

MEDM/ALH/AR

- EPICS CA clients

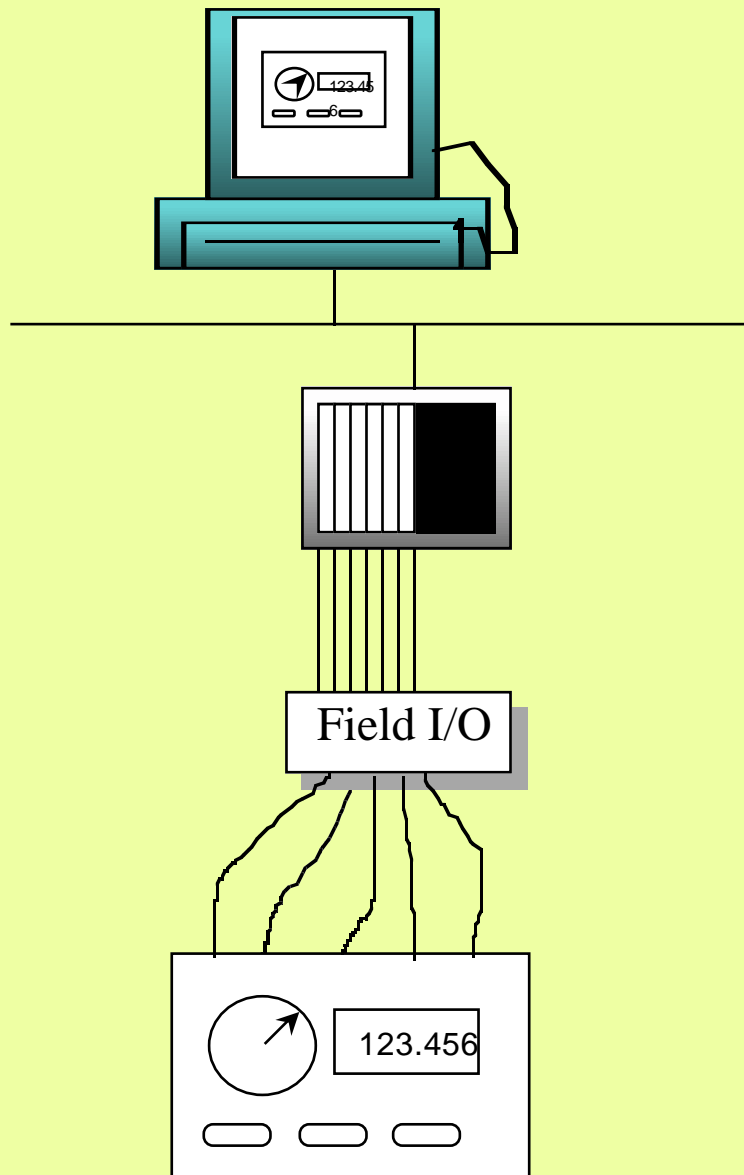
MEDM(Motif Based Editor & Display Manager)

ALH(EPICS Alarm Handler)

AR(EPICS Archiver)

- MEDM practice

EPICS terminology



Operator Interface (OPI)

On OPI, A device is represented as a set of channels (or records).

CA Client software runs on OPI.

I/O controller (IOC)

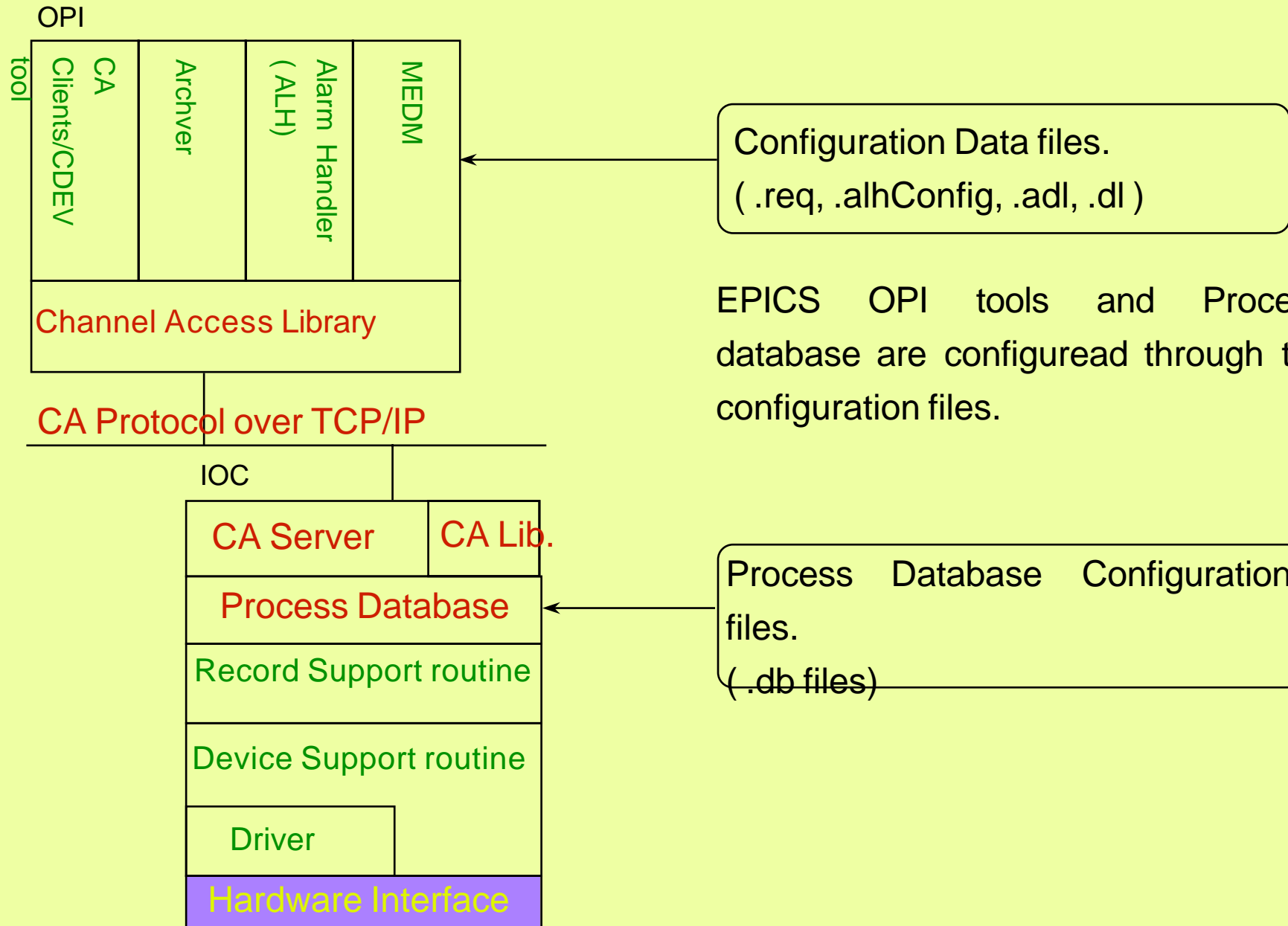
Process Database keeps an information of a device as a set of records.

