EDM

Extensible Display Manager for EPICS

99%: John Sinclair, June 25, 2001

Updated: Kay Kasemir, April 2002

Outline

- EDM Introduction
- Execution
- Creating content
- Editing content
- Object Information
- Process Variables
- Details: color, macros, symbols
- Hands-on exercises

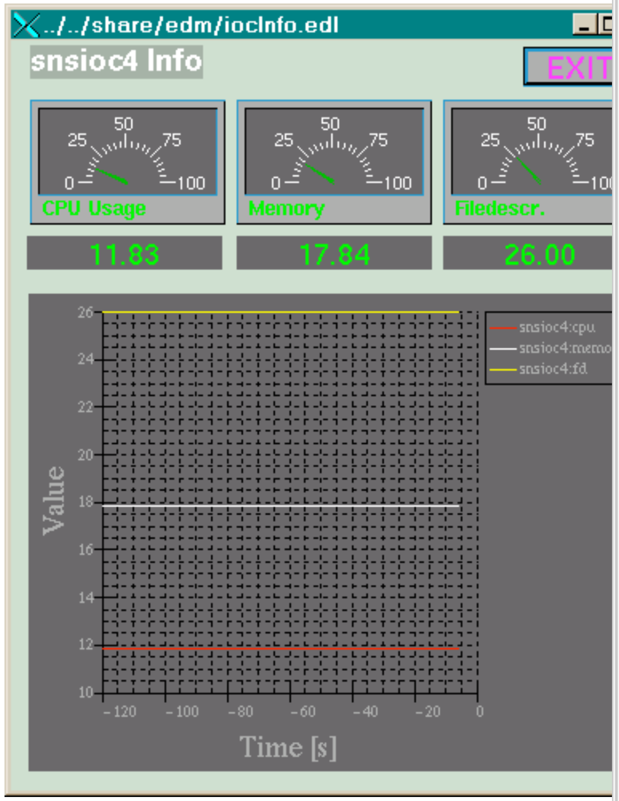
Introduction

- EDM is an interactive GUI builder and execution engine, EPICS documentation uses the term *Display Manager*
- Maintained by ORNL EPICS community
- Component based, thus extensible by other members of the EPICS collaboration

Extensible defined as:

- All “objects” are loaded from shared libraries
- EDM administrator can add & remove objects from the list of available objects without recompiling EDM itself
- Objects are versioned; carefully coded objects can be upgraded without impacting existing displays

Example EDM Operator Screens


../../share/edm/iocInfo.edl

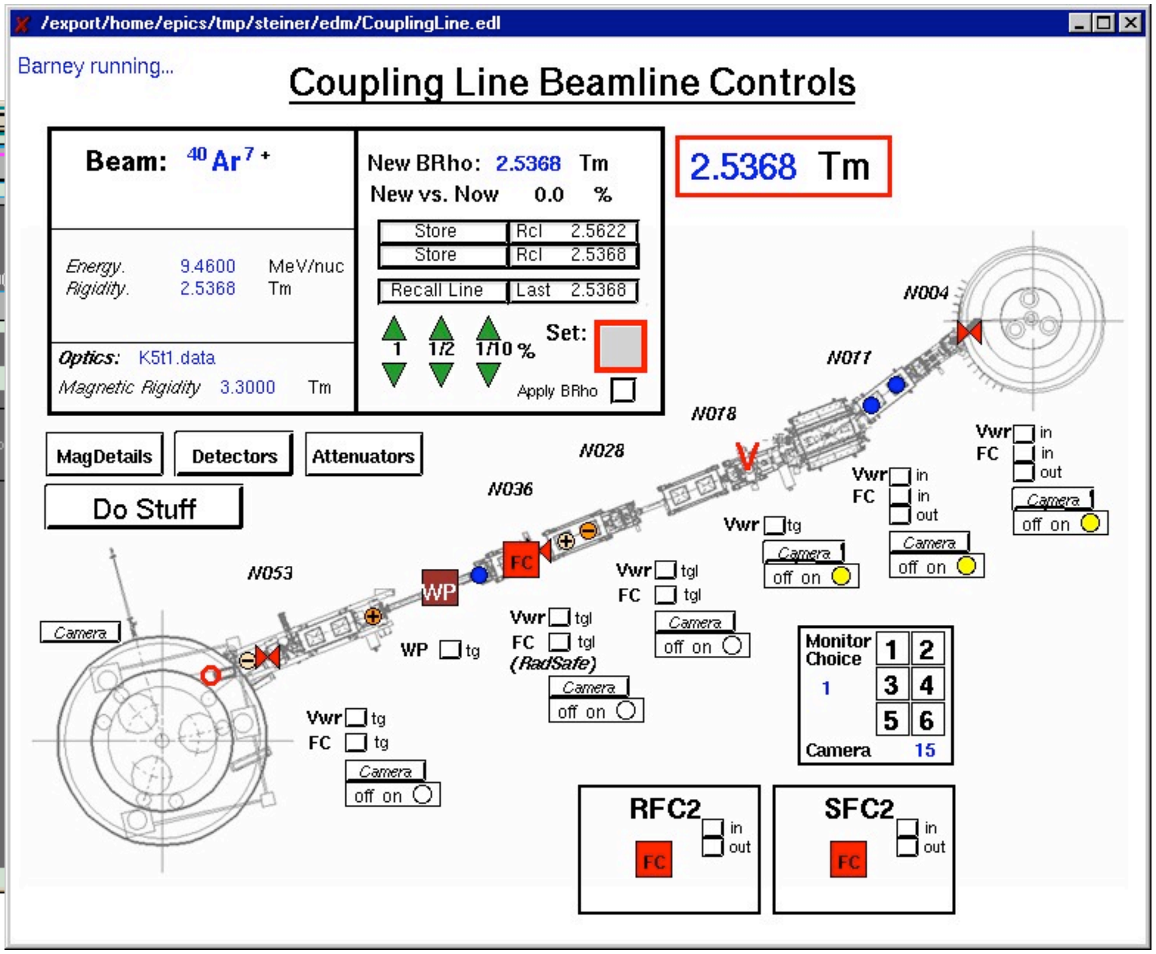
snsioc4 Info

CPU Usage 11.83

Memory 17.84

Filedescr. 26.00

(SNS Linac test)


/export/home/epics/tmp/steiner/edm/CouplingLine.edl

Barney running...

Coupling Line Beamline Controls

Beam: $^{40}Ar^{7+}$

Energy. 9.4600 MeV/nuc

Rigidity. 2.5368 Tm

Optics: K5t1.data

Magnetic Rigidity 3.3000 Tm

New BRho: 2.5368 Tm

New vs. Now 0.0 %

Store | Rcl 2.5622

Store | Rcl 2.5368

Recall Line | Last 2.5368

1 1/2 1/10 %

Set:

Apply BRho

2.5368 Tm

Monitor Choice 1

Camera 15

(Matthias Steiner, Nat'l Superconducting Cyclotron Lab., Michigan State University)

Program Execution

- Prerequisites (one time)
  - Obtain/create/modify colors and fonts file
  - Obtain/create/modify default display scheme
  - Set environment variables
  - Install above files in appropriate locations

Execution Program (cont)

- To create a display scheme:
  - Start edm: type “edm”
  - Create a new display: Menu File/New
  - Edit display properties (middle button menu) and set default fonts and colors
  - Save display scheme as default.scheme
  - Exit EDM
  - Install into $EDMFILES
  - See website documentation for additional capability of display scheme facility

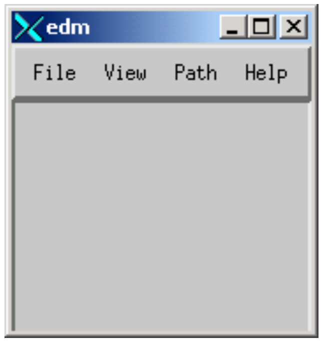
Environment Variables: Example setup

```
# Helpers
export EDMCFG=/home/T1/EDM
export EDMBIN=/cs/epics/extensions/src/edm

# Essential EDM variables
export EDMFILES=$EDMCFG
export EDMOBJECTS=$EDMCFG
export EDMPVOBJECTS=$EDMCFG
export EDMHELPFILES=/cs/epics/extensions/src/edm_cvs/helpFiles

# EDM search path:
# Local, shared data files, ...
export EDMDATAFILES=.

if [ `echo $LD_LIBRARY_PATH | grep -c $EDMBIN` -eq 0 ]
then
   export LD_LIBRARY_PATH=$EDMBIN:$LD_LIBRARY_PATH
fi
alias edm=$EDMBIN/edm
```

Program Execution - Command Line Options

- Define macro replacement
  -m “var1=value1,var2=value2,…”
  (referenced as $(var1) and $(var2))
- Execute mode
  -x (-noedit)
- Typical for operations:
  edm -x -noedit -m “var1=1,var2=2” displayFile

Main Window Operations



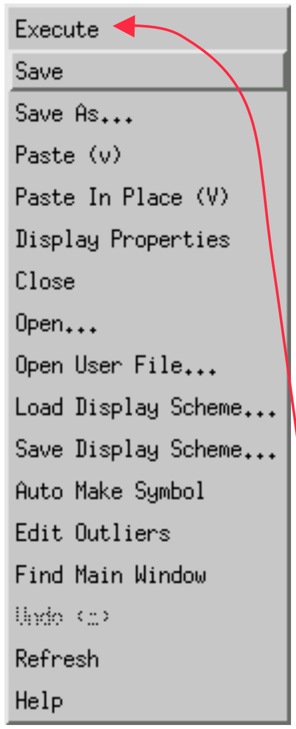
- Initially, edm opens a main window - only the menu bar is used:
  - File/New – Create new display
  - File/Open – Open existing display
  - Path – Select one of the directories listed in EDMDATAFILES variable
  - Help – explains many editing features and explains properties of most objects

For medm Users…

To save a display file:

With no *objects selected(!),* in a display screen, click the middle mouse button on the display background

This menu pops-up



Save
Save As…
Close
Open…
Open User File...

and:

Switch between edit and execute mode

File Operation Notes

- You never need to include the file extension (xxx*.edl*) in a file open or save operation
- “Save As…” to an existing file requires user confirmation
- “Save To Current Path” always requires user confirmation

Creating/Editing Display Content

- Past user observations
  - Expert friendly
  - All mouse buttons, many keys and most of the conceivable combinations of shift/ctrl/double-click are used!
  - Takes some getting-used-to, but in the end allows for *very efficient editing*.
  - If lost: Press left-double-click somewhere on the display where there is no object (to exit line-edit mode).

