Visuals

Motif provides visual indications to show whether a component can respond to input from users. Labels and buttons have had this behavior in previous Motif releases. In OSF/Motif Release 1.2, this behavior has been extended to the following widgets:

- `XmArrowButton`
- `XmList`
- `XmScrollBar`
- `XmText`
- `XmTextField`

1.3.7 Other Visual Changes

OSF/Motif Release 1.2 has made other visual changes as follows:

- Several pixels have changed in the 3-dimensional beveled look.
- Minor changes have been made to the color generation routines. In particular, the new `XmScreen` object contains resources that allow for tailoring the generation of default colors. Some of the default values for thresholds have been adjusted to produce more contrast on color monitors.
- Motif now adds a location cursor to surround all items in a List whenever a List widget has the focus and the current keyboard item is not visible.
- There are minor layout differences because of fixes in geometry management.

1.3.8 Titles for Frames

In OSF/Motif Release 1.2, you can specify a Title widget in a Frame widget. The release has added the following new constraint resources for specifying the position and alignment of the title in the Frame:

- `XmNchildHorizontalAlignment`
- `XmNchildHorizontalSpacing`
- `XmNchildType`
- `XmNchildVerticalAlignment`

1.3.9 Audible Warning

The VendorShell has a new resource, `XmNaudibleWarning`, that can specify whether an audible cue should accompany a warning message. Text widgets determine the value for this resource from the value of `XmNaudibleWarning`.

1.3.10 Color Enhancements

The following three resources have been added to the `XmScreen` widget so that users can specify the default background color and thresholds for shadow calculation:

- `XmNlightThreshold`

- `XmNdarkThreshold`
- `XmNforegroundThreshold`

Motif has added the `XmChangeColor` function that changes the background and other colors for a specified widget.

1.3.11 Baseline Alignment

Motif has added two functions for baseline alignment. The `XmWidgetGetBaselines` function determines the position of the widget's text baseline. The `XmWidgetGetDisplayRect` function determines the size and position of the bounding box for the widget's character string.

1.3.12 Expanded Traversal Set

In OSF/Motif Release 1.2, you can use more widgets to support traversal using the keyboard. For example, inside a tab group, users can now use the arrow keys to traverse to all control descendants that are not contained within a nested tab group and that are eligible to receive focus, even if the controls are not direct children of the tab group.

1.3.13 Two-Dimensional Menu Traversal

With OSF/Motif Release 1.2, the left, right, up, and down traversal arrows now navigate within a menupane. The up and down arrow keys wrap between columns. The right and left arrow keys post the previous or next menupane when they are pressed in the rightmost and leftmost column of the menupane, respectively.

1.3.14 Input Focus

OSF/Motif Release 1.2 has added the `XmNinitialFocus` resource to the Manager class. This resource specifies the first widget to receive input focus. This resource can only specify a widget; it is ignored for all pop-up menus, menubars, option menus, and pull-down menus.

1.3.15 Traversal Access Functions

OSF/Motif Release 1.2 has added the following new functions to support better interaction with keyboard traversal:

- `XmGetFocusWidget`
- `XmGetTabGroup`
- `XmGetVisibility`
- `XmIsTraversable`
- `XmIsVisible`

The `XmTrackingLocate` function has been modified to do the following:

- Field all events, not just a button press.
- Return on any keystroke or button press.

- Be called for nonsensitive widgets.

In addition, the `XmTrackingEvent` function has been added. This function is similar to `XmTrackingLocate`, except that it returns a pointer to the X event.

1.3.16 Virtual Keys

OSF/Motif Release 1.2 has added the `XmTranslateKey` function that allows applications to override the default `XtKeyProc` to handle Motif virtual keys.

Motif defines two new virtual `keysyms`:

- `osfPageLeft`
- `osfPageRight`

You must have the X11 Release 5 `XKeysymDB` installed in `/usr/lib/X11` to use these new virtual keys. Otherwise you get a warning message on application startup. Specify the following information in the `XKeysymDB` file:

```
osfPageLeft                     :1004FF40
osfPageUp                       :1004FF41
osfPageDown                     :1004FF42
osfPageRight                    :1004FF43
```

To comply with the *OSF/Motif Style Guide*, the default binding for `osfMenu` has been changed from `<Key>F4` to `Shift<Key>F10`.

