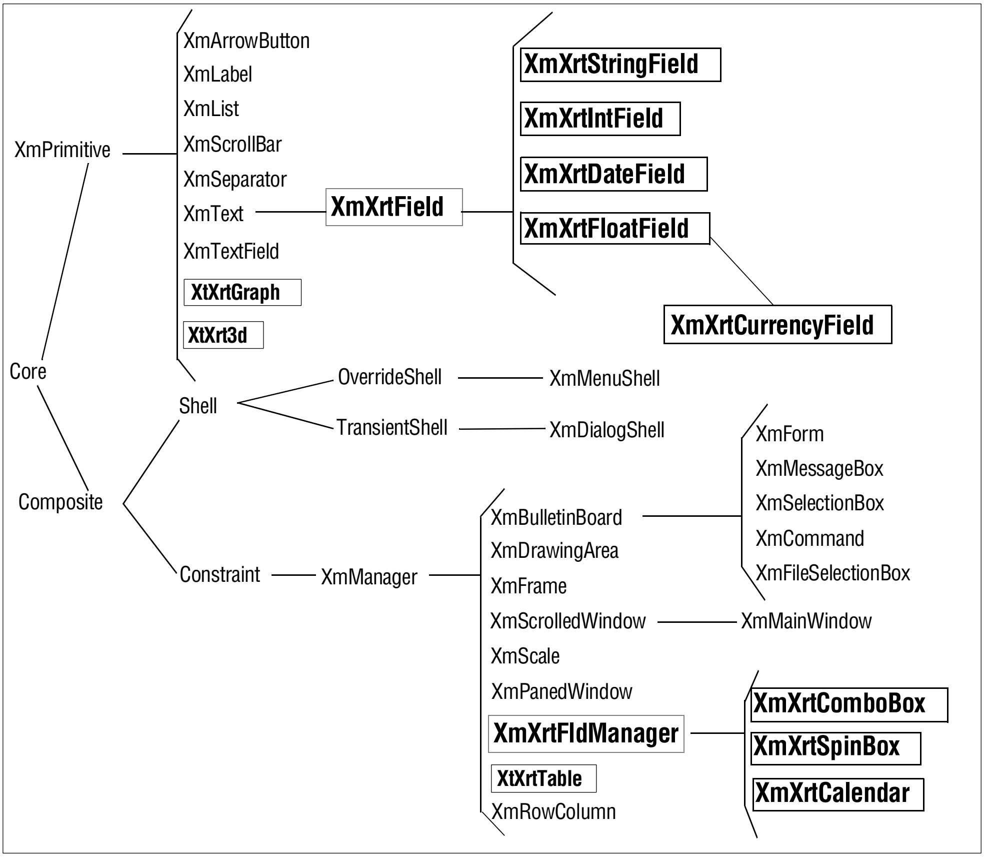


Figure 12   XRT/field widgets' position within the Motif Class Hierarchy

The Xt Intrinsics defines an object-oriented architecture that organizes widgets into classes. For example, the Motif XmForm widget is a subclass of XmBulletinBoard, which is a subclass of XmManager, which is subclassed from the Constraint, Composite and Core widgets.

The XrtField widget class is defined as a subclass of XmText, which is in turn subclassed from XmPrimitive. This means that XmText, XmPrimitive and Core resources can be set in addition to the resources defined by the XrtField widget. For example, you can set the default background color with the `XmNbackground` resource.

XrtComboBox, XrtSpinBox and XrtCalendar are subclassed from XmXrtFldManager.