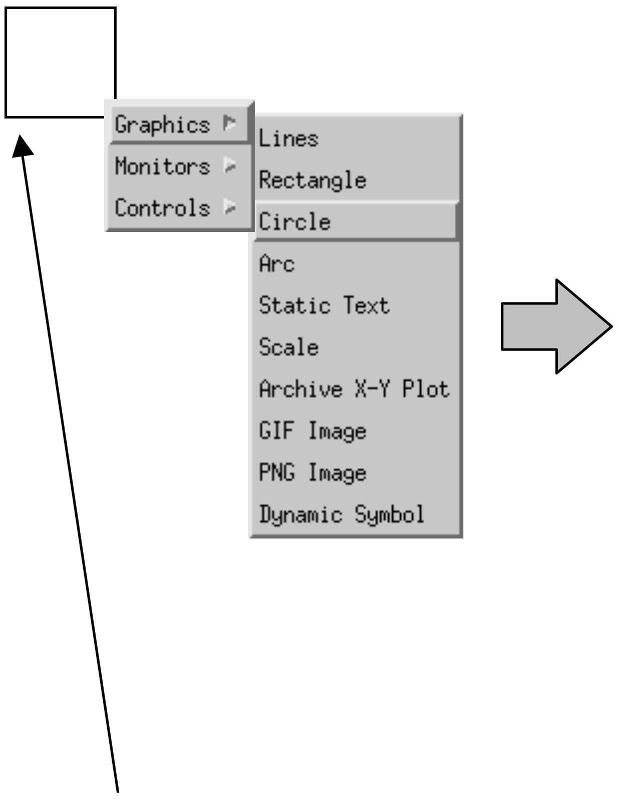
Creating/Editing Display Content Helpful Guidelines

- Use the online help
- Create your own cheat-sheet (you can put this in the form of an edm screen later)
- Line create/edit operations are somewhat complicated – read the help information before working with lines

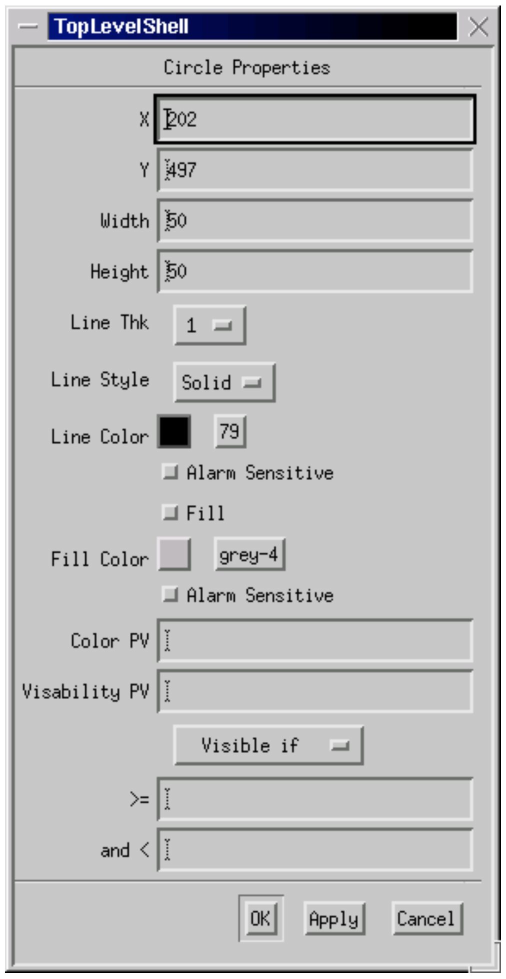
Creating/Editing Display Content (cont)

- Creating objects
- Selecting objects
- Editing object properties
- Moving/Resizing/Aligning

Creating Objects

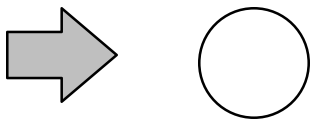


Left mouse button drag to
“rubberband” initial object size



Circle Properties

X	202
Y	497
Width	50
Height	50
Line Thk	1
Line Style	Solid
Line Color	■ 79
	☐ Alarm Sensitive
	☐ Fill
Fill Color	grey-4
	☐ Alarm Sensitive
Color PV	
Visability PV	
	Visible if
>=	
and <	

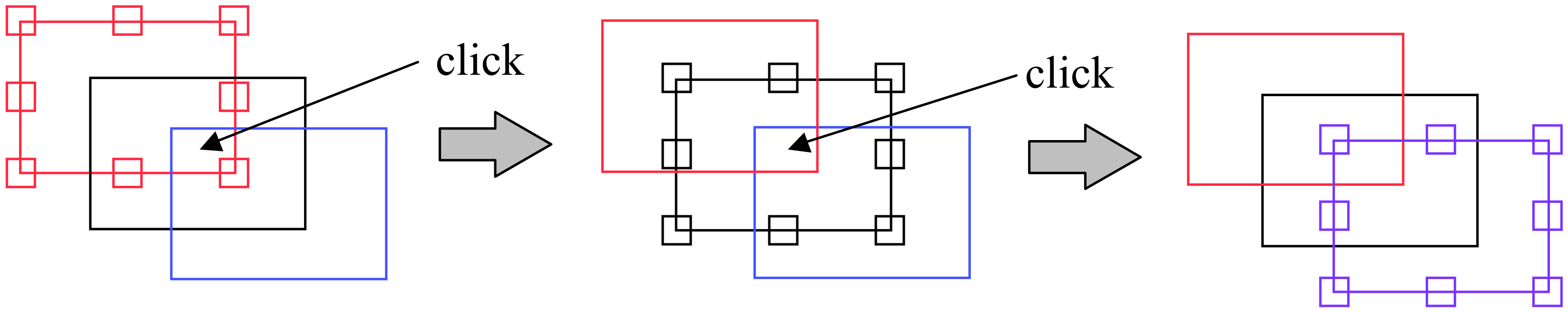


Selecting Objects

- Left button click
  - Single exclusive select: object is selected, currently selected objects are deselected
- Shift-left button click
  - Single inclusive select: object is added or removed from the current group of selected objects
- Control-left button click
  - Cycle through a stack of objects, one at a time

Selecting Objects (cont)

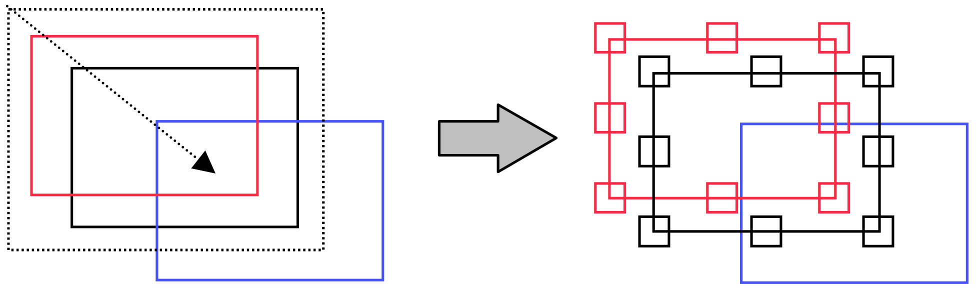
- Control-left button click requires that one and only one object be selected



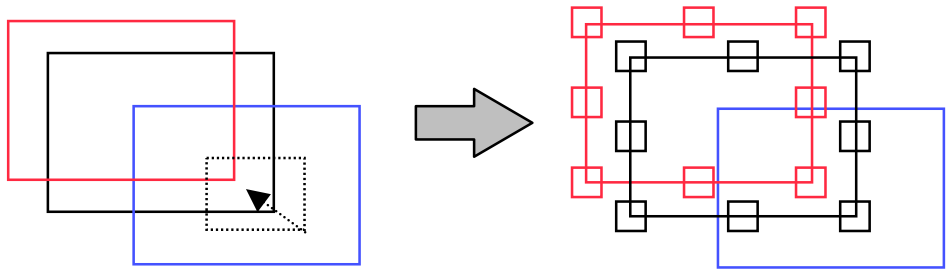
Selecting Objects (cont)

This convenient idea was adopted from AutoCad…

- Middle button drag - objects are added or removed from the current selection group



Top-left to bottom-right:
Select enclosed **objects**

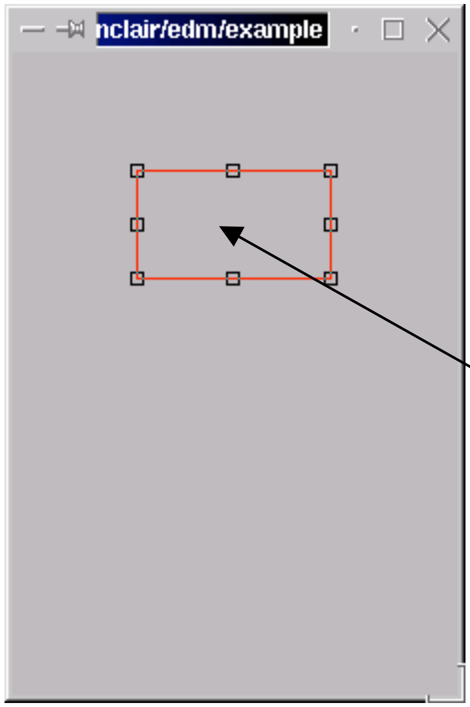


Bottom-right to top-left:
Select enclosed **corners**

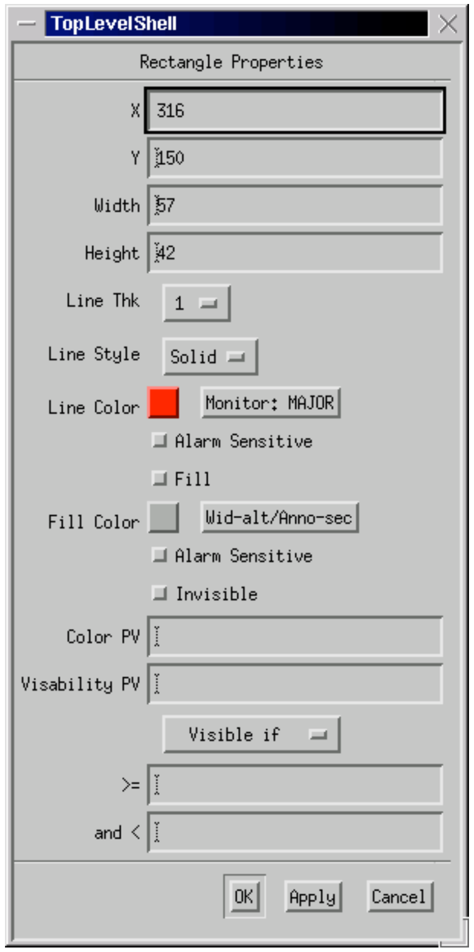
Again:

- Left button rubberband: Create new object
- Middle button rubberband: Select objects

Editing Objects: Property Dialog



left click on selected object



Rectangle Properties

X 316
Y 150
Width 57
Height 42
Line Thk 1
Line Style Solid
Line Color Monitor: MAJOR
Alarm Sensitive
Fill
Fill Color Wid-alt/Anno-sec
Alarm Sensitive
Invisible
Color PV
Visability PV
Visible if
>=
and <