Channel: Basic unit of Data exchange in EPICS.

EPICS CA clients

- General Purpose Tools for control system.(OPI tools)
 - EDD/DM(EPICS OPI Display Editor/ Display Manager)
 - MEDM(Motif based Editor & Display Manager)
 - jdm(java based display manager)
 - ALH (Alarm Handler)
 - AR(EPICS ARchiver)
- Tools to support a development of custom applications
 - Tcl/Tk
 - CaMath/devMath(EPICS CA in Mathematica)
 - java/cdev
 - Python/Tk (under development in KEK)
 - SAD/Tk (Programmable Modeling Program with GUI object) and more

Software Layers in EPICS



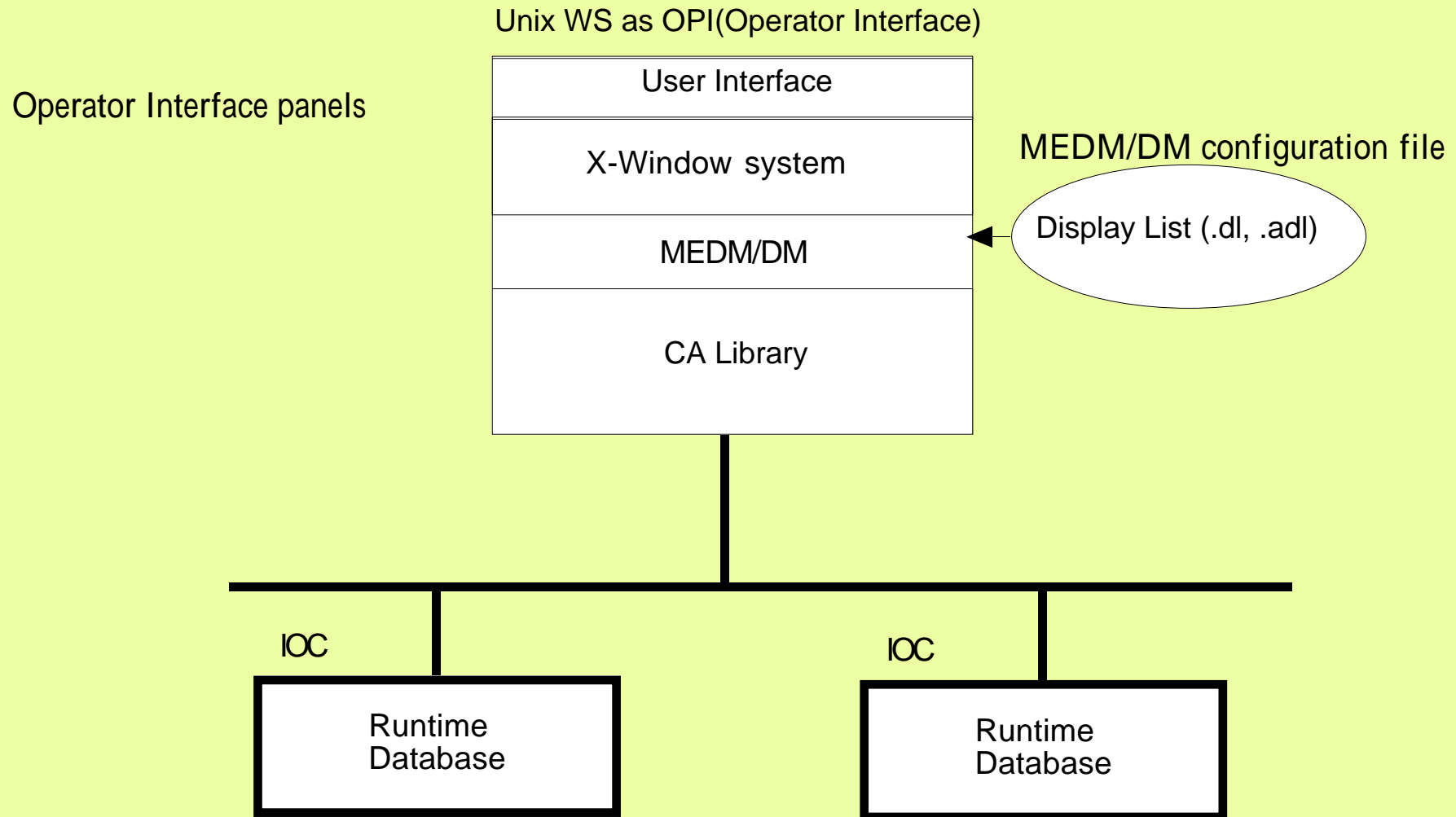
EPICS OPI tools and Process database are configured through the configuration files.