Replacing XmText Widgets

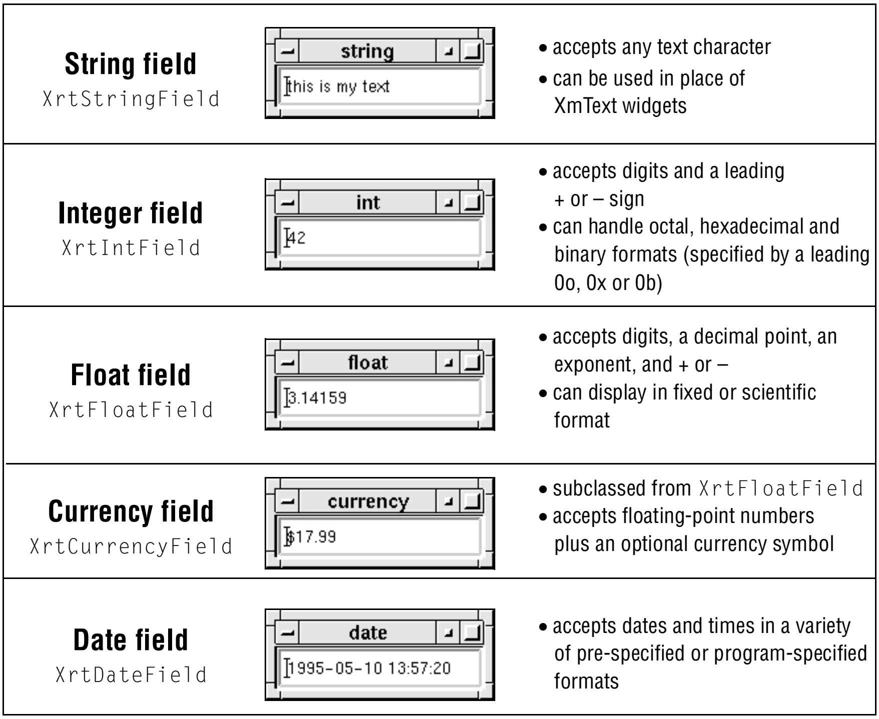
As a subclass of XmText, any XrtField widget can replace any XmText widget in an application.

Replacing XmTextField Widgets

You should be able to replace XmTextField widgets with XrtField widgets without changing the application's behavior, since XmTextField has the same resources as XmText.

2.2 The XrtField Subclasses

XrtField is an abstract class that defines several subclasses. You cannot instantiate the XrtField class. Each of the subclasses (shown in Figure 13) handles a certain type of user input.



