UIM/X

Using UIM/X with the Magic Desktop

ICS

Integrated Computer
Solutions Incorporated

Integrated Computer Solutions, Inc.

54 Middlesex Turnpike, Bedford, MA 01730

Tel: 617.621.0060

Fax: 617.621.9555

E-mail: info@ics.com

WWW: http://www.ics.com

UIM/X Trademarks

Contents

Preface

Overview

This guide explains the special issues concerned with running UIM/X on your Magic desktop.

Who Should Use this Guide

This manual assumes you are familiar with the basics of UIM/X. Before using this manual, review the *UIM/X Beginner's Guide* and the *UIM/X User's Guide.*

This manual also assumes that you have some knowledge of programming, a general understanding of the X Window System, and familiarity with UIM/X and the the Magic desktop. You should also know how to use common items such as menus, buttons, and scroll bars. If you are not familiar with these items, you may find it useful to review *Iris Essentials*, the *OSF/Motif User's Guide* and the *UIM/X Motif Developer's Guide*.

Before you begin, check with your system administrator to ensure that the software has been installed as described in the *UIM/X Installation Guide*.

Before You Read this Guide

This guide makes the following assumptions:

- You are familiar with the basic functions of selecting from menus and dialog boxes; opening, moving, resizing and closing windows, and clicking icons.
- You understand the functions of the three mouse buttons, which this guide refers to as the Select button (left button), the Adjust button (middle), and the Menu button (right). See “Using the Mouse” on page ix for more information.

The UIM/X Document Set and Related Books

This section lists the UIM/X document set, and provides a suggested list for further reading.

The following list is the complete UIM/X document set:

- *UIM/X Installation Guide*. Explains how to install and run UIM/X. Includes information on the files provided with UIM/X, backwards compatibility issues, and compiler considerations.
- *UIM/X Beginner's Guide*. Introduces UIM/X by presenting Novice Mode, the simplified Palette that enables new users to be productive immediately. Includes information on a number of important features for creating, testing, and running applications.
- *UIM/X Tutorial Guide***.** A series of step-by-step tutorials, teaching tools and techniques that will greatly assist you in developing your own applications. Features tutorials in Novice Mode, Standard Mode, and on advanced topics.
- *UIM/X User's Guide*. Explores the UIM/X features common to both Motif and cross-platform development. Includes discussions of how to use UIM/ X's editors to set properties, add behavior, etc.
- *UIM/X Motif Developer's Guide*. An in-depth guide to the widgets, features and capabilities of UIM/X as they relate specifically to Motif development.
- *UIM/X Advanced Topics***.** Describes how to customize UIM/X, including integrating new widget and component classes into the executable. Includes reference information of an advanced technical nature.
- *UIM/X Reference Manual*. A comprehensive list of properties, methods, and events, plus more, for Motif development. Designed for the experienced developer.

Suggested Reading

For more information on designing GUIs, see any of the following books:

- *OSF/Motif Style Guide release 1.2* (Prentice Hall, 1993, ISBN 0-13-643123-2)
- *Visual Design with OSF/Motif* (by Shiz Kobara, Addison-Wesley, 1991, ISBN 0-201-56320-7)
- *New Windows Interface: An Application Guide* (Microsoft Corporation, 1994, ISBN 1-55615-679-0)
- *Human Interface Guidelines: The Apple Desktop Interface* (Addison-Wesley, 1987, ISBN 0-201-17753-6)
- *Indigo Magic User Interface Guidelines* (007-2167-002, 1/96)

How this Guide Is Organized

- *Chapter 1, “Magic Integration,”* covers the basic concepts of starting and running UIM/X on your Magic desktop.
- *Chapter 2, “SGI Widgets,”* describes the SGI Widgets category of the Palette.

Some Terms You Should Know

Certain basic terms recur throughout this guide, and it helps to understand them from the outset.

An *object* is a building block you can use to build an interface with UIM/X.

A *Motif widget* is an object whose appearance and behavior precisely follows the *OSF/Motif Style Guide*. The novice mode of UIM/X supports a number of popular Motif widgets, including Push Button, Label, Text Field, and more.

A compound object consists of several Motif widgets combined into one object for your convenience. The novice mode of UIM/X supports a number of compound objects, including Application Window and Group Box, that save you the time you might otherwise spend creating them.

An *interface* is a window or dialog box that you build up from objects with UIM/X. The novice mode of UIM/X supports four different types of interfaces: Application Window, Secondary Window, Message dialog box, and File Selection dialog box. Certain menu options refer to an interface, such as Save Interface; these act only on your selected interface.

A *project* contains all the interfaces (i.e., windows and dialog boxes) and their associated files for a certain GUI you are building with UIM/X. The program can automatically save and generate code for an entire project in one step. Certain menu options refer to a project, such as Save Project; these act on all the windows and dialog boxes in your project.