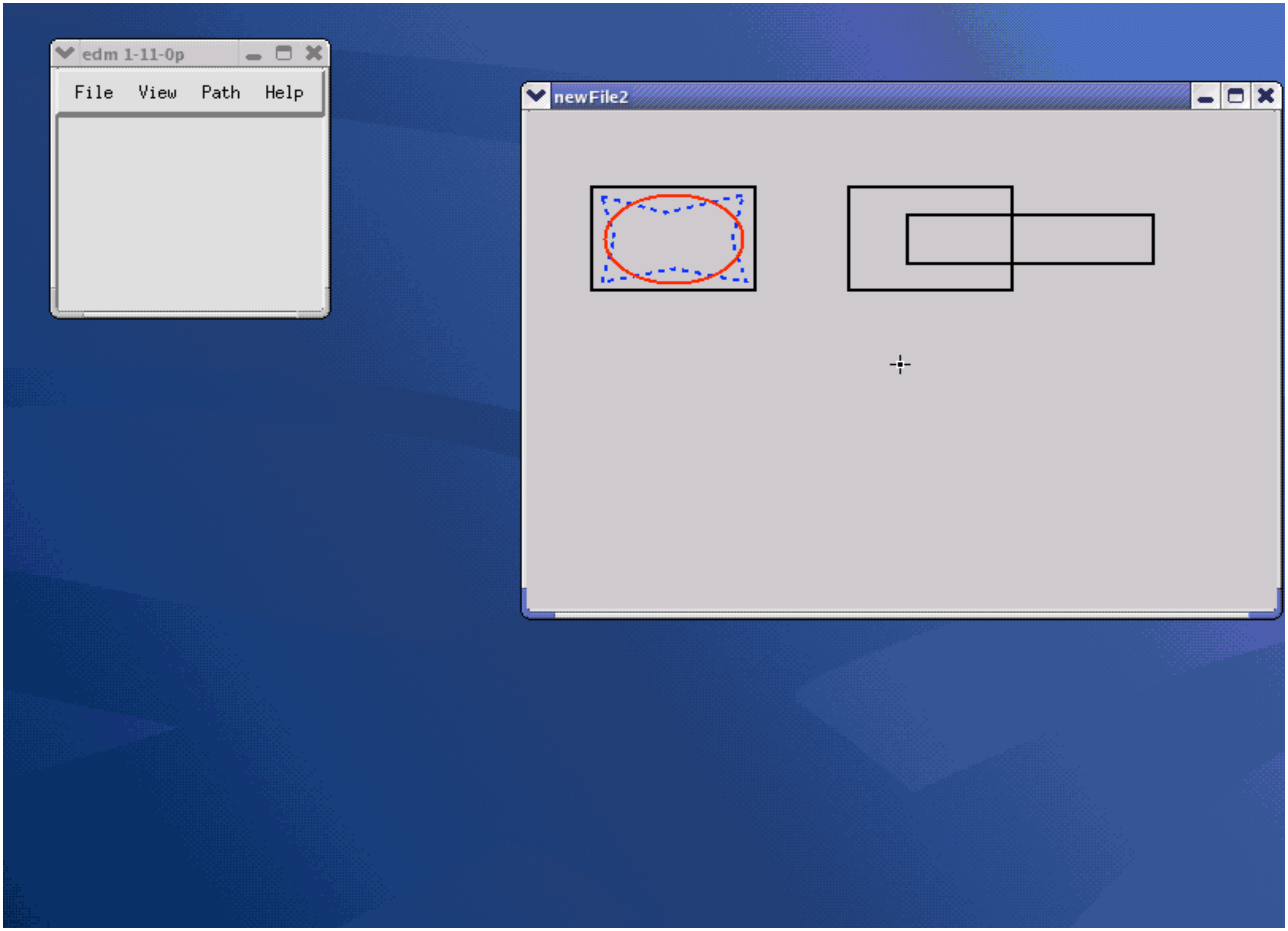
Note:
Property dialog varies with Object type…

Editing Notes

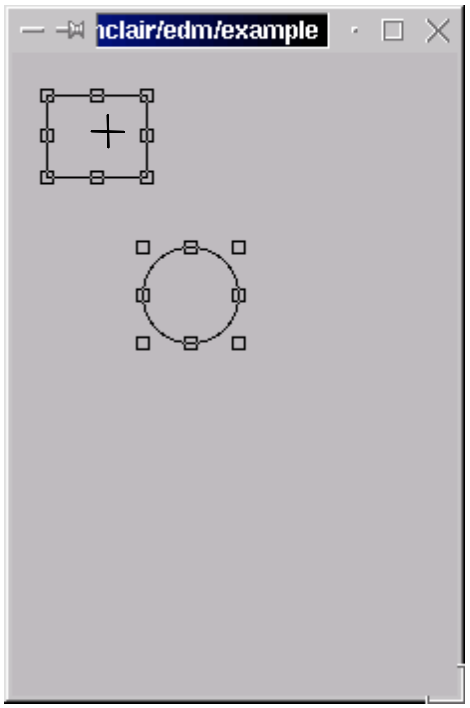
- Clicking on one of a *group* of selected objects brings up the property box for each object, one-by-one, as the OK button is pressed.
- To minimize mouse movement, instead of clicking OK, Apply, or Cancel, you may *double-click* the left, middle, or right button respectively.

Left-Click is Context Sensitive…

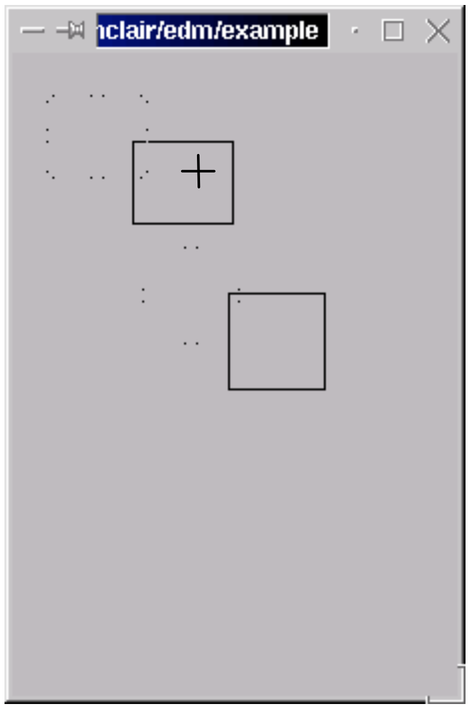
- Left-click can mean select or edit
- Watch video


edm 1-11-0p
File
View
Path
Help
newFile2

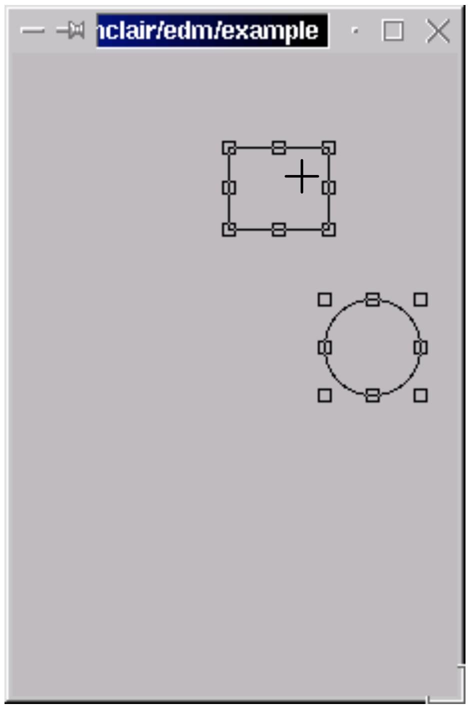
Moving Objects



Place mouse cursor on interior of one object

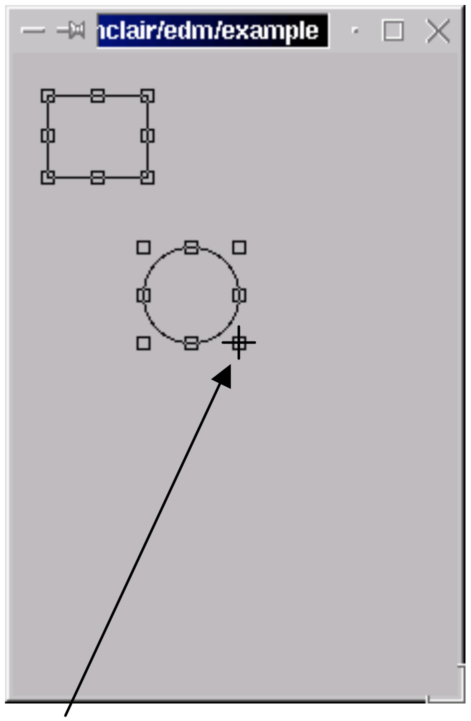


Press left button and drag objects to new location

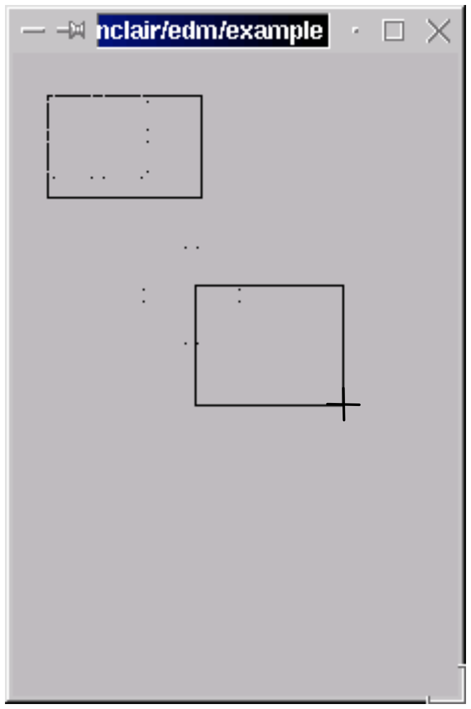


Release mouse button

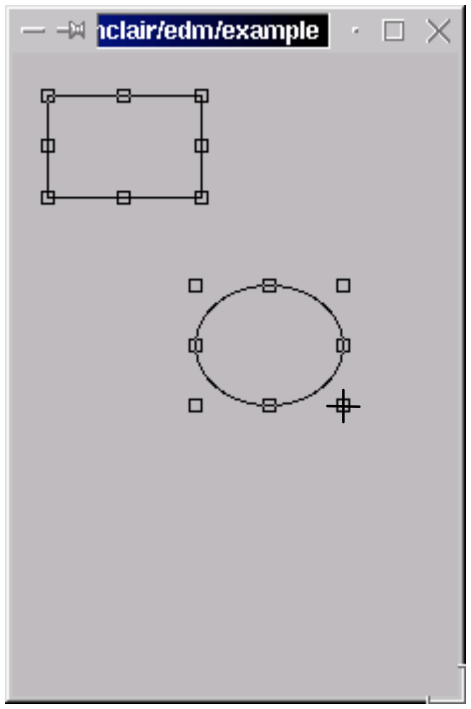
Resizing Objects



Place mouse cursor on control point of one object



Press left button and drag to new size



Release mouse button

Draw/Move/Resize Notes

- Fine control may be achieved on moves and resizes by using *keyboard arrow keys* (mouse button release or click ends op)
- Control key forces move (prevents resize) (Useful for tiny objects where you cannot click “inside” w/o hitting the resize handles)
- Shortcuts to options in the Display Properties
  - M/m key turns ON/off orthogonal move
  - L/l key turns ON/off orthogonal line draw
  - G/g key turns ON/off grid
  - S/s key turns ON/off snap-to-grid