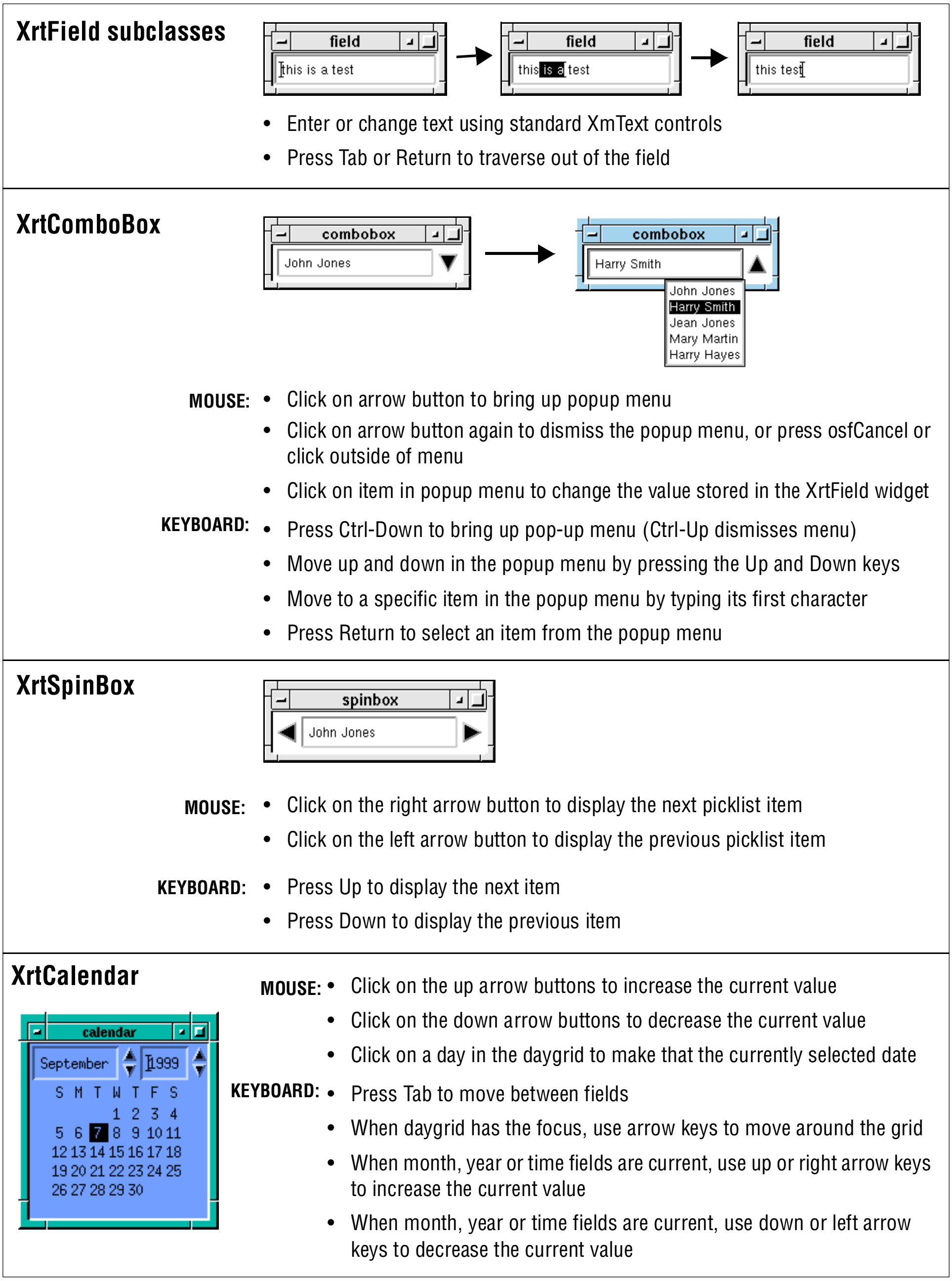
String field `XrtStringField`	• accepts any text character • can be used in place of XmText widgets
Integer field `XrtIntField`	• accepts digits and a leading + or – sign • can handle octal, hexadecimal and binary formats (specified by a leading 0o, 0x or 0b)
Float field `XrtFloatField`	• accepts digits, a decimal point, an exponent, and + or – • can display in fixed or scientific format
Currency field `XrtCurrencyField`	• subclassed from `XrtFloatField` • accepts floating-point numbers plus an optional currency symbol
Date field `XrtDateField`	• accepts dates and times in a variety of pre-specified or program-specified formats

Figure 13 The subclasses of the XrtField widget

In this document, the term *field* refers to a widget that belongs to one of the subclasses of XrtField.

2.3 Default User Interaction

Figure 14 illustrates the most common ways users interact with XRT/field widgets. For a complete list of user interactions, refer to the translation tables in Appendix D on page 141.



XrtField subclasses

- Enter or change text using standard XmText controls
- Press Tab or Return to traverse out of the field

XrtComboBox

MOUSE:

- Click on arrow button to bring up popup menu
- Click on arrow button again to dismiss the popup menu, or press osfCancel or click outside of menu
- Click on item in popup menu to change the value stored in the XrtField widget

KEYBOARD:

- Press Ctrl-Down to bring up pop-up menu (Ctrl-Up dismisses menu)
- Move up and down in the popup menu by pressing the Up and Down keys
- Move to a specific item in the popup menu by typing its first character
- Press Return to select an item from the popup menu

XrtSpinBox

MOUSE:

- Click on the right arrow button to display the next picklist item
- Click on the left arrow button to display the previous picklist item

KEYBOARD:

- Press Up to display the next item
- Press Down to display the previous item

XrtCalendar

MOUSE:

- Click on the up arrow buttons to increase the current value
- Click on the down arrow buttons to decrease the current value
- Click on a day in the daygrid to make that the currently selected date

KEYBOARD:

- Press Tab to move between fields
- When daygrid has the focus, use arrow keys to move around the grid
- When month, year or time fields are current, use up or right arrow keys to increase the current value
- When month, year or time fields are current, use down or left arrow keys to decrease the current value

Figure 14 XRT/field User Interactions

2.4 Storing Values in Fields

To store a value in a field, you can set either of two resources: XmNxrtFldInternalValue or XmNvalue, though for optimal XRT functionality, it is best to use XmNxrtFldInternalValue. The exception is when you want to set visual cues for a field (for example, have *dd/mm/yy* display in the field so the user knows what can be entered where). In this case, you need to use XmNvalue.