Configuration files for EPICS OPI tools

Tool	File Extension	data type
EDD/DM	.dl/.adl	binary/ASCII
MEDM	.adl	ASCII
ALH	.alhConfig	ASCII
AR(AR_cmd)	.req	ASCII

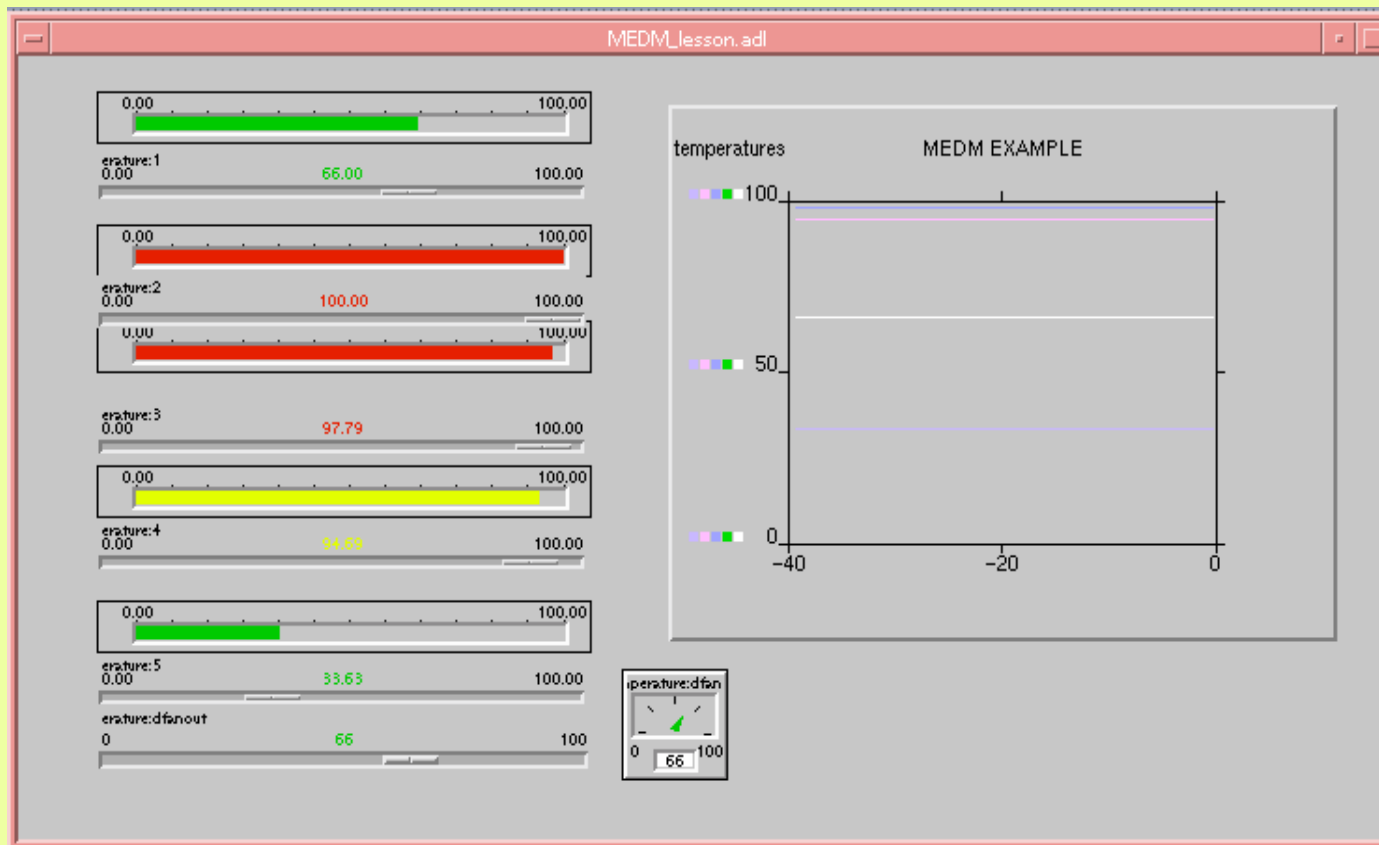
*) File Formats of .adl files for EDD/DM and for MEDM are slightly different. Unification of two formats is in progress.

OPI Overview



OPI Overview

- DM/MEDM are CA client tools to display operator interface panels.
- DM/MEDM are configured by configuration files.
- EDD/MEDM are used to create the configuration files.



MEDM command line syntax

```
% medm [-x | -e] [-local | -cleanup] [other options]
        [-displayFont alias | scalable]
        [-macro "<name>=<value>,<name>=<value>..." ]
        [...X based options...] [display file names]
```

parameters:

-displayFont select alias for scalable fonts
-macro apply macro substitution

options:

-x startup medm in execute-only ("dm") mode
-e startup medm in edit/execute ("edd/dm") mode (the default)
-readonly run in read only mode (no CA put)
-local don't participate in remote display protocol
-cleanup support remote display protocol, ignore existing instances
-cmap use private color map
-bigMousePointer use big cursors
-silent reduced error messages
-wmPositionPolicy define the WM positioning policy : FRAME or USER
-verbose display verbose error messages
-debug provide debugging information
-synchro run X11 protocol in synchronous mode
-help this message

X11 resource options:

-display
-displayGeometry

Display filenames:

a list of ascii display list files(*.adl)

MEDM Environment variables

MEDM_WM_POSITION_POLICY

value = FRAME / USER, to define the WM positioning policy

MEDM_COLOR_RULE

to define the name of color rule file, If the env is not set, then medm reads file ColorRules in current directory.

If there is not such file, it uses default

MEDM_HELP command for Medm_HELP(mwm environment)

PSPRINTER PostScript Printer Name

EPICS_DISPLAY_PATH Path name to find display list files.

MEDM_MAIL_CMD

EPICS_MEDM_DEFAULT_MB3_DISPLAY

MEDM X resources

Default X resource file : **Medm.ad**

Custom resource setting: **~/.Xdefaults**

Release Notes at : <http://www-kryo.desy.de/~romsky/>

Add-ons for 2.4.4

- [MEDM help.](#)
- [Replace option for Related Display element.](#) *Made by Fred Vong from ANL.*
- [Save edited and non saved displays when MEDM crashes.](#)
- [Associated menu for Composite element.](#)
- [Check for identical display.](#)
- [Reading archive data into StripChart element.](#)
- [X Window resource file Medm.ad.](#)
- [Fixed bugs in version 2.4.3 from 2.3.0 which was one from what we started.](#)
- [Fixed bugs in version 2.4.4 from 2.4.3.](#)

New in the release 2.4.5

- Added an icon for display windows.
- For all monitor elements, except **Cartesian Plot** and **Valuator**, press on mouse button 3 will lead to activate a new display with macro record substitution by PV name from the element. A name of the display will be read in from system environment EPICS_MEDM_DEFAULT_MB3_DISPLAY.
- Press and just release on **Composite** elements with defined associated menu will activate an action of the first item in the associated menu. This done to make an operator's manipulations swifter.