Conventions Used in this Guide

Typographic Conventions

The following table describes the typographic conventions used in this guide.

Typeface or Symbol	Meaning	Example
`AaBbCc12`	The names of commands, files, and directories; or onscreen output; or user input.	Edit your `.login` file. `%You have mail.` Use `ls -a` to list all the files.
AaBbCc12	A placeholder you replace with your actual value; or words to be emphasized; or book titles.	To delete a file, type `rm` *filename.* You *must* be `root` to do this. See Chapter 6 in the *User's Guide.*
File⇒Open	The Open option in the File menu.	Choose the File⇒Open command.
Alt+F4	Press both Alt and F4 at once.	Press Alt+F4 to exit.
Return	The key on your keyboard marked Enter, Return, or .	Press Return.

Installation Directories

Product installation directories can depend on the platform or the user's preferences. To keep things simple, this guide uses general names for product installation directories. The following table lists the name and the corresponding product installation directory:

Name	Description
uimx_directory	The UIM/X installation directory

Using the Mouse

Before starting the tutorial, take a moment to review the location and usage of your mouse buttons, as illustrated in the Figure P-1and the following table:

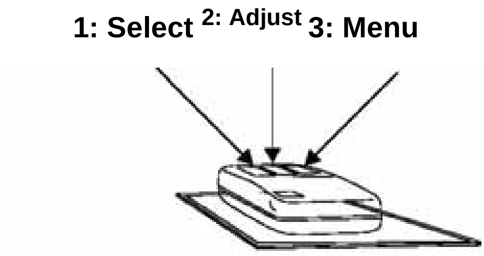


Figure P-1 The Mouse Buttons

Button:	Called:	Is used for:
1	Select	Selecting objects, menus, toggles, and options.
2	Adjust	Resizing and moving objects.
3	Menu	Displaying popup menus.

Throughout this book, you will use the mouse buttons along with the mouse pointer to make selections, move the input pointer, or position the text insertion point. You can perform any of the following mouse operations.

Operation	Description
Point to	Move the mouse to make the pointer go as directed.
Press	Hold down a mouse button.
Release	Release a mouse button after pressing it.
Click	Quickly press and release a mouse button without moving the mouse.
Drag	Move the mouse while pressing a mouse button.
Double-click	Click a mouse button twice in rapid succession without moving the mouse pointer.
Triple-click	Click a mouse button three times in rapid succession without moving the mouse pointer.

In general, instructions for mouse operations include the name of the mouse button. The exceptions are Click, Double-click, and Drag. These common operations may be described without specifying a mouse button. For example:

- Click on the `applWindow1` icon in the Interfaces Area of the Project Window.
- Drag the Push Button icon from the Palette.

In these cases, use the Select button to click and double-click, and the Adjust button to drag.

Magic Integration

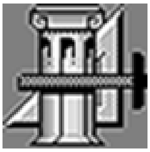
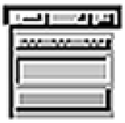
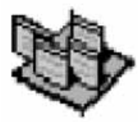
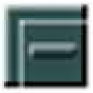
1

Overview

This chapter covers the basic concepts of starting and running UIM/X on your Magic desktop.

Running UIM/X with Magic

UIM/X is fully integrated with the Magic Desktop. You can start UIM/X and load files by working with desktop icons. The following table describes the different UIM/X icons.

Desktop Icon	Description
	UIM/X icon. To start UIM/X, double-click on this icon. You can also start UIM/X and load a file by dropping the file icon on this icon. Pressing <Alt> while double-clicking on the icon causes a launch dialog to be displayed. The dialog's text field displays the command line for starting UIM/X. You can then modify the command-line options in the text field before starting UIM/X. You can select language mode before starting UIM/X by clicking on the icon, displaying its pop-up menu using the Menu mouse button, and selecting the appropriate menu option: ANSI C, K&R C, or C++. After you start UIM/X, you can load files by dropping their icons on either the Interfaces Area or the Palettes Area of the Project Window.
	UIM/X interface file (`.i`). To start UIM/X and load the interface, double-click on this icon. You can also drop it on the UIM/X icon to start UIM/X and load the interface. To load the interface into UIM/X, drop the icon in the Interfaces Area or Palettes Area of the Project Window.
	UIM/X project file (`.prj`). To start UIM/X and load the project, double-click on the icon. You can also drop it on the UIM/X icon to start UIM/X and load the project. To load the project into UIM/X, drop the icon in the Interfaces Area or Palettes Area of the Project Window.
	UIM/X palette file (`.pal`). To start UIM/X and load the palette, double-click on the icon. You can also drop it on the UIM/X icon to start UIM/X and load the palette. To load the palette into UIM/X, drop the icon in the Palettes Area or the Interfaces Area of the Project Window.