Using XmNxrtFldInternalValue

The XmNxrtFldInternalValue resource stores the value using the data type appropriate for the widget subclass.

Field Subclass	Internal Value Type
XrtStringField	string
XrtIntField	long
XrtFloatField	double
XrtCurrencyField	double
XrtDateField	struct tm

For example, the following creates an integer field and uses XmNxrtFldInternalValue to set its value:

```
field = XtVaCreateManagedWidget("field",
    xmXrtIntFieldWidgetClass,   toplevel,
    XmNxrtFldInternalValue,     42,
    NULL);
```

When the user exits a field, the value typed by the user is copied from XmNvalue to XmNxrtFldInternalValue (with type conversion performed when necessary).

XmNxrtFldInternalValueLong

If the widget is a date field, the XmNxrtFldInternalValueLong resource contains, if possible, the value of XmNxrtFldInternalValue represented as an integer of type *long*. (This is only possible if the date is in a range roughly between the years 1902 and 2033.)

Using XmNvalue

XmNvalue contains the actual text string stored and displayed in the field. The following uses XmNvalue to set the initial value of an integer field:

```
field = XtVaCreateManagedWidget("field",
    xmXrtIntFieldWidgetClass,   toplevel,
    XmNvalue,                   "42",
    NULL);
```

If XmNvalue is set to a value inappropriate for the widget's subclass (such as "abc" for an integer field), the value is displayed but is flagged as invalid.

Note: The only time `XmNvalue` needs to be used is in the context of setting visual cues in a masked field. See "Specifying Visual Cues" on page 45.

2.5 Edit and Display Formats

XRT/field allows you to specify two formats that control the appearance of the value of a field:

- The *edit format*, which controls the appearance of a value while it is being edited.
- The *display format*, which controls the appearance of a value while it is not being edited.

For details on edit formats, see section 3.2 on page 36. For information on display formats, see section 3.8 on page 53.

2.6 Conversion, Completion and Validation

XRT/field allows you to manipulate the field's value during the editing process in three ways:

- You can *convert* the value from one form to another. For example, the value can be converted from lower-case to upper-case.
- You can *complete* the value by adding a prefix or suffix to it. For example, you can add a *.c* suffix to a file name.
- You can *validate* the value.

These operations can be performed at various points in the field editing process. For details, see the following section.

2.7 The Editing Process

The field editing process can be broken down into the following steps:

- The user first enters the widget (gives the field input focus): this is called an *enter field action.*
- The user types one or more keystrokes. Each keystroke typed is called a *keystroke action.*
- The user performs a paste or drag and drop operation: this is called a *paste action.*
- Finally, the user exits the widget by pressing Return or Tab, or by moving the mouse pointer out of the widget: this is called a *user exit action.*

The application can itself modify the value stored in the field. A modification not initiated by a user is known as a *programmatic change action.*

XRT/field allows you to add processing after any of these actions. The sequence of interactions and possible processing is shown in Figure 15.

2.7.1 Enter Field Action Processing

When an enter field action occurs, you can provide the following:

- A list of callback routines to be invoked following an enter field action. For details on how to specify enter callbacks, see section 4.2 on page 61.
- An edit format that determines how the value is to be displayed while being edited. For more information on edit format specification, refer to section 3.2 on page 36.

2.7.2 Keystroke Action Processing

When a keystroke action occurs, you can do the following:

- Convert the (partially typed) value from one format to another. For details on the conversion capabilities of XRT/field, refer to section 3.6 on page 51.
- Determine whether the typed key is valid. For information on validation, refer to section 3.4 on page 40.

2.7.3 Paste Action Processing

When a paste action occurs, you can do the following:

- Convert the pasted value from one format to another. For details on the conversion capabilities of XRT/field, refer to section 3.6 on page 51.
- Determine whether the pasted value is valid. For information on validation, refer to section 3.4 on page 40.