New in the release 2.4.8

- New resource **Sensitivity PV** was added into every Control element resource list (Valuator, Text Entry, etc.).
- New resource **Display Type** was added into Display element. This made an available to have all display within modal dialog windows.
- Two Help infos describing mouse/button actions are available. Help button is on a main window menubar. One info is for actions in EDIT mode, another one is for EXECUTE mode.
- Some resources of an element can be viewed in run-time in Properties dialog invoked by Ctrl-MB3.
- Command line option `-readonly` was added. This made all Control element insensitive for actions of user.
- `medm -help` will show to user all available MEDM options.

New concept of Color Rule

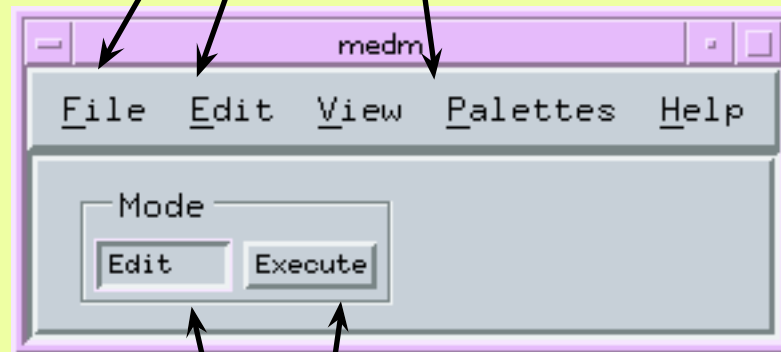
- MEDM in startup time reads ColorRules file and constructs internal list of available color rules. The file consists of an arbitrary number of color rules. Each color rule can consist of an arbitrary number (not just 16 as it was before) of rule entries (low boundary, high boundary, and color index). The ColorRules file has to keep all color rules been using in all used files. New environment MEDM_COLOR_RULE keeps a path to this file. If the env is not set, then medm reads file ColorRules in current directory. If there is not such file, MEDM uses default color rules hard coded into the source.

New resources was added to Bar element:

1. Color Mode (static/alarm/discrete)
2. Scale Type (linear/logarithm)
3. Show Bar (on/off)
4. Show Alarm Limits (on/off)
5. Show Scale (on/off)
6. Show Alarm Limits (from channel/local)
7. Low Limit, float point value
8. High Limit, float point value

MEDM Startup Screen

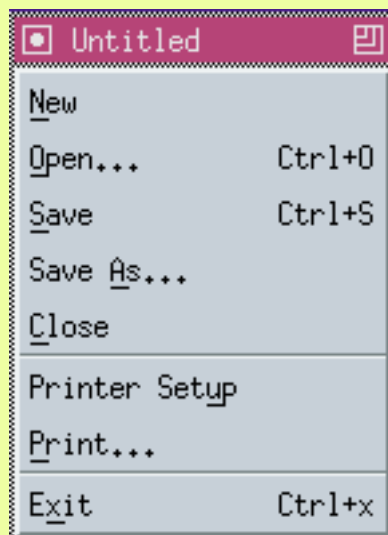
Menus of MEDM



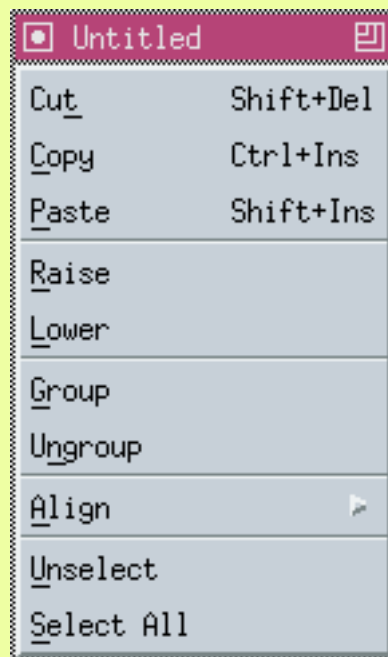
MEDM has two modes, "Edit" and "Execute". You can go and back between two modes with these buttons.