In X11 Release 5, the HP `keysyms` in the `XKeysymDB` file have the prefix `hp`. This prefix is not reflected in the HP bindings file in the `/bindings` directory. If you are using an X11 Release 5 `XKeysymDB` file, you might see warning messages at application startup. To eliminate these warning messages, add the `hp` prefix as follows to the appropriate lines in the file:

```
osfDelete       :               <Key>hpDeleteChar
osfInsert       :               <Key>hpInsertChar
osfPrimaryPaste :               <Key>hpInsertLine
osfQuickPaste   :               <Key>hpDeleteLine
```

OSF/Motif Release 1.2 has a new client, `xmbind`, that sets up the virtual bindings for use by Motif applications. Since virtual binding is automatically set up at Motif Window Manager (MWM) startup, you only need to use `xmbind` if MWM is not used or if you need to change the virtual bindings without restarting MWM.

Virtual bindings can now be specified by individual vendors. If there is no `.motifbind` file in the home directory, you can use the `xmbind.alias` file to provide a mapping from the server vendor name to the bindings file. You can set up user vendor bindings as well as system-wide vendor bindings.

1.3.17 Resource Management

OSF/Motif Release 1.2 has the following new functions for managing representation types:

- `XmRepTypeRegister`
- `XmRepTypeAddReverse`
- `XmRepTypeValidValue`

- XmRepTypeGetRegistered
- XmRepTypeGetId
- XmRepTypeGetNameList
- XmRepTypeGetRecord

These functions are useful for developers who want to define new resource converters that use an enumerated set of values.

1.3.18 Changes for CUA and Windows Compliance

In OSF/Motif Release 1.2, pressing the Return key or using the key bound to osfActivate (usually the Enter key on the numeric keypad) no longer activates a button that is outside a menu. For example, pressing such a key no longer pops up an OptionMenu or activates a ToggleButton in a dialog box.

If your application has a default button associated with an XmBulletinBoard, pressing Return (except in a multiline XmText), Ctrl/Return, or the key bound to osfActivate while the focus is in the XmBulletinBoard now activates the default button.

1.4 Changes and Enhancements to Specific Widgets

This section summarizes the changes to specific widgets that were made in OSF/Motif Release 1.2.

1.4.1 XmClipboard

OSF made several corrections to the XmClipboard function parameters. These changes are binary compatible with earlier releases of Motif. However, in some instances, you might see warning messages when you recompile your applications. The modifications involved changing (char *) to XtPointer, int to long, and (int *) to (long *). The related functions affected by these modifications are:

- XmClipboardCopy
- XmClipboardCopyByName
- XmClipboardInquireCount
- XmClipboardInquireFormat
- XmClipboardRegisterFormat
- XmClipboardRetrieve
- XmClipboardStartCopy
- XmClipboardWithdrawFormat

1.4.2 XmCommand

In OSF/Motif Release 1.2, a correction was made to XmCommandGetChild so that it now accepts XmDIALOG_WORK_AREA as a value for the child argument.

1.4.3 XmList

To enchance its capabilities for managing lists, OSF/Motif Release 1.2 includes the following new functions:

- XmListAddItemsUnselected
- XmListDeletePositions
- XmListGetKbdItemPos
- XmListIsPosSelected
- XmListPosToBounds
- XmListReplaceItemsUnselected
- XmListReplaceItemsPosUnselected
- XmListReplacePositions
- XmListSetKbdItemPos
- XmListUpdateSelectedList
- XmListYToPos

The XmList widget includes a new translation:

- <Copy>

  Copies the selection to the clipboard.

XmList includes a new action:

- ListScrollCursorVertically()

  Scrolls the cursor vertically based on an input percentage or a y position.

Note

> This action was mistakenly named ListScrollCursorVisible in OSF/Motif Release 1.2. The name will be corrected in a later release of Motif.

The XmNvisibleItemCount resource has been modified so that the default value is dynamic, and is based on the item count and the height.

