E
OPEN LOOK User-interface Compliance

This appendix lists the ways that the XView Toolkit is not compliant with the *OPEN LOOK Graphical User Interface Functional Specification*. It is not a complete list of the ways that OpenWindows 3.0 is not an OPEN LOOK UI-compliant environment. OPEN LOOK UI compliance has two components: toolkit compliance and environment compliance.

An OPEN LOOK UI-compliant toolkit allows a developer to write an application that will be OPEN LOOK UI-compliant if run with an OPEN LOOK UI window manager. The toolkit might also support the application running successfully with, for example, a MOTIF™ window manager, but in such a configuration, the application would not be OPEN LOOK UI-compliant. An OPEN LOOK UI-compliant environment consists of an OPEN LOOK UI window manager, file manager, workspace properties window, and other such utility programs. To guarantee an OPEN LOOK UI application, the developer must write the application with an OPEN LOOK UI-compliant toolkit *and* run the application in an OPEN LOOK UI-compliant environment.

This list is in three parts. The first part consists of those features missing from XView 3.0 that are specified as Level 1 OPEN LOOK UI features. The second part lists some of the Level 2 OPEN LOOK UI features supported by XView 3.0. The third part lists the rest of the Level 2 OPEN LOOK UI features, which are not supported by XView 3.0.

E.1 Level 1 Features Not Supported in XView 3.0

The Level 1 features listed on the following pages are not supported in XView 3.0.

E.1.1 Keyboard and Mouse Customization

XView 3.0 hard-codes the bindings for function keys, mouse buttons, and mouse modifiers that OPEN LOOK UI says the user should be able to customize.

An OPEN LOOK UI toolkit should allow the user to specify the keys used for CUT, COPY, PASTE, PROPERTIES, UNDO, CANCEL, DEFAULTACTION, NEXTFIELD, and PREVFIELD.

- In XView 3.0, these key bindings are hard-coded to L10, L6, L8, L3, L4, L1, Return, Tab, and Shift-Tab.

An OPEN LOOK UI toolkit should allow the user to change the mouse buttons used for SELECT, ADJUST, and MENU; the specified default mouse button bindings are LEFT, MIDDLE, and RIGHT.

- In XView 3.0, the specified default mouse button bindings are hard-coded.

An OPEN LOOK UI toolkit should allow the user to change the mouse modifiers used for SETMENUDEFAULT, DUPLICATE, PAN, and CONSTRAIN. The specified default modified mouse actions are Control-RIGHT for SETMENUDEFAULT, Control-LEFT for DUPLICATE, Meta-LEFT for PAN, and LEFT-and-MIDDLE-chorded for CONSTRAIN.

- In XView 3.0, the specified defaults for SETMENUDEFAULT and DUPLICATE are hard-coded, and PAN and CONSTRAIN are not supported.

In an OPEN LOOK UI toolkit, clicking ADJUST when there is no selection will set an initial insert point (the same as clicking SELECT).

- In XView 3.0, this selects a single character.

In an OPEN LOOK UI toolkit, the user can bind a modified mouse action to selecting a single character, but the default binding is NONE.

- In XView 3.0, selecting a single character is hard-coded to clicking ADJUST when there is no selection.

E.1.2 Default Buttons in Pop Ups

In an OPEN LOOK UI toolkit, notices, command windows, and property windows must always have a "default button," even when there is only one button, and that the DEFAULT-ACTION keyboard accelerator always invokes the default button.

- XView 3.0 does not provide this automatically, but applications can implement this feature using XView primitives.

E.1.3 Help

In an OPEN LOOK UI toolkit, help text can have bold text, italic text, and glyphs (small pictures).

- XView 3.0 does not support this.

E.1.4 Window Background

In an OPEN LOOK UI toolkit, window backgrounds are used to access the Window menu, to select a window, and to move it by dragging.

- XView 3.0 does not support this.

E.1.5 Notices

In an OPEN LOOK UI toolkit, notices do not freeze the screen; input to other applications is always possible.

- In XView 3.0, making notices that freeze the screen is determined by the user (application programmer).

In an OPEN LOOK UI toolkit, each window of an application displays the standard busy pattern in the header when a notice is displayed.

- XView 3.0 this is determined by the user (application programmer).

E.1.6 Text Functions

In an OPEN LOOK UI toolkit, an `UNDO` after an `UNDO` reverses the effect of the `UNDO`, restoring the original state.

- In XView 3.0, the second `UNDO` undoes the next-previous edit. There is no way in XView 3.0 to reverse the effect of an `UNDO`.

E.1.7 Control Items

In an OPEN LOOK UI toolkit, an abbreviated menu button can have a text field to the right of the menu button. The text field is used to add items to the menu.

- XView 3.0 the application level supports this behavior as an option.

In an OPEN LOOK UI toolkit, menu buttons (and abbreviated menu buttons) highlight on `MENU` down, and change to the standard busy pattern on `MENU-up` in stay-up mode.

- XView 3.0 does not provide this.

In an OPEN LOOK UI toolkit, a default setting for exclusive and non-exclusive settings is only displayed when the controls are used on a menu.

- XView 3.0 indicates a default setting for exclusive and non-exclusive settings even when the controls are used in command and property windows.

In an OPEN LOOK UI toolkit, an indeterminate state is defined on exclusive and non-exclusive settings.

- XView 3.0 does not support this.

In an OPEN LOOK UI toolkit, the bold border width on exclusive and non-exclusive settings is adjusted (to either 2 or 3 pixels) depending on the display resolution.

- XView 3.0 does not provide this.

E.1.8 Property Windows

In an OPEN LOOK UI toolkit, there is a required Settings pop-up menu for a property window.

- XView 3.0 does not provide this automatically, but applications can implement this menu themselves.

In an OPEN LOOK UI toolkit, property windows have two required buttons, `Apply` and `Reset`, and an optional button, `Set Default`.

- Again, XView 3.0 does not provide this automatically, but applications can implement this feature using XView primitives.

In an OPEN LOOK UI toolkit, there is a way to have two active selections when using property windows.

- XView 3.0 does not support this.

In an OPEN LOOK UI toolkit, when there are two active selections, the `Apply` button becomes a menu button with `Original Selection` and `New Selection` items.

- XView 3.0 does not support this.

E.2 Level 2 Features Supported in XView 3.0

The following Level 2 features are supported in XView 3.0.

- Abbreviated buttons.
- Nonstandard basic windows.
- Numeric text fields with increment/decrement buttons.
- Some keyboard accelerators.
- Splittable panes.
  - *Missing*: Dimming the pane's border or its contents to indicate that the pane is about to disappear when removing a split pane using cable anchors.
- Split View and Join Views items on Scrollbar menu.
- Dragging text to move/copy.

- Quick Move and Quick Duplicate on (most) text.
- Some Level 2 Workspace Properties.
- Blocking pop-up windows.
- Multi-line text areas.
- Read-only gauges.
- Automatic scrolling.
- View must be updated while scrollbar elevator is dragged.
- Glyphs in scrolling lists.
- Sliders with end-boxes and tickmarks.
- Vertical sliders.
- Soft function keys.
- Window scaling.

E.3 Level 2 Features Not Supported in XView 3.0

The following Level 2 features are not supported in XView 3.0.

- Change bars in property windows.
- Edit menu for text fields.
- Menus containing more than one type of control.
- Resizable panes.
- Selectable panes.
- Minimum scrollbar.
- Page-oriented scrollbar.
- Panning.
- Hierarchical scrolling lists.