MEDM menus

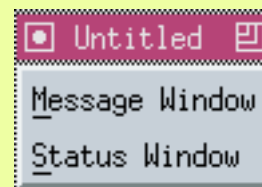
File Menu



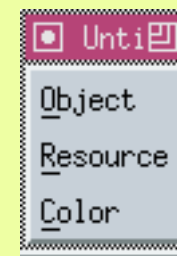
Edit Menu



View Menu



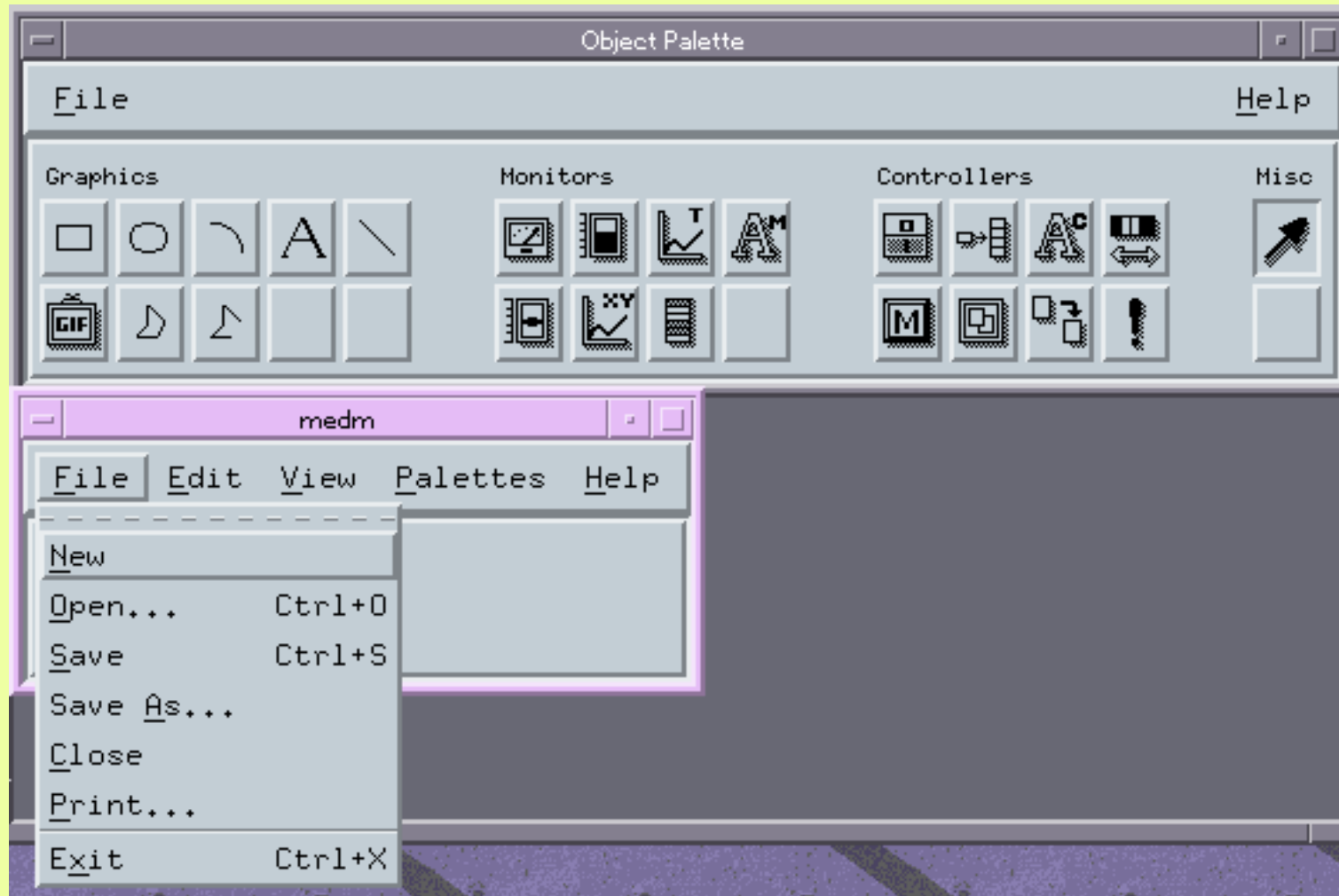
Palettes Menu



Help Menu

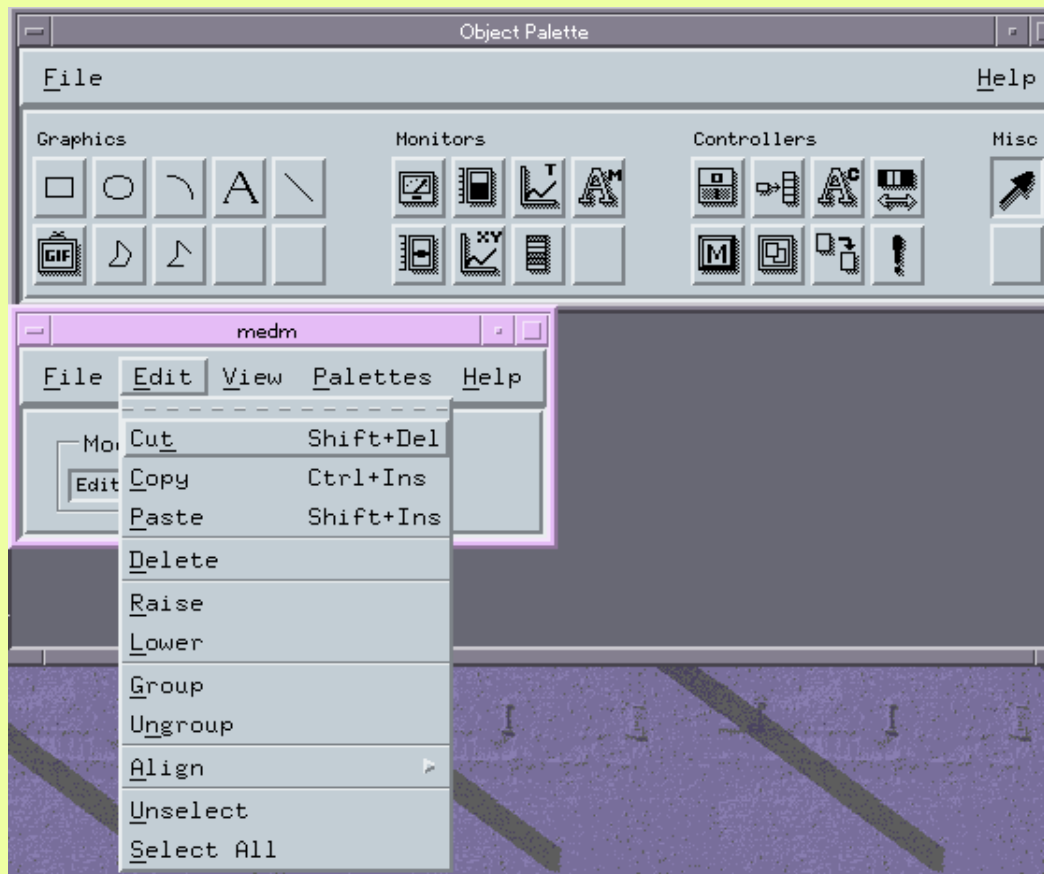