SGI Widgets

2

Overview

This chapter describes the SGI Widgets category of the Palette.

The Default Palette

The Ux Palette, shown in Figure 2-1, is the default palette displayed when you start UIM/X. Its file name is `Uxsgi.pal` and it is found in *uimx_directory*`/palettes.` This palette contains a new category called SGI Widgets, which is described in detail later in this chapter.

You can change the default palette that UIM/X displays on start-up by using resources. The UIM/X resource file contains the following resource specification:

```
Uimx3_0*UxStartingPalettes.value: Uxsgi.pal
```

The `UxStartingPalettes` resource specifies the palette files loaded at start-up. These palette files are loaded before any files specified on the UIM/X command line. These palettes are not saved with projects saved by the user.

If you wish to have UIM/X start with the Ux Palette, but with no SGI Widgets category, change the `UxStartingPalettes` resource to look as follows:

```
Uimx3_0*UxStartingPalettes.value: Ux.pal
```

Note: You can load more than one palette file at once. To load more than one palette, you must insert \n between palette file names. For more information about working with palettes, see the *UIM/X User's Guide*.

The Palette is shown below. Some categories have been collapsed for display purposes:

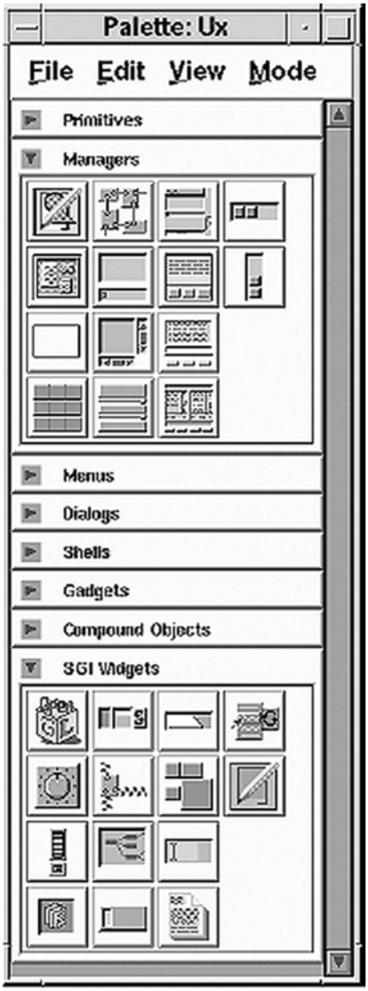
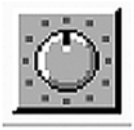


Figure 2-1 The Palette, Showing the SGI Widgets Category

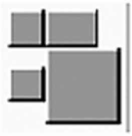
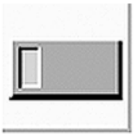
SGI Widgets

The SGI Widgets provided by the Ux Palette are shown in the following table, along with their names, suggested uses, and how the end user activates them in your interface. While these SGI Widgets are not official Motif widgets, they still follow the *OSF/Motif Style Guide*.

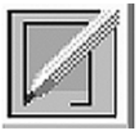
Name	Icon	Suggested Uses
dial		Purpose: to allow users to input or modify a value from within a range of values.You can specify whether the dial's input control takes the form of a knob or a pointer. Use a dial to specify an analog value within a given range. To activate a dial, drag its input control (knob or pointer) clockwise or counterclockwise.
dropPocket		Purpose: to receive and display Desktop icons. When users drop Desktop file icons onto the dropPocket, the dropPocket determines the name of the icon and returns information describing the icon to the application in the callback. When users drag an acceptable icon over the dropPocket, the dropPocket background changes color and the dropPocket displays the dropped icon. If the type of a file is not known, or if the file doesn't exist, the dropPocket displays the icon for unknown file types. Use a dropPocket as a quick and easy means to furnish your application with icon information. Activate a dropPocket by dragging and dropping a Desktop icon into it.
dynaMenu		Purpose: a packaged pop-up menu that supports dynamic changing of menu entries. Use a dynaMenu as an easy means to create a history menu. Activate a dynaMenu by clicking on it with the Select mouse button, then selecting the menu item of your choice from the list.

Name	Icon	Suggested Uses
finder		Purpose: a customizable widget for various applications, including directory navigation and accelerating text selection of long objects such as file names. A finder comprises a text field, a drop pocket, a navigation bar, and a recycle button. The navigation bar contains buttons representing each directory in the path name. When the user clicks on a path bar button, the finder sets the current directory to the directory listed underneath that button. Use a finder to accelerate text selection and to navigate through file systems. Activate a finder by dropping a Desktop file icon into the drop pocket to find the path name for the file, or by dragging an icon out of the drop pocket and putting it on the Desktop. Select a button on the navigation bar to change the current directory. Click the recycle button to display a list of directories that the user has selected during the current finder session. Select an item from the recycle list to change the current directory to the selected directory.