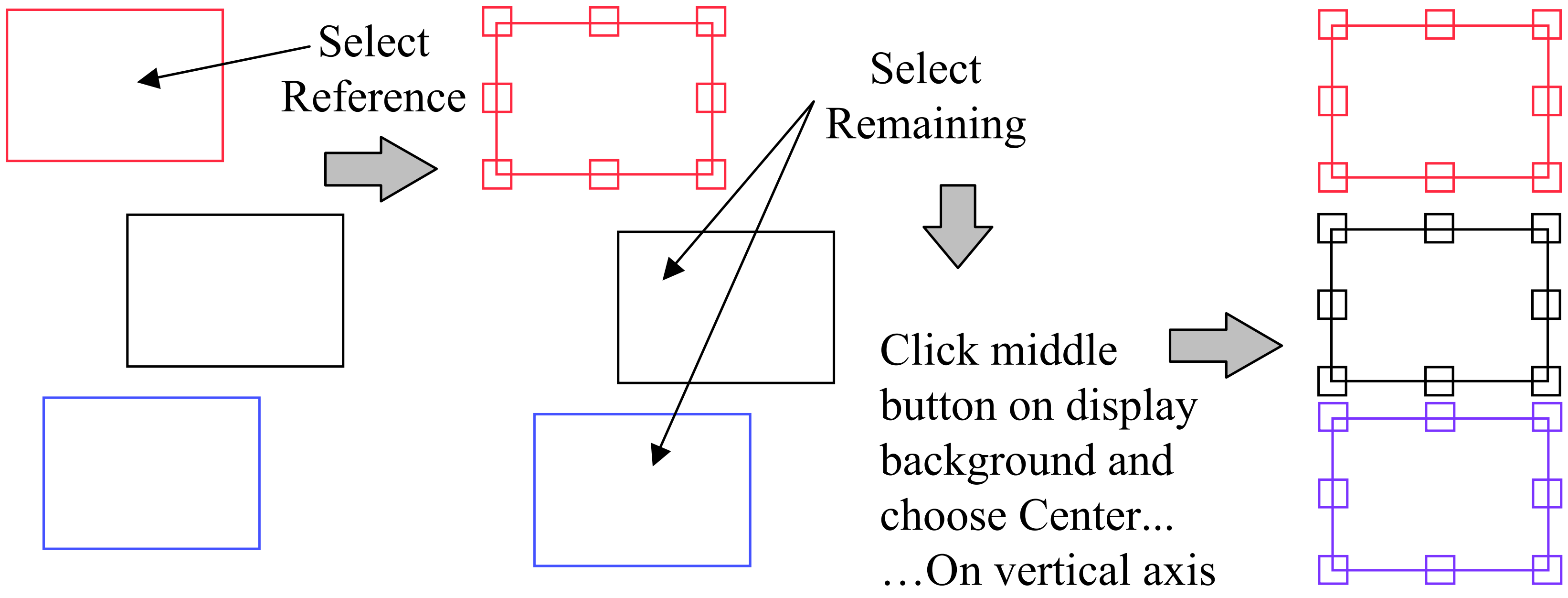
Alignment Operations

- Reference Independent
  - Align left, right, top, bottom
- Reference Dependent
  - Center: horizontal, vertical, both
  - Size: width, height, both
  - Distribute: vert axis, horiz axis
  - Distribute Midpoint: vert axis, horiz axis

Reference Dependent Operations

- First object selected is used as reference
- If no reference object is specified, an appropriate object is chosen (topmost, leftmost, etc.)

Example Center-Align Operation



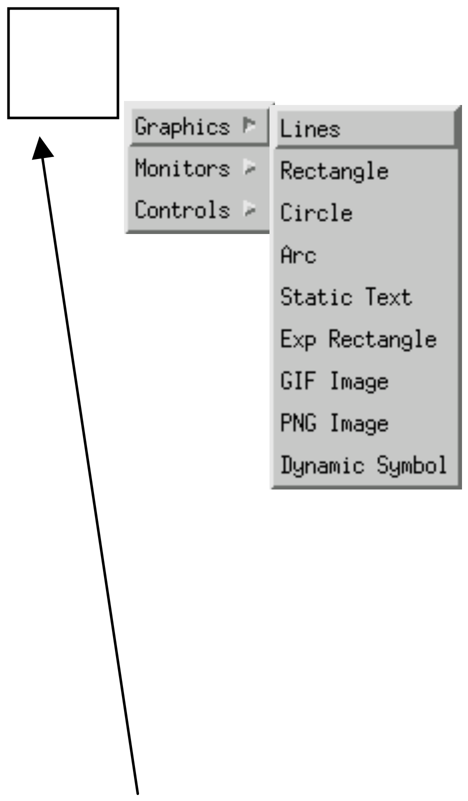
Misc. Operations

- Raise, Lower (u,d)
- Copy, Cut, Paste (c,x,v)
- Group, Ungroup ([,])
- Flip H & V
- Rotate CW & CCW
- Group Edit
- Undo

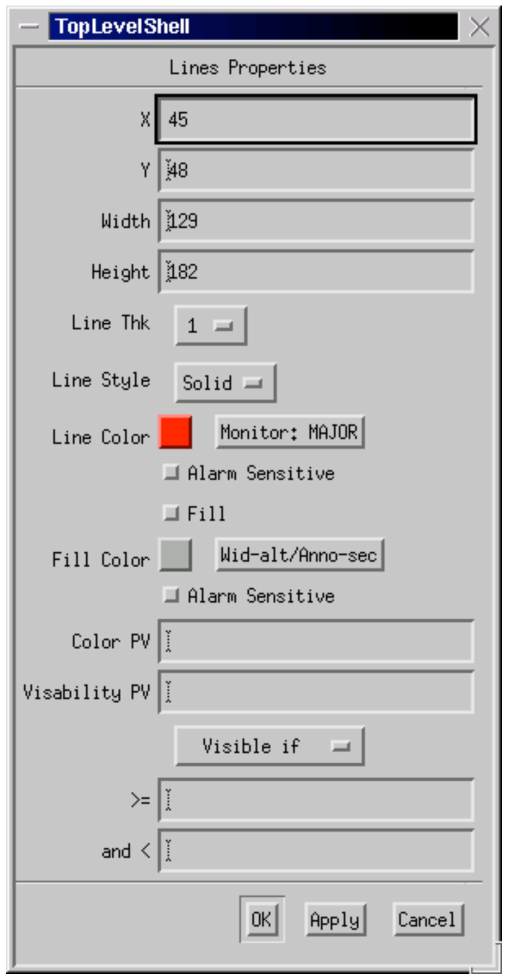
Undo

- Most useful for move, resize, & alignment operations
- Current limitations:
  - Cannot undo edit operations
  - Cut, Group, and Ungroup : Flush undo stack

A bit different: Creating Lines



Left mouse button drag

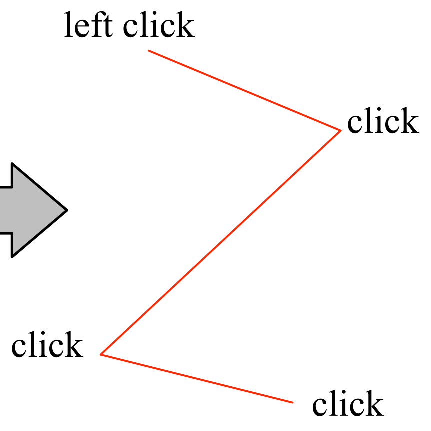


TopLevelShell

Lines Properties

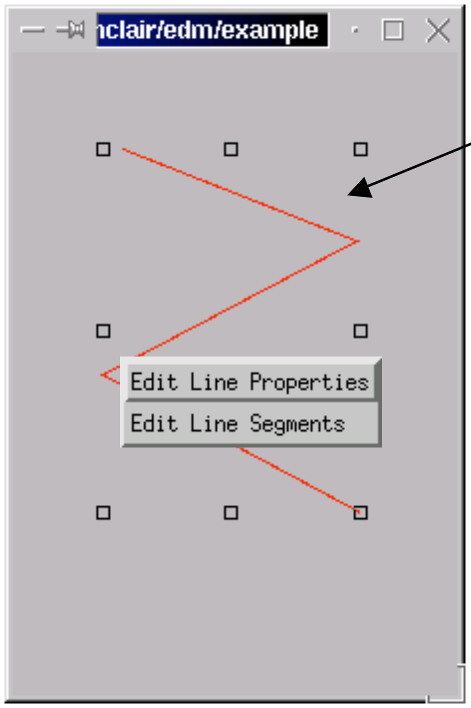
X 45
Y 48
Width 129
Height 182
Line Thk 1
Line Style Solid
Line Color Monitor: MAJOR
Alarm Sensitive
Fill
Fill Color Wid-alt/Anno-sec
Alarm Sensitive
Color PV
Visability PV
Visible if
>=
and <