File Menu



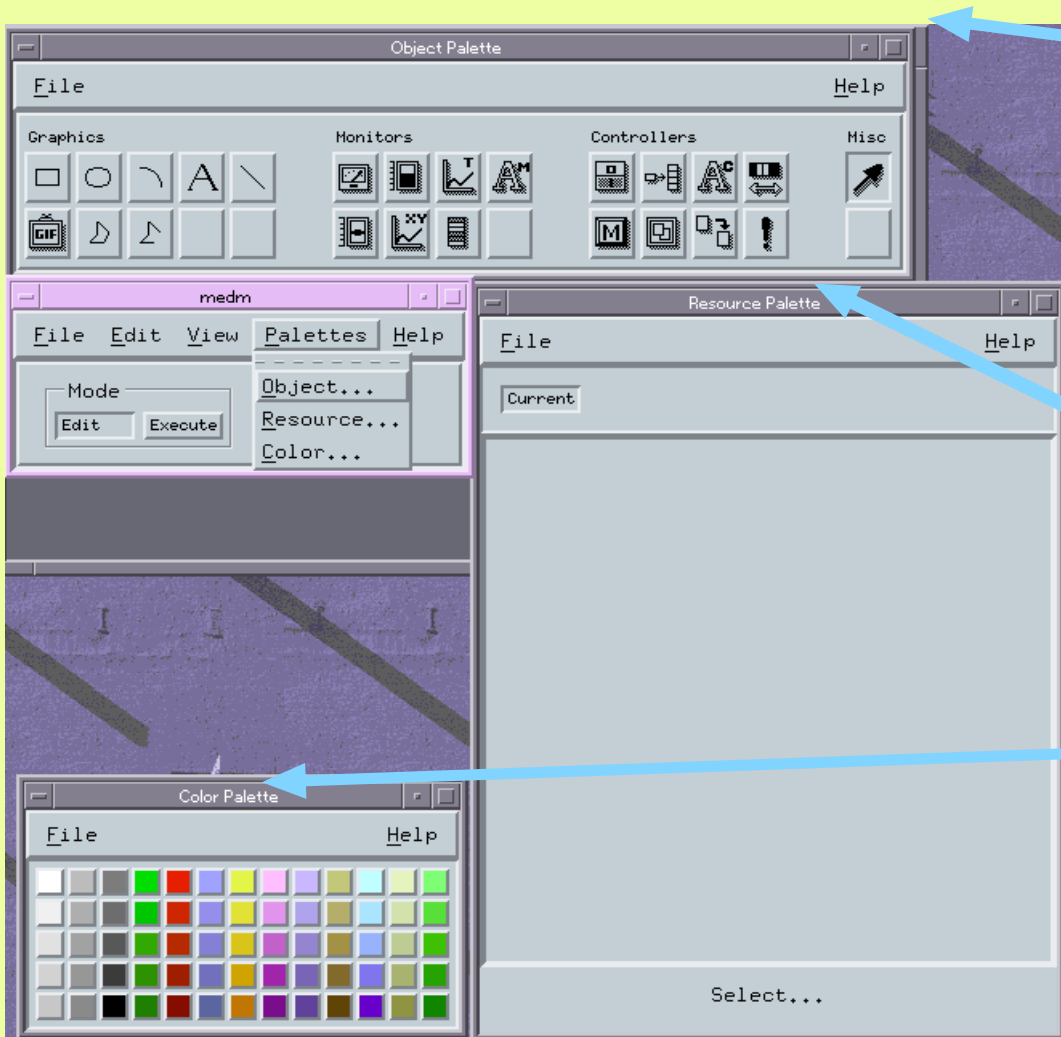
Edit Menu

Edit menu has menu items to edit objects on a display window.



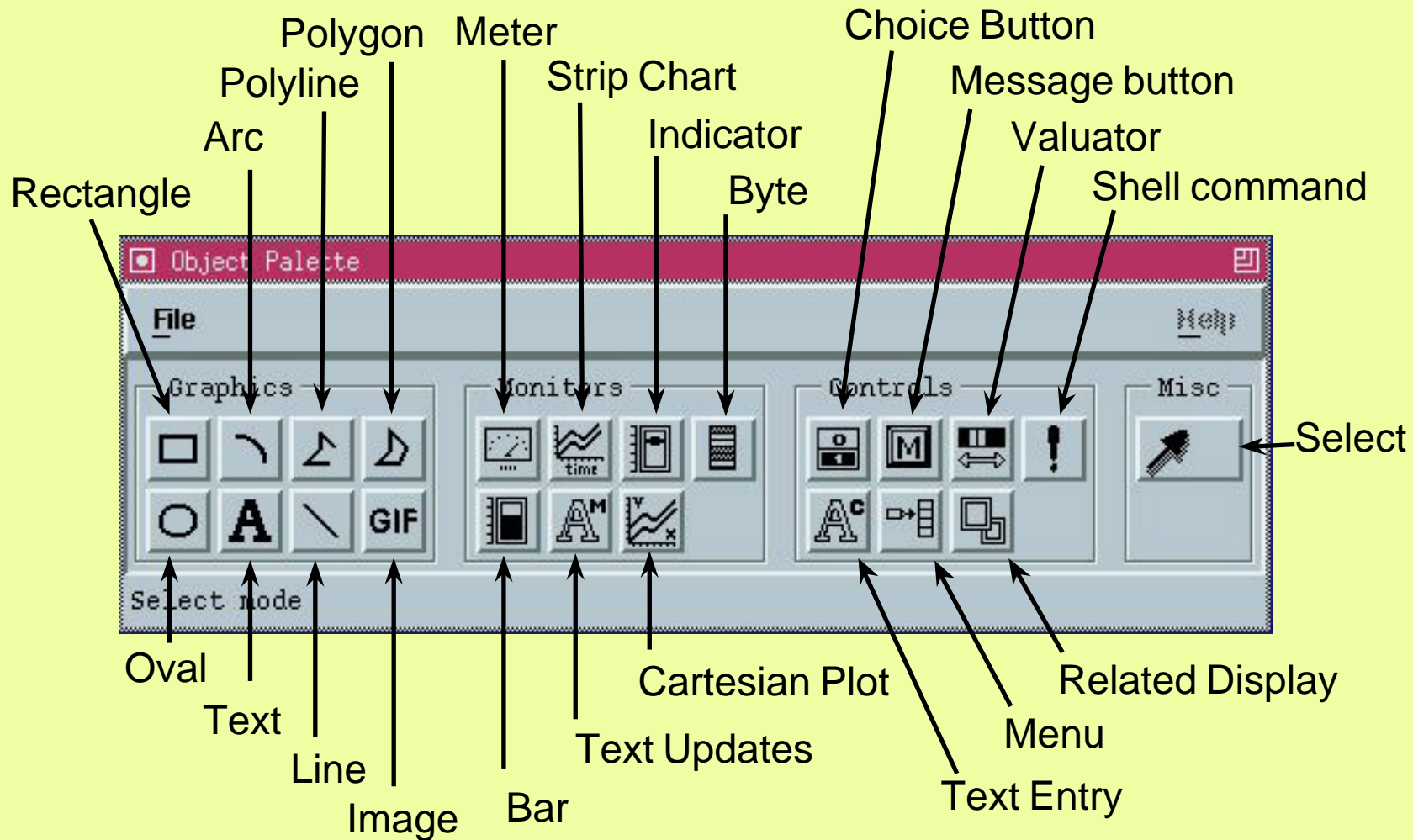
Palettes Menu

MEDM has three palettes.