In OSF/Motif Release 1.2, if the selectedItems and selectedItemCount resources for a list are set in a resource file, the location cursor appears over the last item in the selectedItems list, not the first selected item.

1.4.4 XmMessageBox

In OSF/Motif Release 1.2, MessageBox supports the addition of one MenuBar, one work area, and multiple PushButton children.

A new dialog type, XmDIALOG_TEMPLATE, creates a MessageBox that contains only a Separator. The application provides additional children.

XmCreateTemplateDialog creates an XmDIALOG_TEMPLATE XmMessageBox inside a DialogShell.

1.4.5 XmRowColumn and Menus

OSF/Motif Release 1.2 adds a new resource, `XmNentryVerticalAlignment`, that specifies the vertical alignment style.

Another resource, `XmNunpostBehavior` has been added to the `XmScreen` object. This resource can be set to enable external button events to be replayed after a menu is unposted.

1.4.6 XmScrollBar

In OSF/Motif Release 1.2, `XmScrollBar` includes a new translation:

- `<Cancel>`

  Cancels the current slider drag.

1.4.7 XmScrolledWindow

OSF/Motif Release 1.2 adds the function, `XmScrollVisible`, that scrolls an automatic scrolled window to make a partially or completely obscured widget visible.

Another resource, `XmNtraverseObscuredCallback`, has been added that specifies a list of callbacks that are called when a traversal event is requested to a nonvisible widget. A new callback structure, `XmTraverseObscuredCallbackStruct`, has been added to support this callback.

1.4.8 XmSelectionBox, XmFileSelectionBox

In OSF/Motif Release 1.2, the `XmSelectionBox` and `XmFileSelectionBox` widgets support the addition of `MenuBar` and `PushButton` children, as well as a work area child.

A new resource, `XmNchildPlacement`, controls the location of the work area child.

The value, `XmDIALOG_TEMPLATE`, has been added to the `XmNdialogType` resource.

By default, `XmSelectionBoxDialog` and its subclasses use `XmTextField` instead of `XmText`. You can revert to the earlier behavior by defining `USE_TEXT_IN_DIALOGS` when your application builds `XmSelectionBox` or any of its subclasses.

1.4.9 XmText

OSF/Motif Release 1.2 has added two functions to `XmText` for making update changes to the widget: `XmTextDisableRedisplay` and `XmTextEnableRedisplay`.

Two other functions facilitate string manipulation: `XmTextFindString` and `XmTextGetSubstring`.

In Release 1.2, the destination cursor now follows the insert cursor and is no longer independently drawn.

`XmText` includes three new translations:

- `<Backspace>`

  Deletes any non-null primary selection.

- `<Delete>`

  Deletes any non-null primary selection.

- `<LeaveWindow>`

  Continues a selection action by scrolling after a time delay.

`XmText` includes two new actions:

- `scroll-cursor-vertically()`

  Scrolls the cursor vertically based on a `y` position.

- `toggle-overstrike()`

  Switches between insert and overstrike modes.

Note

There is a potential problem in both `XmText` and `XmTextField` with rendering strings in fonts or font sets that contain characters whose ascenders can rise above the font ascent. If the text containing these characters is highlighted, any overlapping descenders in the previous line may be overwritten by the ascenders in the succeeding line.

1.4.10 XmTextField

OSF/Motif Release 1.2, the `XmTextField` widget has a new resource, `XmNfocusCallback`, that specifies the callbacks to be called when the widget accepts input focus.

Another new function, `XmTextFieldGetSubstring`, gets a substring by length from a widget.

In Release 1.2, the destination cursor now follows the insert cursor and is no longer independently drawn.

`XmTextField` includes two new translations:

- `<Backspace>`

  Deletes any non-null primary selection.

- `<Delete>`

  Deletes any non-null primary selection.

1.4.11 XmToggleButton, XmToggleButtonGadget

In OSF/Motif Release 1.2, setting `XmNfillOnSelect` to be true when `XmNindicatorOn` is false now causes the background of a set `XmToggleButton` to be filled with `XmNselectColor`.

The default value for `XmNfillOnSelect` is dynamic so that it matches the state of `XmNindicatorOn`.