OK Apply Cancel



shift-click
or
double click
to finish

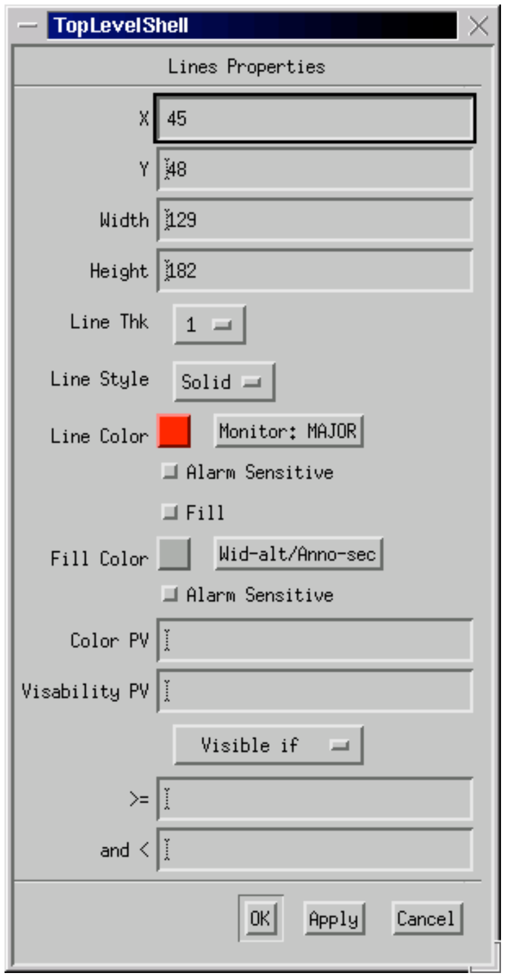
Editing Line Properties



left click on selected object

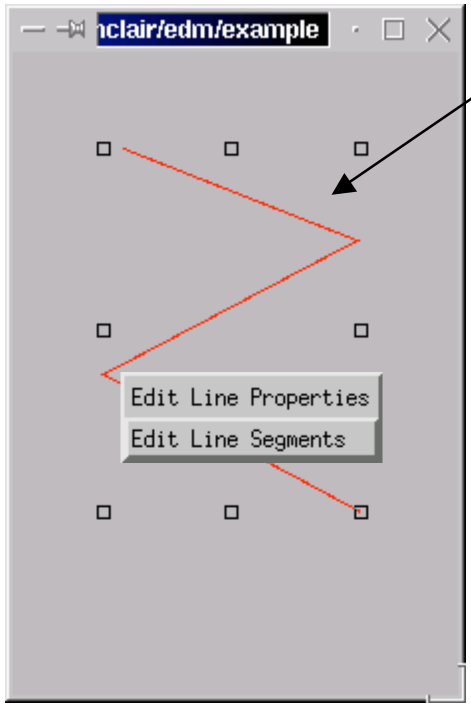
Menu Appears

Choose Edit Line Properties



X	45
Y	48
Width	129
Height	182
Line Thk	1
Line Style	Solid
Line Color	🟥 Monitor: MAJOR
	Alarm Sensitive
	Fill
Fill Color	Wid-alt/Anno-sec
	Alarm Sensitive
Color PV	
Visability PV	
	Visible if
>=	
and <	

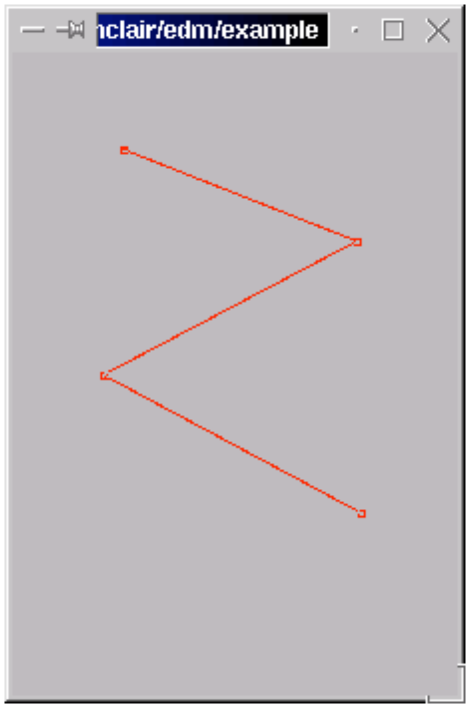
Editing Line Segments



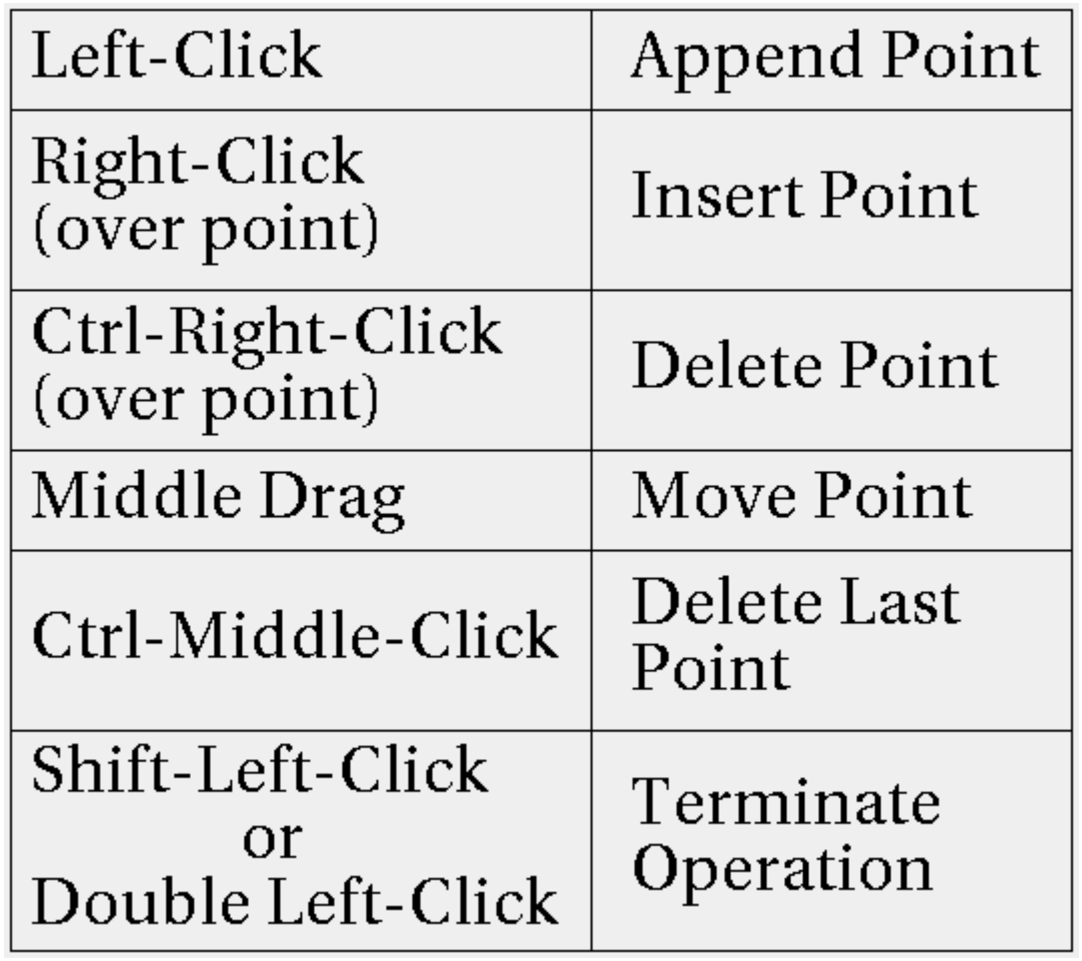
left click on selected object

Menu Appears

Choose Edit Line Segments



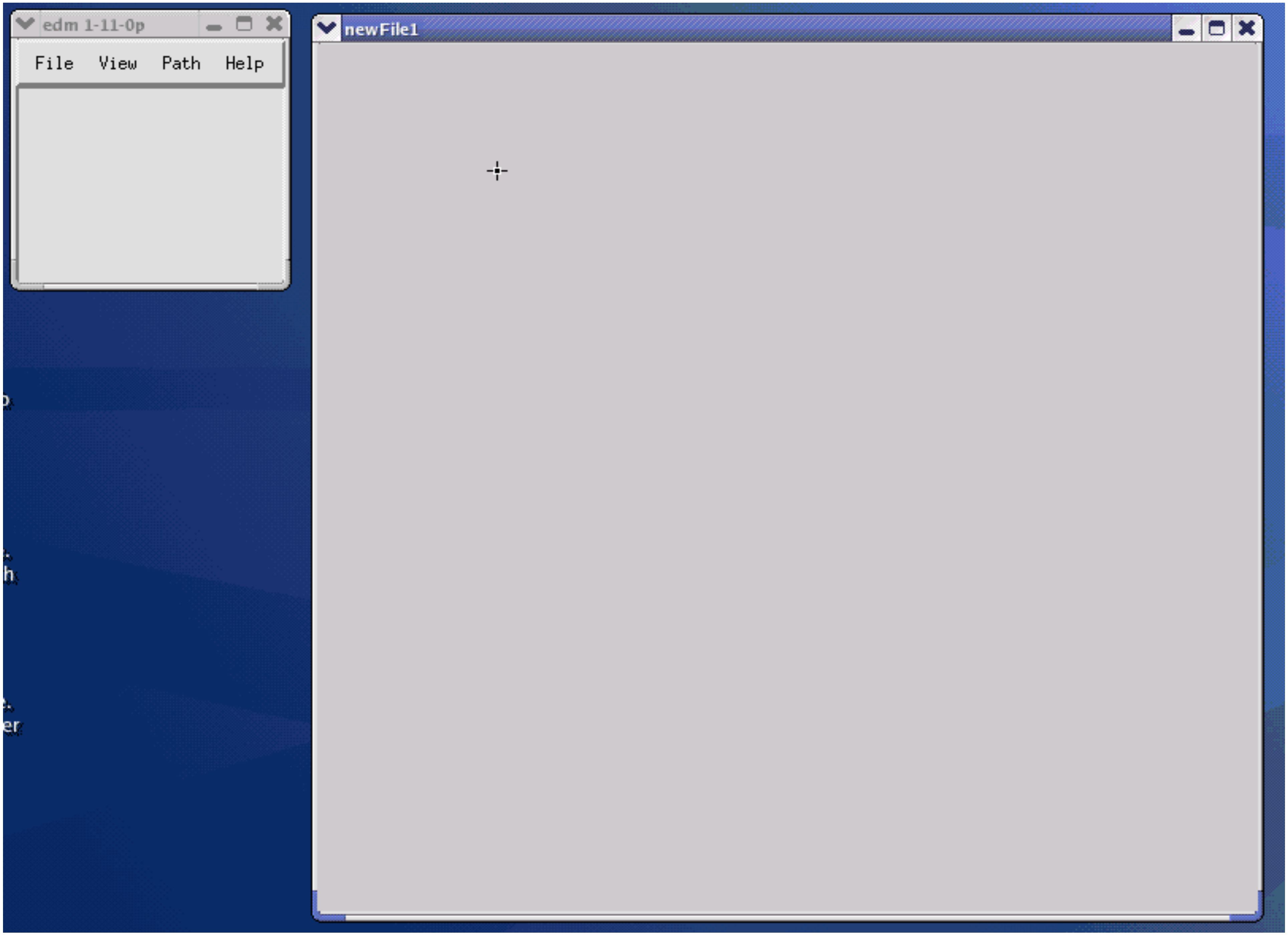
Editing Line Segments (cont)



Left-Click	Append Point
Right-Click (over point)	Insert Point
Ctrl-Right-Click (over point)	Delete Point
Middle Drag	Move Point
Ctrl-Middle-Click	Delete Last Point
Shift-Left-Click or Double Left-Click	Terminate Operation

Editing Line Segments (cont)