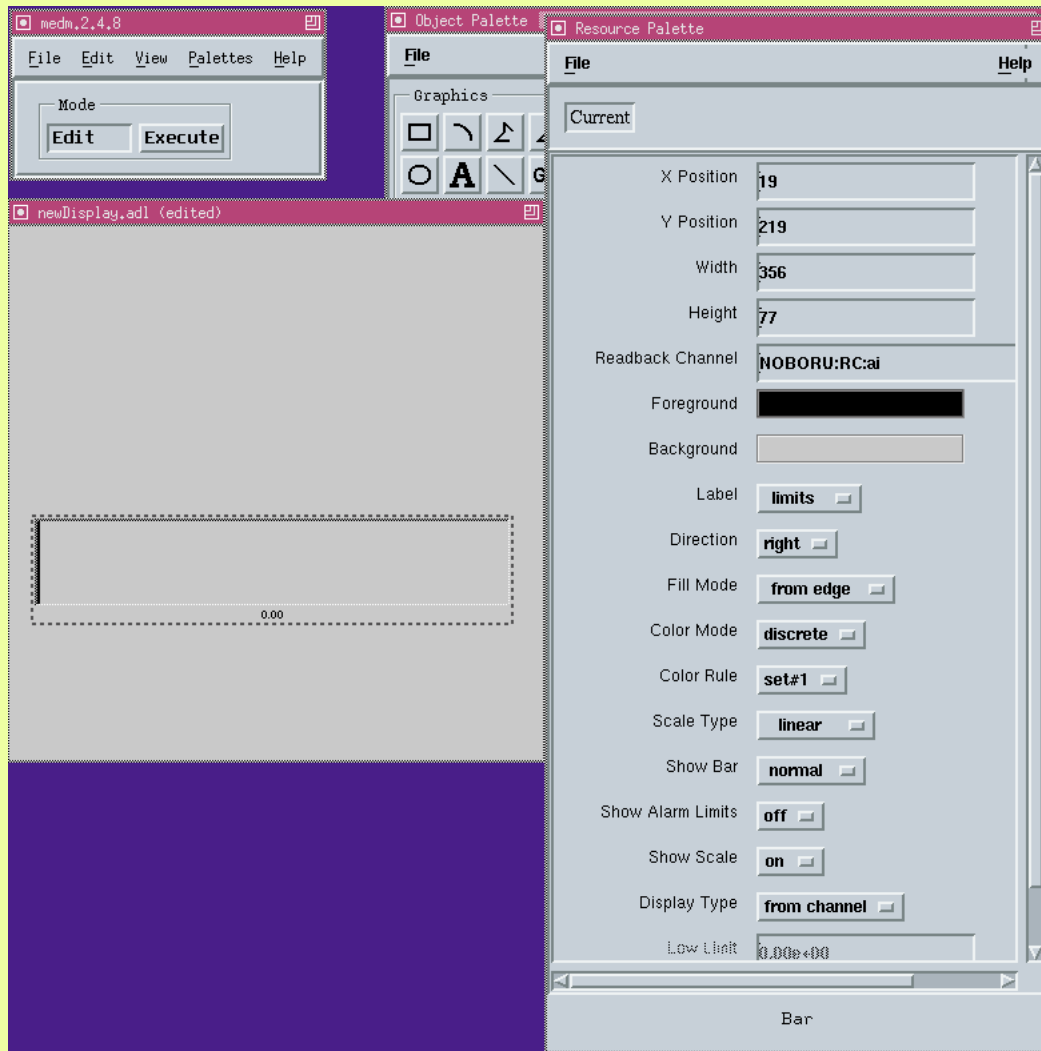


- **Objects Palettes:** Objects pallets displays objects used in MEDM. A object belongs to one of three categories, Graphics, monitors and controllers.
- **Resource Pallet:** Resource pallet displays information of a selected object in a display window.
- **Color Pallet:** Color pallet is used to change color of objects.

Object Palette



Create a object on a new display window.



1. Choose bar object in the object palette (bar object in the example).
2. Place a cross cursor within the display window.
3. Press a mouse button 1 and drag the mouse to change the size of the object.
4. release a mouse button.
5. Fill connected channel name in the resource palette.

Mouse operation in MEDM.

MB1: **select** an object (or objects) on the screen and highlight. the resource palette is updated to reflect the selected item's internal data. a drag operation under MB1 selects a group of objects on the screen (including those objects which are bounded by the selection rectangle).

Shift-MB1: **multiple-select.** a set of objects are selected for operations (such as grouping).

MB2: selected object(s) are **move**d while MB2 is depressed and deposited on button release. if no objects are currently selected, the object under the cursor when MB2 is depressed is made the current object for moving. to cancel the effect of current move, the cursor may be dragged off the current display window and the button released. this cancels the effect of the move.

Ctrl-MB2: selected object(s) are **resized** while MB2 is depressed. if no objects are currently selected, the object under the cursor when MB2 is depressed is made the current object for resizing. to cancel the effect of the current resize, the cursor may be dragged off the current display window and the button released. this cancels the effect of the resize. N.B. this mechanism performs ABSOLUTE resizing, in which all objects are extended in width and height by the magnitude of the x and y mouse motion. for PROPORTIONAL resizing in which selected objects resize consistently, the object must first be grouped.

MB3: **pop up applicable menus.**

MEDM Exercise

1. Startup MEDM.
2. Create a new display in MEDM.
3. Create Title using "Text" graphic object.
4. Create "bar" object on the display.
5. Fill "readback channel " field with a channel name "EPICS:SAMPLE:AI:n". Here <n> is a group number of your group.
6. Change attributes of the "bar" objects to,
label : limit
Color Mode: alarm
7. Create "valuator" object which is also connected to "EPICS:SAMPLE:AO:n".
8. Turn a mode of medm to "execution" and see what happen when you operate the valuator.

ALH

(EPICS Alarm Handler)

ALH:

- is an EPICS OPI tool(CA Client)
- displays Alarm Status(Severity) of EPICS records specified in the configuration file(.alhConfig).
- has a Graphical User Interface based on Motif.
- notifies alarm status to the operator by,
 - Color,
 - Sound(Beep),
 - Any Unix command.
- writes alarm/operator log into the files.

ALH sample display

MEDM

The screenshot displays the ALH sample display interface, which is divided into two main sections: MEDM (top) and ALH (bottom).