1.5 Motif Window Manager Enhancements

This section highlights the enhancements to the Motif Window Manager (MWM).

1.5.1 Changes to MWM

OSF/Motif Release 1.2 incorporates the following enhancements to MWM:

- An internationalized `.mwmrc` file
- Internationalized dialog messages
- A built-in default root menu
- Support for the `<Return>` continuation character (\) in the `.mwmrc` file
- Search capability with `XBMLANGPATH` for bitmap files
- Support for pop-down and replay event behavior in `mwm` menus
- Documentation of the widget names used by `mwm`
- Support for scrolled window traversal to scrolled-off children in the icon box
- Support for treating the `<Alt>` and `<Meta>` key modifiers as two distinct modifiers
- Support for the `SHAPE` nonrectangular window extension

1.5.2 New and Enhanced MWM Resources

The OSF/Motif Release 1.2 window manager includes the following new or enhanced resources:

- `feedbackGeometry`

  Sets the position of the move/resize feedback window. The default position is the center of the screen.

- `frameBorderWidth`

  Now bases its default value on the size and resolution of the screen.

- `iconPlacement`

  Now takes the addition value, `tight`, which specifies automatic icon placement with no gaps between icons.

- `maximumClientSize`

  Can now take the values `vertical` and `horizontal`.

- `moveOpaque`

  Controls whether an image of the window or just an outline of the window is moved.

- `resizeBorderWidth`

  Now bases its default value on the size and resolution of the screen.

- `usePPosition`

  Uses the values of `on`, `off`, or `nonzero` to control whether program-specified positions are used.

1.5.3 New and Enhanced MWM Functions

OSF/Motif Release 1.2 has the following new and enhanced MWM functions:

- `f.lower`

  Includes a `within` argument to move the window within the application stacking order, but retains the parent window below the children rule. The function also includes a `freeFamily` argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

- `f.minimize`

  Can now be used from an icon in an icon box.

- `f.raise`

  Includes a `within` argument to move the window within the application stacking order, but retains the parent window below the children rule. The function also includes a `freeFamily` argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

- `f.raise_lower`

  Includes a `within` argument to move the window within the application stacking order, but retains the parent window below the children rule. The function also includes a `freeFamily` argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

- `f.restore`

  Restores a window to its previous state. Double clicking on a root icon is bound to this function rather than to `f.normalize`.

- `f.restore_and_raise`

  Restores a window to its previous state and raises it to the top of the window stack. Double clicking on an icon in an icon box is bound to this function rather than to `f.normalize`.

- `f.screen`

  Traverses to the screen specified by *`arg`*. Legal values for *`arg`* are: `next`, `prev`, `last`, or a specific screen number.

1.5.4 New MWM Action

OSF/Motif Release 1.2 has one new MWM action:

- `<Alt> <Esc>`

  This key combination behaves similarly to `f.next_key`, except that the window is always raised, regardless of the value of `focusAutoRaise`.

1.6 Changes to the User Interface Language (UIL)

OSF/Motif Release 1.2 has the following changes in the User Interface Language (UIL):

- There is a new command flag, `–s`, that enables the use of `setlocale` and the creation of localized Compound Strings.

Note

There is a serious problem in parsing double quoted strings with the `–s` flag. If you need to use this flag for parsing double quoted strings, you must obtain a patch from the OSF.

- New UIL syntax now supports font sets and font tables.
- New UIL syntax now supports wide character strings.
- Support has been added for using widget references as callback tags.
- New UIL syntax specifies the resources of automatically created children.
- Syntax changes to the Widget Meta-Language (WML) allow for the definition of automatically created children of composite widgets.
- UIL can now use the `–wmd` *`file`* flag to read binary databases (WML files) that contain WML information.
- Mrm includes two new functions:
  - `MrmOpenHierarchyPerDisplay`

    This function is the same as the old `MrmOpenHierarchy` function, except that in the new function, the display is passed as an explicit argument. This function replaces `MrmOpenHierarchy`.
  - `MrmFetchBitmapLiteral`

    This function fetches a bitmap literal with a depth of 1.