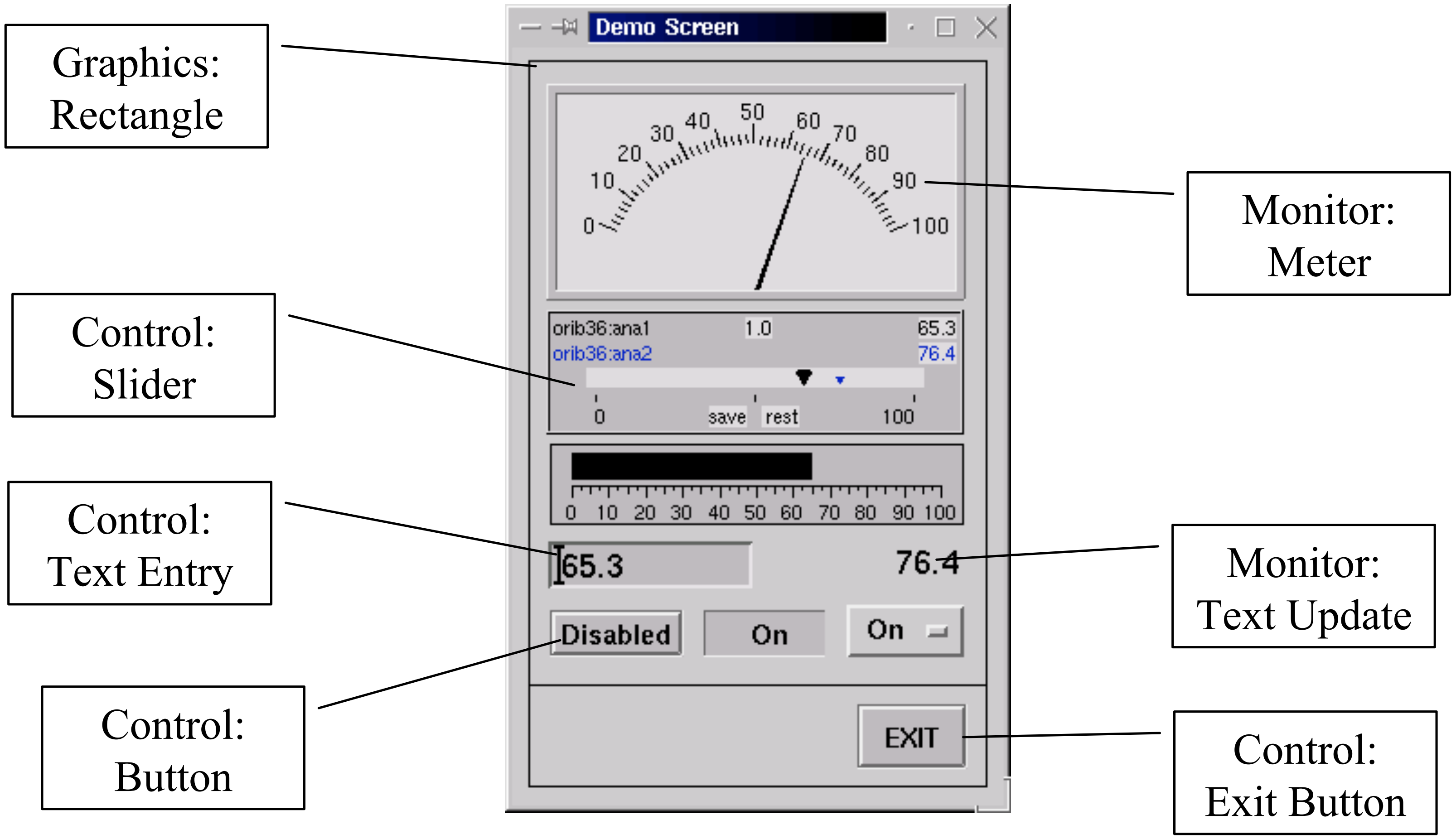
- Lots of details but, once mastered, easy to manipulate lines
- Watch video


edm 1-11-0p
File
View
Path
Help
newFile1

Group Edit

- Change visual attributes of all selected objects
- Change PV names for all selected objects

EDM Objects

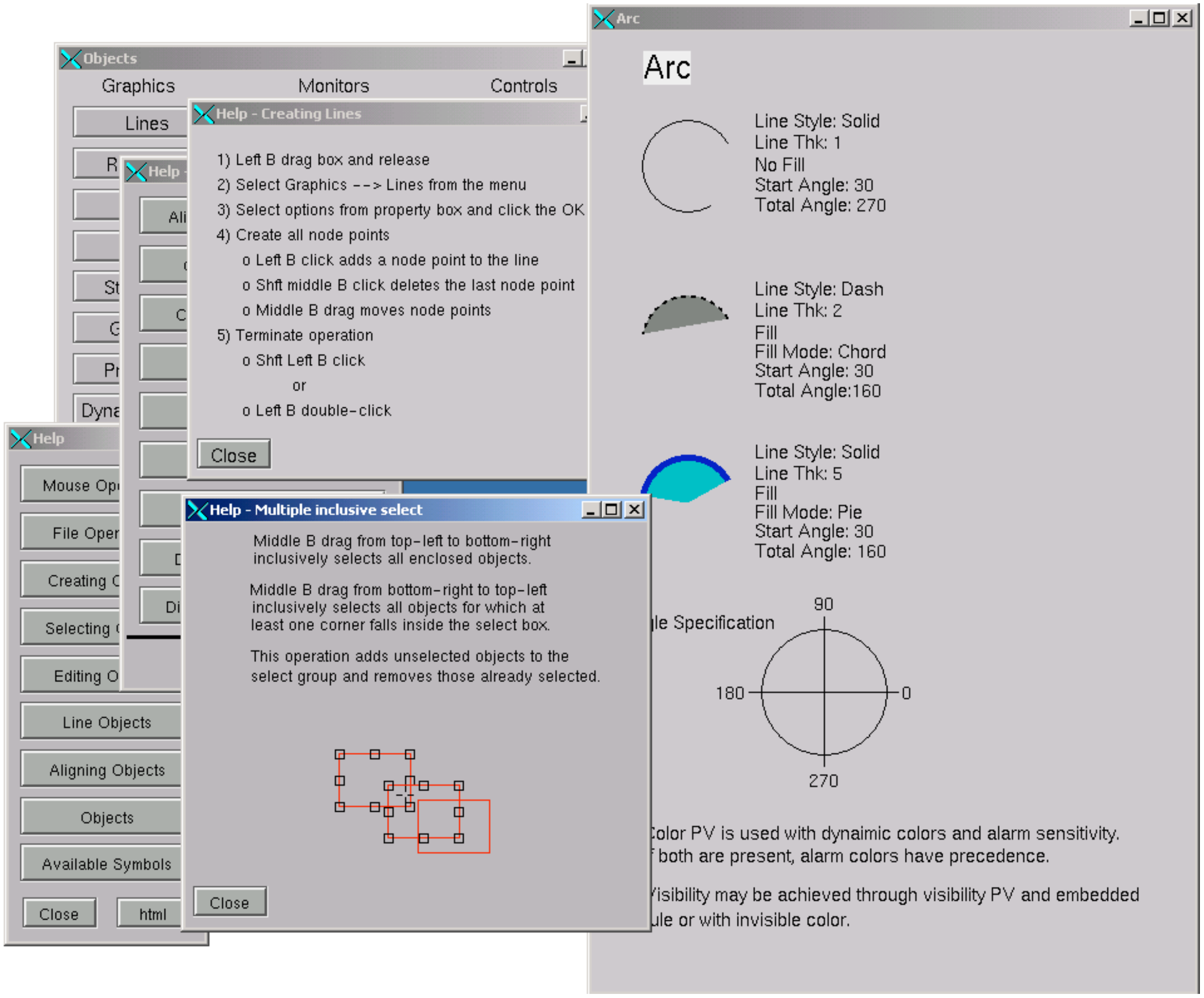


Graphics: Rectangle
Monitor: Meter
Control: Slider
Control: Text Entry
Monitor: Text Update
Control: Button
Control: Exit Button
Demo Screen
0
10
20
30
40
50
60
70
80
90
100
orib36:ana1
1.0
65.3
orib36:ana2
76.4
0
save
rest
100
0 10 20 30 40 50 60 70 80 90 100
65.3
76.4
Disabled
On
On
EXIT

Object Categories

- Graphics
  Do not require a process variable
  - Lines, rectangle, circle, arc, text, gif, png, dynamic symbol, embedded window
- Monitors
  Display current value of process variables
  - Meter, bar, message box, symbol, text update, x/y graph,...
- Controls
  Modify value of process variables, change displays
  - Text, slider, button, menu button, message button, up-down button, related display, shell command,exit button …

Online Help



Help - Creating Lines

1) Left B drag box and release
2) Select Graphics --> Lines from the menu
3) Select options from property box and click the OK
4) Create all node points
   o Left B click adds a node point to the line
   o Shft middle B click deletes the last node point
   o Middle B drag moves node points
5) Terminate operation
   o Shft Left B click
   or
   o Left B double-click

Help - Multiple inclusive select

Middle B drag from top-left to bottom-right inclusively selects all enclosed objects.

Middle B drag from bottom-right to top-left inclusively selects all objects for which at least one corner falls inside the select box.

This operation adds unselected objects to the select group and removes those already selected.

Line Style: Solid
Line Thk: 1
No Fill
Start Angle: 30
Total Angle: 270

Line Style: Dash
Line Thk: 2
Fill
Fill Mode: Chord
Start Angle: 30
Total Angle:160

Line Style: Solid
Line Thk: 5
Fill
Fill Mode: Pie
Start Angle: 30
Total Angle: 160

le Specification

olor PV is used with dynaimic colors and alarm sensitivity. both are present, alarm colors have precedence.

isibility may be achieved through visibility PV and embedded ule or with invisible color.