Name	Icon	Suggested Uses
glwMdrawArea		Purpose: provides a window with the appropriate visual and colormaps needed for open GL. Use a glwMdrawArea to integrate open GL applications into Motif interfaces. A glwMdrawArea cannot be activated.
graph		Purpose: to allow the user to display any group of widgets as a graph, with each widget representing a node. Use a graph to arrange all nodes either horizontally or vertically according to an internal layout algorithm. Activate a graph by clicking on a node or an arc to select it. Use area select to select multiple nodes and arcs.

Name	Icon	Suggested Uses
grid		Purpose: a container widget that arranges its children in a two-dimensional grid of arbitrary size. You can separately designate each row and column of the grid as having a fixed size or as having some degree of stretchability. You can also resize each child in either or both directions, or force a child to a fixed size. Use a grid to group children (for example, pushButtons) and determine their resize capability. The ResizeVertical and ResizeHorizontal resources determine whether the Grid can resize the child to fill the cell in the vertical and horizontal directions. A grid cannot be activated.
ledButton		Purpose: to activate an operation and display toggle status. Use a ledButton to activate functions on your interface. A ledButton can contain either a text label or an icon to indicate its purpose. Activate a ledButton by clicking it.
progressIndicator		Purpose: a scale widget with a thermometer visual to show progress in a Working dialog. When SlidingMode is set to `slider`, it is an ordinary scale. Use a progressIndicator to allow users to monitor percentage of completion of a task. A progressIndicator cannot be activated when using the thermometer visual.

Name	Icon	Suggested Uses
sgText		Purpose: to provide a single-line or multi-line text editor. The sgText is an advanced version of the Text widget. Use an sgText to accept typed input from the end user. You can choose the background and foreground colors for selected text, as well as the background color for text selected with an error status. Activate an sgText by clicking within it and then typing.
sgTextField		Purpose: to accept, display, or edit one line of text. The sgTextField is an advanced version of the TextField widget. Use an sgTextField to accept one line of typed input from the end user or to display one line of read-only text. You can choose the background and foreground colors for selected text, as well as the background color for text selected with an error status. Activate an sgTextField by clicking in it and then typing, or double-clicking to select all the existing text and then typing to replace it.

Name	Icon	Suggested Uses
springBox		Purpose: to provide a container that arranges its children in a single row or column based on a set of spring constraints assigned to each child. Use a springBox to specify the “springiness” for both the widget’s size and position relative to its siblings.You control the springiness of a widget’s size by setting the values of its VerticalSpring and HorizontalSpring resources. A value of zero means the child cannot be resized in that direction. A springBox cannot be activated.
thumbWheel		Purpose: to allow users to input or modify an analog value. Use a thumbWheel to input or modify a value, either from within a range of values or from an unbounded (infinite) range. Activate a thumbWheel by dragging its wheel in either direction. Click the home button, located outside the wheel region to reset the wheel’s value to a known position.
visualDrawingArea		Purpose: a general layout widget that may have many children and imposes no constraints on their layout. Use a visualDrawingArea to display graphics. A visualDrawingArea cannot be activated.

Swidget Methods

The SGI Widgets each have a set of swidget methods. These swidget methods allow the developer to use connections to specify behavior for the SGI Widgets. The swidget methods for each widget are listed in the following table:

SGI Widget	Swidget Methods
dial	GetDialValue Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground SetDialValue Unmanage
dropPocket	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground Unmanage
dynaMenu	DynaMenuAddHistoryItem DynaMenuClearHistory Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground Unmanage UpdateDisplay

SGI Widget	Swidget Methods
finder	FinderAddHistoryItem FinderClearHistory GetFinderTextString Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetFinderTextString SetForeground Unmanage UpdateDisplay
glwMdrawArea	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground Unmanage
graph	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground Unmanage UpdateDisplay
grid	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetDialogTitle SetForeground Unmanage UpdateDisplay

SGI Widget	Swidget Methods
ledButton	GetToggleState Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground SetInsensitivePixmap SetLabelPixmap SetLabelString SetToggleState Unmanage
progressIndicator	GetValue Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground SetTitleString SetValue Unmanage UpdateDisplay
sgText	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground Unmanage
sgTextField	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground Unmanage

SGI Widget	Swidget Methods
springBox	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetDialogTitle SetForeground Unmanage UpdateDisplay
thumbWheel	GetValue Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground SetValue Unmanage
visualDrawingArea	Insensitive Manage Sensitive SetBackground SetBackgroundPixmap SetForeground Unmanage UpdateDisplay

Note: The following methods are inherited from the widget's parent: Insensitive, Manage, Sensitive, SetBackground, SetBackgroundPixmap, SetForeground, Unmanage, UpdateDisplay.

Index

Index

Index