MEDM Section: The top window, titled "ALHsample.adl", shows a grid of five monitors. The second monitor in the second row is highlighted in red, indicating an alarm. Below the monitors are three columns labeled "MAIN", "Group1", and "Group2". Under each column, there are two rows of circular indicators. The "CALC" row has red circles under "MAIN" and "Group2", and a green circle under "Group1". The "ALH SEVRPV" row has red circles under "MAIN" and "Group2", and a green circle under "Group1". A "medm.2.4.8" dialog box is open, showing "Mode" with "Edit" and "Execute" buttons.

ALH Section: The bottom window, titled "Alarm Handler", shows a list of alarm channels. The "MAIN" channel is selected and highlighted in red. Other channels include "group1", "group2", "ALHsample:group2_alarm", "ALHsample:monitor6", "ALHsample:monitor7", "ALHsample:monitor8", "ALHsample:monitor9", and "ALHsample:monitorA". A smaller "Alarm Handler" dialog box is open, showing a red bar with the text "MAIN".

At the bottom of the interface, there is a status bar with the following text:

Mask <CDATL>: <Cancel,Disable,noAck,noackT,noLog>

Group Alarm Counts: <INVALID,MAJOR,MINOR,NOALARM>

Channel Alarm Data: Current(Status,Severity),Highest Unack(Status,Severity)

Filename: ALHsample.alhConfig

ALH

ALH Group Window displays Tree of Alarm Group status.

Channel/Sub group Status

Highest Alarm Status in the sub-tree

Operator Acknowledgment Buttons

Mask

Launch Unix Command

Guidance Button:

Severity Code

Alarm Handler

MAIN

File Action View Setup Help

MAIN <--AT--> (0,2,0) **R R** ALHsample:group2_alarm **R** **G** **P** <-----> <LINK...

group1 <--AT-->

R R group2 <---T--> (0,2,0) **R R** ALHsample:monitor6 **G** **P** <---T-->

ALHsample:monitor7 **G** **P** <---T--> <HIHI,MAJOR

ALHsample:monitor8 **G** **P** <---T-->

ALHsample:monitor9 **G** **P** <---T-->

ALHsample:monitorA **G** **P** <---T-->

Mask <CDATL>; (Cancel,Disable,noAck,noackT,noLog) SilenceForever

Group Alarm Counts: (INVALID,MAJOR,MINOR,NOALARM) SilenceCurrent

Channel Alarm Data: Current(Status,Severity),Highest Unack(Status,Severity)

Filename: ALHsample,alhConfig

MASK

- **Add/Cancel Monitors**
 - **Enable/Disable Alarm Events**
 - **Ack/NoAck Alarm Changes**
 - **Ack/NoAck Transient Alarms**
 - **Log/NoLog Alarm**
-
- **Force Process Variable to change Mask dynamically.**
 \$FORCEPV <ForcePVName> <ForceMask> <Force Value> <resetValue>

ALH configuration file

```
GROUP      NULL      MAIN
$ALIAS     ROOT
$COMMAND   ShowHelp
$SEVRPV    ALHsample:status
$SEVRCOMMAND  UP_MAJOR echo "Help me!" | mail manager
$GUIDANCE
Put Your Guidance text here.
$END
GROUP      MAIN      group2
$SEVRPV    ALHsample:group2status
$SEVRCOMMAND  UP_ANY echo "GROUP2 Alarm up_any"
CHANNEL    group2 ALHsample:group2_alarm      --A--
$COMMAND   medm related_display.adl
CHANNEL    group2 ALHsample:monitor6      ---T-
CHANNEL    group2 ALHsample:monitor7      ---T-
CHANNEL    group2 ALHsample:monitor8      ---T-
```

ALH: Communication to the others

- Put new status into the EPICS Record(Process Variable)
- Launch Unix Program

```
$SERVPV <ServPVName>
```

The line starting with \$SERVPV is optional. It is required only when a user wants to write the severity value of the group or channel to a process variable.

\$SEVRCOMMAND <severtyChangeValue> <any valid Unix command syntax >

The line starting with \$SEVRCOMMAND is optional. It is required only when a user wants to start a Unix process when the alarm severity value for a group or channel changes. A single group or channel may have multiple \$SEVRCOMMAND lines. This line defines the change in the severity necessary to start the process and defines the process to be started. Valid severity change values are -

```
UP_INVALID   UP_MAJOR   UP_MINOR   UP_ANY
DOWN_MAJOR   DOWN_MINOR  DOWN_NO_ALARM  DOWN_ANY
```

\$STATCOMMAND <alarmStatusStringValue> <any valid Unix command syntax >

The line starting with \$STATCOMMAND is optional and valid only for an alarm channel. It is required only when a user wants to start a Unix process when the alarm status for the channel becomes a specified value. A single channel may have multiple \$STATCOMMAND lines. This line defines the status value necessary to start the process and defines the process to be started. Valid alarmStatusString values are defined in the alarmString.h header file. Example alarm status string values are -

```
NO_ALARM READ WRITE HIHI HIGH READ_ACCESS  
LOLO LOW STATE COS COMM WRITE_ACCESS  
TIMEOUT HWLIMIT CALC SCAN LINK  
SOFT BAD_SUB UDF DISABLE SIMM
```

\$ALARMCOUNTFILTER <inputCount> <seconds>

The line starting with \$ALARMCOUNTFILTER is optional. It is required only when a user wants the alarm handler to filter the registration of alarms for an alarm channel. This line defines alarm count and seconds required for alarm registration. To register as an alarm, an alarm channel must remain in alarm state for more than <inputSeconds> seconds OR the alarm channel must enter into an alarm state from a no-alarm state more than <inputCount> times within <inputSeconds> seconds.

```
%alh [-c] [-f filedir] [-l logdir] [-a alarmlogfile]
[-o opmodlogfile] [-m alarmlogmaxrecords] [Xoptions]
[configfile]
```

options:

configfile	Alarm configuration filename
-c	Alarm Configuration Tool mode
-f filedir	Directory for all files
-l logdir	Directory for log files
-a alarmlogfile	Alarm log filename
-o opmodlogfile	OpMod log filename
-m maxrecords (default 2000)	alarm log file max records
-v	DEBUG mode

Environment Variable:

ALARMHANDLER	config file directory
--------------	-----------------------

MEDM: Color Rule File

colorRule <setname1>;

<lower value1> <uppler value1> <index1>;

<lower value2> <uppler value2> <index2>;

.....

colorRule <setname2>;

<lower valueA> <uppler valueA> <indexA>;

<lower valueB> <uppler valueB> <indexB>;

.....