Process Variables

- Many EDM objects accept PVs to
  - show the PV value (Monitors)
  - control the PV value (Controls)
  - change color or visibility based on the PV (all types)
- Format:
  - `EPICS\fred`
    Use EPICS ChannelAccess to connect to “fred”
  - `fred`
    Use default method which is “EPICS” ⇒ same as above
  - **LOC\locpv1=d:0 (name=type:value)**
  - **LOC\locenum1=e:0,zero,one,two (name=e:init,state0,state1...)**
  - `CALC\sum(fred, 2)`
    Use CALC PV “sum”, provide arguments “fred” and “2”.
  - `XY\fred`
    Use method XY (not implemented)…

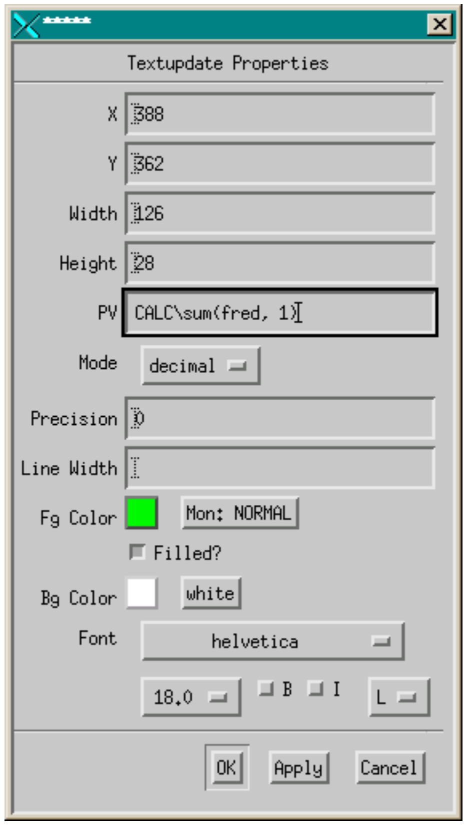
“CALC” PVs

```
# File calc.list (EDMFILES)

# sum(A,B)
sum
# Implementation:
A+B

# F2C(A)
F2C
(A-32)*5/9
```

- CALC: Formula ala CALC record
- Selected via prefix “CALC\”
  (default is EPICS = Channel Access)
- Examples:
  - Convert Deg.F into Deg.C inside EDM:
    `CALC\F2C(EPICS\temp_F)`

    `CALC\{(A-32)*5/9}(temp_F)`

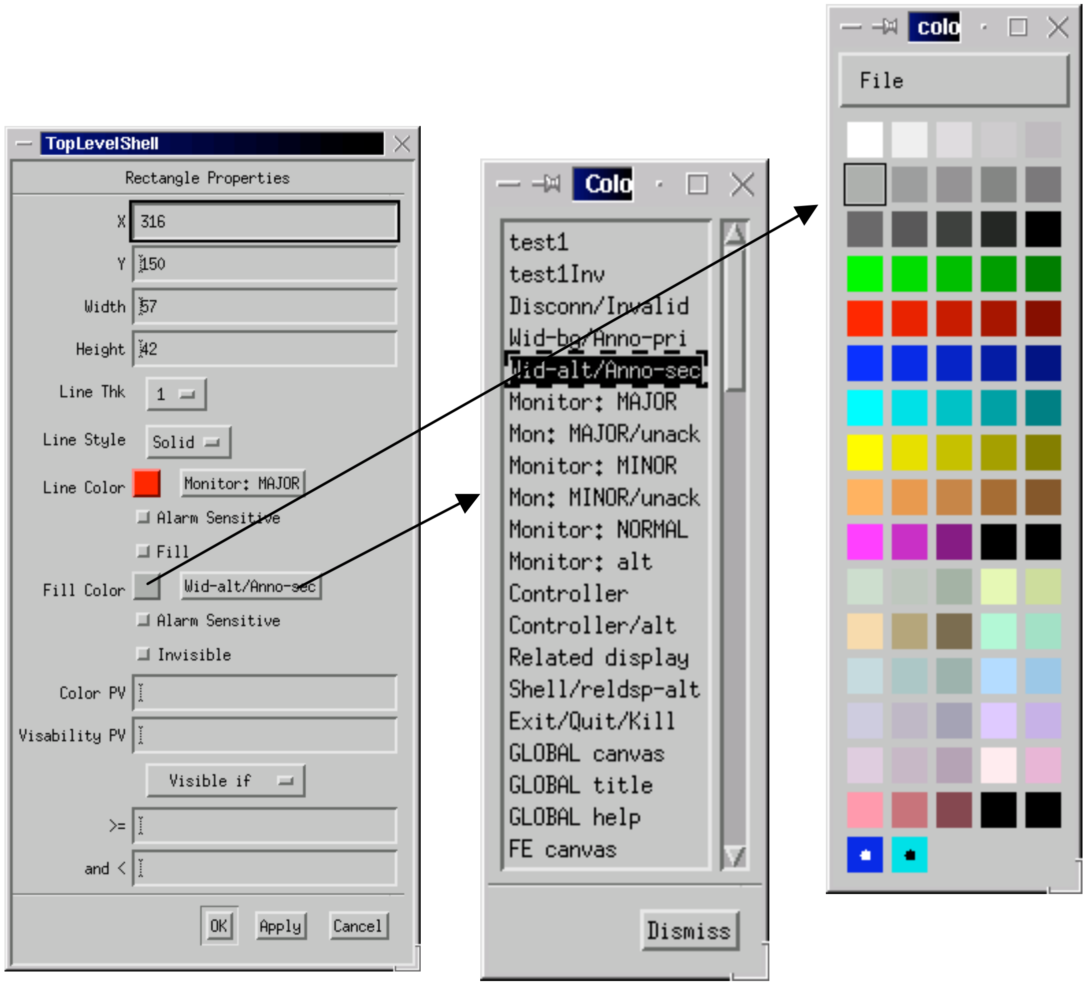
    Volume of Martini from ingredients:
    `CALC\sum(gin,CALC\sum(water,tonic))`



Textupdate Properties

X	388
Y	362
Width	126
Height	28
PV	CALC\sum(fred, 1)
Mode	decimal
Precision	0
Line Width	1
Fg Color	🟩 Mon: NORMAL
	Filled?
Bg Color	⬜ white
Font	helvetica
	18.0 B I L

Specifying Color



Rectangle Properties

X	316
Y	150
Width	57
Height	42
Line Thk	1
Line Style	Solid
Line Color	🟥 Monitor: MAJOR
	☐ Alarm Sensitive
	☐ Fill
Fill Color	Wid-alt/Anno-sec
	☐ Alarm Sensitive
	☐ Invisible
Color PV	
Visability PV	
	Visible if
>=	
and <	

```
test1
test1Inv
Disconn/Invalid
Wid-bg/Anno-pri
Wid-alt/Anno-sec
Monitor: MAJOR
Mon: MAJOR/unack
Monitor: MINOR
Mon: MINOR/unack
Monitor: NORMAL
Monitor: alt
Controller
Controller/alt
Related display
Shell/reldsp-alt
Exit/Quit/Kill
GLOBAL canvas
GLOBAL title
GLOBAL help
FE canvas
```

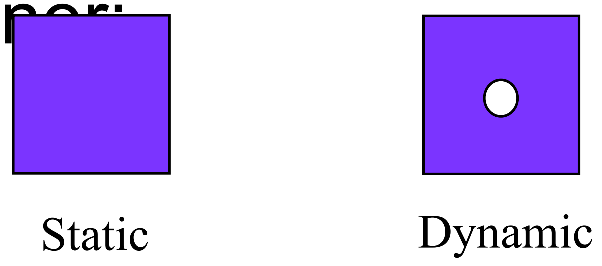
- Color may be specified visually or by name
- Website documentation explains the current color file format
- The color palette dialog shows names as “tooltips”
- *Decoration or Meaning?*
  Example:
  The same shade of red might be available as both “red” and “Monitor: MAJOR”. Pick the one that fits the desired purpose.

Color - Static and Dynamic

- Some color entries are dynamic and are associated with a color rule
- In execute mode, dynamic colors change as a function of the color rule operating on the current value of an associated PV
- When selecting “alarm sensitive”, the color will change based on the PV alarm severity.

Color - Static and Dynamic

- Colors may be specified for various object attributes and appear as one or more buttons in object property dialog boxes. Dynamic colors are differentiated from static colors in the following manner:



- For color function, refer to the colors.list file and internal site documentation

Color Rules

- Color Rules are defined in the edm color.list file. The following is an example of a rule:

```
rule Red-or-Blue
{
  <5          : red
  >=5         : blue
}
```

  – This color will be “red” or “blue” depending on the value of the PV.

- Some objects provide a separate “Color PV” that can be used instead of the “main” PV for rule evaluation.

EDM Macro Expansion

- Macro symbol sources
  - Command line
  - Related Display or embedded window parameter
  - Multiplexor Object
- At run-time, symbol expands to associated value

e.g. command line option -m "one=1"

at run-time, $(one) $\longrightarrow$ 1

Symbols

- EDM implements a primitive symbol facility
- Symbols are multi-state objects where each state maps to a value range of an associated EPICS PV
- 64 states max, color and size may be changed per symbol instance if so desired

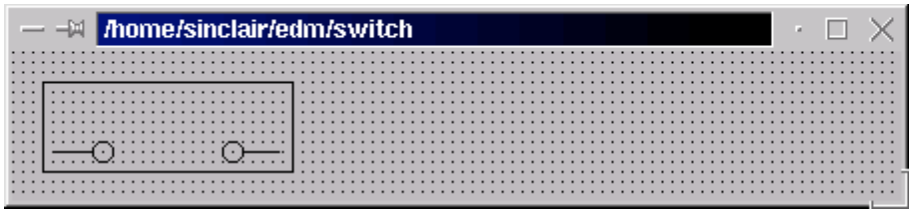
Symbols (cont)

- An EDM symbol is nothing more than a standard display file where each symbol state is represented as a group of objects
- Only one grouping level is allowed
- The visual ordering corresponds to the ordering of states
- EDM contains an auto-make symbol command to perform the grouping and ordering

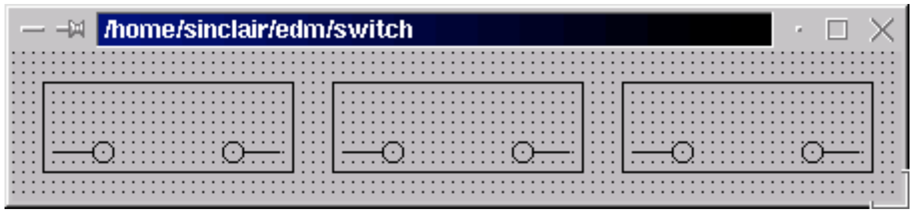
Creating Symbols

1. Create a rectangle corresponding to the geometric boundaries of the symbol, check the invisible attribute of this rectangle

2. Draw the invariant visual components of the symbol

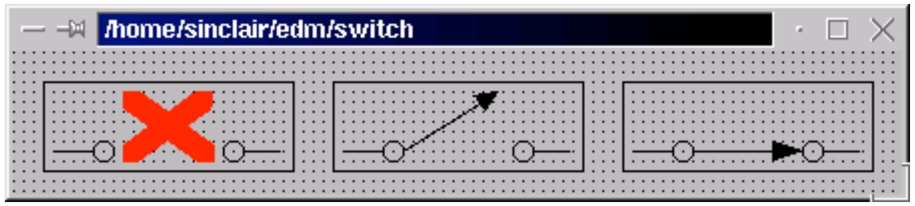


3. Copy this information and paste it N times, you now have N+1 visual states



Creating Symbols (cont)

4. Draw the state dependent visual components, the first state should be the out-of-band state, the second state is displayed in edit mode



5. Make sure no grouped objects exist, click the middle mouse button on the display background, and choose *Auto make symbol* from the menu
6. Save the EDM display file, this file may now be used as a symbol file

