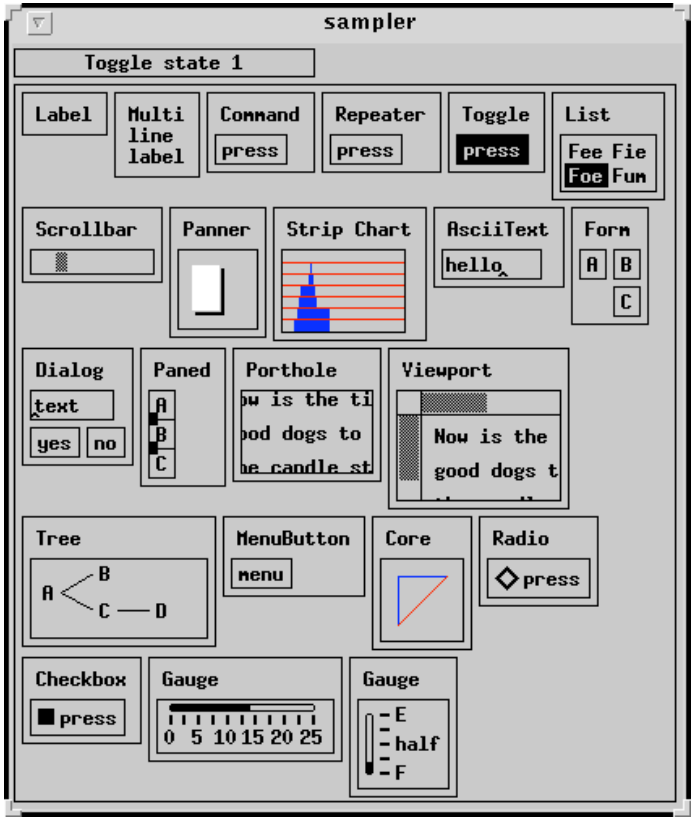
Widget Sets

- Collection of user interface components
- Together with WM, define look&feel of system
- Several different ones available for X
  - Athena (original, simple widget set, ca. 20 widgets, 2-D, no strong associated style guide) — Xaw... prefix
  - Motif (Open Software Foundation, commercial, 2.5-D widget set, >40 widgets, industry standard for X, comes with style guide and UIL)—Xm... prefix
- Programming model already given in Intrinsics
  - Motif just offers convenience functions

Athena Widget Set

- Original, free, extensible
- Ugly, simple
  - Simple -- Base class for all other Athena widgets. Does nothing, but adds new resources such as cursor and border pixmap.
  - Label -- Draws text and/or a bitmap.
  - Command -- Momentary push-button
  - Toggle -- Push-button with two states.



Athena

- Repeater -- Command that repeatedly calls its associated callback function for as long as it's held.
- MenuButton -- Push-button that brings up a menu.
- Grip -- Small widget used to adjust borders in a Paned widget.
- List -- Widget to allow user to select one string from a list.
- Scrollbar -- Widget to allow user to set a value; typically to scroll another widget.
- Panner -- Widget to allow user to scroll in two dimensions. See the editres program for a good example.
- StripChart -- Widget to display a scrolling graph.
- Box -- Composite widget which simply lays children out left-to-right.
- Form -- Constraint widget which positions children relative to each other.

Athena

- Dialog -- Form widget for dialog boxes.
- Paned -- Constraint widget which allows user to adjust borders between child widgets.
- Porthole -- Composite widget which allows a larger widget to be windowed within a smaller window. Often controlled by Panners.
- Viewport -- Constraint widget which acts like a Porthole with scrollbars.
- Tree -- Constraint widget which lays its children out in a tree.
- SimpleMenu -- Shell which manages a simple menu.
- Sme -- RectObj which contains a simple menu entry (blank).
- SmeBSB -- Menu entry with a string and optional left & right bitmaps.
- SmeLine -- Menu entry that draws a seperator line.