Deploying Symbols

- A symbol instance is created like any other EDM object
- One property of a symbol instance is the symbol file name; this is the file discussed previously
- An exercise will illustrate this entire process in detail

Summary

- EDM is only one of the available EPICS display managers
  - EDM has many useful editing features to support efficient display manipulation
  - List of EDM objects can be extended, even new PV types can be added

Exercises

- For all excercises, know *where* you are!
  e.g. if you are user 3, change your working directory to
  ```
   <your training dir>/t3
  ```
  In most cases, edm will load & save files from there. Only for color and default schemas will it go to the $EDMFILES directory
- Know *who* you are (training user t1, t2, …) and what IOC you are using, then start edm as
  ```
  edm –m “user=t3” &
  ```

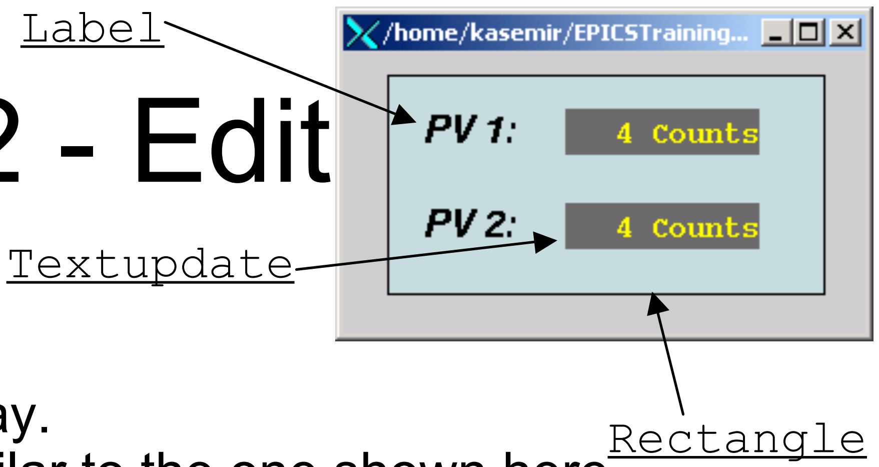
Exercise 1- Start, Display Schemes

- Execute edm: Type something like `edm –m “user=t3” &`
- Create a new display: File/New
- Invoke the middle(!) mouse button menu, select “Display Properties” (with the left(!) mouse button).
- In here, you can change the background color and select default fonts and colors for newly created display elements.
  Select a background color, click OK.
- You would save the “Display Scheme” in case you want to create a standard setup (backgound color etc.) for part or all of your displays. In fact EDM automatically loads the file named “$EDMFILES/default.scheme” on startup, so by saving a scheme under that name, you determine the start-up defaults.

.. Display schemes

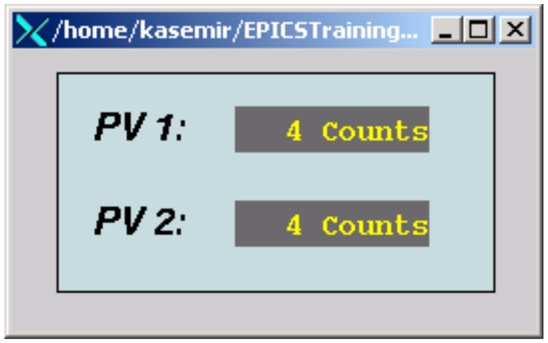
- One quirk/feature is that EDM wants to save & load all display schemes in a global EDM configuration directory ($EDMFILES). That makes sense when you want to share display schemes with others, especially for the “default.scheme”. For this training, however, we suggest to keep all your files in your home directory.
- Invoke the middle-button menu again, select *Save Display Scheme*, navigate to your training directory, and create *training.scheme* (yes, the Motif-type file dialog is strange).
- From the main window, choose File/Exit.
  There will be a warning because you didn’t save the display.
  That’s OK: We don’t care about the display, we only wanted to create a “Display Scheme”!

Exercise 2 - Edit



- Execute edm, create a new display.
  We intend to create a display similar to the one shown here.
- Apply the *training.scheme* from Exercise 1:
  Middle-button menu, “Load Display Scheme…”, navigate to your directory, …
- Save the display as “example2”
  (from now on, save every once in a while just in case …)
- Create
  - two Labels “PV 1” and “PV 2”: Left-button-rubberband the approximate area. Select Graphics/Static Text. Enter “PV 1” as the text. Select font helvetica 18. Press OK. Same for “PV 2”.
  - two Monitor/Textupdate objects, for PV names use e.g. “$(user):aiExample” and “$(user):calcExample”.
  - one Graphics/Rectangle, make it “filled”
    Note: To change the stacking order, select objects, then use the middle-button menu to raise or lower them.

Exercise 2…

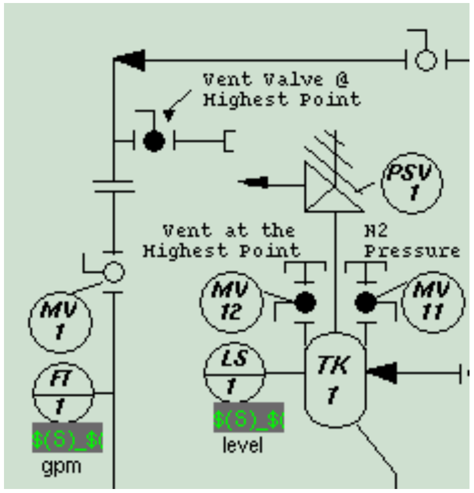


PV 1: 4 Counts

PV 2: 4 Counts

- Use Select, move, resize, align, …
  until the display looks a bit like the example shown on this slide
  (it's shown in execute mode)
- These help to finish quicker:
  - Display Properties: snap-to-grid, ortho move & line draw
  - Copy/paste
- Switch to "execute" mode: de-select all objects, click the middle mouse button, and choose *execute* from the menu

Exercise 3 – More Editing



- Create a new display, save it as example3
- Unless you already remember everything: Launch Help/Line Objects
- Use Graphics objects (circles, text and mostly lines) to create some of the elements you see in the screenshot
- Select several objects at once, change color or font or … via the "Edit/Display Properties" option

Exercise 4 - PVs

- Execute edm and open example2.
  We will see how different PVs can be represented in different ways.
- Save the display file as “example4”
- From the example2. there should be a text update for the record “$(user):calcExample”
- Add a Monitor/Meter uses the same PV “$(user):calcExample” - execute
- Add a text entry control to the .CALC field of the record:
  Create a Control/Textentry, use “$(user):calcExample.CALC” as a PV name.
- Add a Control/Menu Button to the .SCAN field of the same record (e.g. PV name “$(user):calcExample.SCAN”) - execute
- Add a text update that displays a calculated PV, e.g. “CALC\sum($(user):calcExample, 2)” - execute

Exercise 5 – Colors, Macros

- Execute edm with the option “-m user=t1” (or t2, t3, t4, …).
  We will see how macros get passed to related displays.
- Open last example, save as “example5”.
  This will be our main display that launches off two related displays.
- Add a label (Graphic/Static text) that shows “User $(user)”
- Add two Control/Related Display buttons
  - Set “File” to “relatedDsp”, “Macros” to “param=1”,
    “Button Label” to “Rel. 1”
  - Config. Of second button:
    Set “File” to “relatedDsp”, “Macros” to ”param=2”, “Button Label” to “Rel. 2”
- Add a text entry control
  - Obtain the Control PV name from an instructor or use “$(user):aoExample”. This same PV name will be used in a color rule inside the related display
- Add an exit button. Check the “Exit Program” option.

Exercise 5…

- Create a new display to be used as the related display, save it as “relatedDsp”
  - Create a static text object with *Text Value* set to
    “Related Display, param=$(param)”
  - Create a rectangle, choose a dynamic color for *line color*, use the PV name from above
  - Create an exit button. Do <u>not</u> check the “Exit Program” option.
- Save *relatedDsp* and close it.

Exercise 5…

- Execute “example5”
- Click the Related Display button, the associated display should appear and the static text object should display the symbol value
- Change the value of the PV from the example1 text control, the rectangle color should be determined by the color rule
- Click the Exit Button on each related display.
- Click the Exit Button on “example5”, the main screen.
- Exit edm?

Exercise 6: Symbols

- Create a new display, save it as “switch”
- Follow the “Creating symbol” slides to create the states of a simple switch:
  - undef, open, closed
- Details:
  - Create invisible rectangle
  - Draw invariant symbol components
  - Copy image and paste two copies to the display
  - Draw state dependent components

Exercise 6

- Arrange images in a rows/columns ordering, first state is upper-left, last is lower-right
- If any objects have been grouped, ungroup now
- Click middle button, choose *Auto make symbol*
- Save symbol file as switch & close the display.

- Create new display “example6”
  - Add text entry to control e.g. “$(user):aoExample”
  - Add a symbol instance (Monitor/Symbol)
    - Use symbol file recently created, use same PV as referenced in text entry object. Select 3 items, configure each as follows:
      Item 1: 0 <= PV value < 1
      Item 2: 1 <= PV value < 2
      Item 3: 2 <= PV value < 3
- Execute example6

Exercise 7: Command Button

- Open any of the examples
- Add a shell command button to start StripTool
- Execute
- Note how you can drag & drop (middle button) PV names from an edm object to StripTool’s channel name field

Exercise 8 – Colors, Macros with embedded window

- This exercise demonstrates how macros get passed to embedded windows.
- Open “example2”, save as “example8”.
- Add an embedded window large enough to show the contents of relatedDsp from example 5.
  - Set “Display Source” to “Menu”
  - Set “PV” to “LOC\select=i:0”
  - Check “Center”
  - Click “Menu Info” and configure the menu properties
    Set 1st “Label” to “One”, “File” to “embDsp”, “Macros” to “param=1”
    Set 2nd “Label” to “Two”, “File” to “embDsp”, “Macros” to “param=2”
    Close window and Click OK on main property window

Exercise 8…

- Add two message buttons above the embedded window area
  - Set 1st “Destination PV” to “LOC\select=d:0”, “Press Label” to “1”, “Press Value” to '”0”, and “Release Label” to “1”
  - Set 2nd “Destination PV” to “LOC\select=d:0”, “Press Label” to “2”, “Press Value” to “1”, and “Release Label” to “2”
- Add a text entry control
  - Obtain the Control PV name from an instructor or use “$(user):aoExample”. This same PV name will be used in a color rule inside the related display
- Add an exit button. Check the “Exit Program” option.
- Open file relatedDsp
- Edit the exit button and check “Apply to parent”
- Save as embDsp, and close.

Exercise 8…

- Execute “example8”
- Click the Message buttons, the associated display should replace the embedded window contents and the static text object should display the symbol value
- Change the value of the PV from the text control, the rectangle color should be determined by the color rule.