Athena

- Text -- Base class for all other text classes.
- TextSink -- Base class for other text sinks.
- TextSrc -- Base class for other text sources.
- AsciiText -- Text widget which handles 8-bit or 16-bit text.
- AsciiSink -- Helper object which displays 8-bit text.
- AsciiSrc -- Helper object which manages text in memory or file.
- MultiSink -- Same as AsciiSink, but handles 16-bit text.
- MultiSrc -- Same as AsciiSrc, but handles 16-bit text.

What Is Motif?

- Style Guide (book) for application developer
- Widget set (software library) implementing Style Guide
- Window Manager (mwm)
- UIL (User Interface Language)

The Motif Widget Set

- Simple Widgets: XmPrimitive
  - XmLabel, XmText, XmSeparator, XmScrollbar,...
- Shell Widgets: Shell
  - Widgets talking to Window Manager (root window children)
  - Application shells, popup shells,...
- Constraint Widgets: XmManager
  - Containters like XmDrawingArea, XmRowColumn,...
  - Complex widgets like XmFileSelectionBox,...

Programming with Motif

- Initialize Intrinsics
  - Connect to server, allocate toolkit resources
- Create widgets
  - Building the dynamic widget tree for application
  - Tell Intrinsics to manage each widget
- Realize widgets
  - Sensitize for input, per default also make visible (map)
- Register callbacks
  - Specify what app function to call when widgets are triggered
- Event loop
  - Just call Intrinsics (XtMainLoop()) – app ends in some callback!

hello.c: A Simple Example

```
#include <X11/Intrinsic.h>
#include <X11/StringDefs.h>
#include <X11/Xlib.h>
#include <Xm/Xm.h>
#include <Xm/PushB.h>

void ExitCB (Widget w, caddr_t client_data, XmAnyCallbackStruct
*call_data)
{
  XtCloseDisplay (XtDisplay (w));
  exit (0);
}

void main(int argc, char *argv[])
{
  Widget toplevel, pushbutton;

  toplevel = XtInitialize (argv [0], "Hello", NULL, 0, &argc, argv);
  pushbutton = XmCreatePushButton (toplevel, "pushbutton", NULL, 0);
  XtManageChild (pushbutton);

  XtAddCallback (pushbutton, XmNactivateCallback, (void *) ExitCB,
NULL);

  XtRealizeWidget (toplevel);
  XtMainLoop ();
}
```


Resource files in X

- Where does the title for the PushButton come from?
- → Resource file specifies settings for application
- Syntax: Application.PathToWidget.Attribute: Value
- Resource Manager reads and merges several resource files (system-, app- and user-specific) at startup (with priorities as discussed in reference model)

File "Hello":

```
Hello.pushbutton.labelString:   Hello World
Hello.pushbutton.width:         100
Hello.pushbutton.height:        20
```

User Interface Language UIL

- Resource files specify late refinement of widget attributes, but cannot add widgets
- Idea: specify actual widget tree of an application outside C source code, in UIL text file
  - C source code only contains application-specific callbacks, and simple stub for user interface
  - UIL text file is translated with separate compiler
  - At runtime, Motif Resouce Manager reads compiled UIL file to construct dynamic widget tree for app
- Advantage: UI clearly separated from app code Decouples development

X/Motif: Evaluation

- Availability: high (server portability), standard WS for Unix
- Productivity: low for Xlib-based and widget development, but high using widget set, esp. Motif
- Parallelism: external yes, internal no - in original design, one app can freeze server with big request
- Performance: fairly high (basic graphics were faster than Windows on same hardware), widget sets add graphical and layout overhead, but can hold client-side resources

X/Motif: Evaluation

- Graphics model: RasterOp
- Style: exchangeable through widget set and WM
  - Note: apps cannot rely on a certain WM functionality
- Extensibility: low
  - Requires modifying Xlib source, usually also Xt and widget set source, applications using extension not backwards compatible and portable anymore
- Adaptability: very high (multiple resource files, UIL)
- Resource sharing: possible via RIDs
- Distribution: yes, BWS, WM & apps on different machines

X/Motif: Evaluation

- API structure: Xlib procedural, Xt/widget set OO
  - Graphics apps need to use both APIs!
- API comfort: high with Motif (even UIDS available)
- Independence: low with Xlib (visuals), high with Motif
- Communicating apps: via RIDs in server for resources, clipboard for text & graphics