For: jba
Printed on: Fri, Nov 5, 1993 11:15:05
From book: xmca
Document: titlePage
Last saved on: Fri, Nov 5, 1993 09:01:33
Document: preface
Last saved on: Thu, Sep 17, 1992 10:17:25
Document: toc
Last saved on: Fri, Nov 5, 1993 09:24:37
Document: xmca
Last saved on: Fri, Nov 5, 1993 09:18:50
Document: Appendix
Last saved on: Mon, Sep 14, 1992 14:29:33

X–window Monitoring of Channel Access
User's Guide

Ben–chin K. Cha
and
Fred Vong
November 5, 1993

DRAFT

The Controls and Computing Group
Accelerator Systems Division
Advanced Photon Source
Argonne National Laboratory

PREFACE

This document serves as the XMCA user's guide for using X–window to monitor or display the information of IOC. It describes the user interface and the functionalities of XMCA. The user interface for the single channel monitoring program, PROBE, is also described in this document.

XMCA provides easy access of IOC devices information through an X–window mouse–driven operating interface. It dynamically loads the desired IOC devices into the core, lets users select directly from the loaded list, and eliminates the pain of interactively entering the device names. It provides the X–window texture display feature of detailed operating information of IOC devices

This is a preliminary version of XMCA. When new important features are identified, XMCA may undergo further upgrading.

The contributions of Bob Daly and Ned Arnold are greatly appreciated.

CHAPTER 1 INTRODUCTION 6
1.1 Purpose 6
1.2 Interface to IOC 6
1.3 Limitation 6

CHAPTER 2 XMCA OVERVIEW 8
2.1 Summary 8
2.2 XMCA Files 9
2.2.1 Input Static Database File 9
2.2.2 Input IOC List File 9
2.2.3 Backup File 9
2.2.4 Output Report File 9
2.3 Display Window Selection Types 9
2.3.1 Single Channel Selection 9
2.3.2 Multiple Channel Selection 10
2.3.3 Whole List Selection 10

CHAPTER 3 XMCA MAIN WINDOW 11
3.1 Main Window Layout 11
3.2 Main Window Menu Bar 11
3.2.1 Exit Menu 11
3.2.2 LoadDB Menu 11
3.2.3 LoadList Menu 12
3.2.4 SubList Menu 12
3.2.5 TypeList Menu 12
3.2.6 SubString Menu 12
3.2.7 WholeList Menu 12
3.2.8 Help Menu 12
3.3 Work List Message 12
3.4 Single Selection Buttons 12
3.4.1 caGet Button 13
3.4.2 caPut Button 13
3.4.3 caProbe Button 13
3.4.4 caGetWF Button 13
3.4.5 caGetDB Button 13
3.5 Multiple Selection Button 13
3.5.1 Monitor Value Button 13
3.6 Whole List Buttons 13
3.6.1 Alarm_All Button 13
3.6.2 Info_All Button 13

CHAPTER 4 SINGLE SELECTION WINDOWS 14
4.1 CaGet Window Layout 14
4.2 CaPut Window Layout 14
4.3 CaProbe Window Layout 15
4.4 CaGetWF Window Layout 15
4.5 CaGetDB Window Layout 16
4.6 Single Channel Selection Window Layout 16

CHAPTER 5 MULTIPLE SELECTION WINDOWS 18
5.1 Multiple Channel Selection Window 18

5.1.1 Help Button 18
5.1.2 Cancel Button 19
5.1.3 Save Button 19
5.1.4 Ok Button 19
5.2 Selected Channels Info Window 19
5.2.1 Scrolled Window Contents 19
5.2.2 Close 20
5.2.3 Update 20
5.2.4 Start 20
5.2.5 Stop 20
5.2.6 Print 20
5.2.7 Cntl 20
5.3 Multiple Probe Window 20
5.3.1 Put Button 21
5.3.2 Get Button 21
5.3.3 Close 21

CHAPTER 6 WHOLE LIST WINDOWS 22
6.1 Whole List Alarm Info Window 22
6.1.1 Scrolled Alarm Window Contents 22
6.1.2 Close 22
6.1.3 Update 22
6.2 Whole List Info All Window 23
6.2.1 Scrolled Info Window Contents 23
6.2.2 Close 24
6.2.3 Update 24
6.2.4 Start 24
6.2.5 Stop 24
6.2.6 Backup 24
6.2.7 Restore 24
6.2.8 Print 24

CHAPTER 7 INVOKING XMCA 26
7.1 Setup Requirement 26
7.2 Running XMCA 26
7.3 Remote Usage 26

CHAPTER 8 XMCA FILES 27
8.1 Default File Type Settings 27
8.2 XMCA List File Description 27
8.2.1 Creating Complete IOC List File 27
8.2.2 Example List File 28
8.3 XMCA Backup File Description 28
8.3.1 Example Backup File 29
8.4 XMCA Output Report 29

CHAPTER 9 SYSTEM REQUIREMENTS 30
9.1 Pre And Post Conditions 30
9.2 Channel Access Monitor Events 30
9.3 Channel Access Get / Put Values 30

9.4 Special Hardware Required 30
9.5 Special Software Required 30

CHAPTER 10 PROBE USER INTERFACE 32
10.1 What is Probe 32
10.2 Getting Started 32
10.3 Layout of the Probe 32
10.4 Connecting a Process Variable 33
10.5 Monitoring a Process Variable 33
10.6 Changing the Value of a Process Variable 33
10.7 Getting the Record Setting 34
10.8 Getting the Run Time History 35
10.9 Changing the Display Format 35
10.10 Exiting the Program 35

APPENDIX 35
Table 1 35
Table 2 36

CHAPTER 1 INTRODUCTION

The X–window Monitoring of Channel Access (XMCA) program is a Motif application which displays or monitors single or multiple channels of any Input / Output Controller (IOC) . XMCA is available on the Unix Sun4 platforms. The purpose of this document is to provide the information required for running XMCA from a user's point of view.

Chapter 1 gives the introduction and purpose of XMCA and an overview of XMCA is contained in Chapter 2. The XMCA Main Window is described in Chapter 3, single channel windows are in Chapter 4, multi–channel windows are described in Chapter 5, and whole list information windows are described in Chapter 6. Invoking XMCA and setup requirements are described in Chapter 7, while XMCA channel(or process variable) list files, backup files, and report files are presented in Chapter 8. System requirements are listed in Chapter 9, and user interface for the single channel monitoring program, **probe**, is given in Chapter 10.

1.1 Purpose

The primary functions of the XMCA are:

* Provide flexible loading of any database file or IOC device file based on user selection.

* Allow monitoring of single or multiple channels with Motif window display system.

* Easily display and generate an information report of the loaded IOC.

* Provide a convenient IOC debug tool on Sun4.

* Provide the feature of simple backup / restore of monitored channels.

1.2 Interface to IOC

XMCA communicates with the IOC through channel access function calls.

1.3 Limitation

A user can dynamically load any database / IOC file into XMCA. Each IOC file should consist of a list of type and name for channels (i.e. process variables) defined in any DCT database. To allow

XMCA to display detailed operating information for the whole list, the number of process variable names in each database / IOC file can not exceed 2000. The maximum subgroup allowed for XMCA is 500. Automatic monitoring of the whole list is allowed only when the number of process variable names of the IOC file is less than 50.

CHAPTER 2 XMCA OVERVIEW

2.1 Summary

XMCA is a Motif X–window application running on an Operator Interface (OPI). The philosophy is that an operator has no need to use the keyboard except to enter a new value for a channel device. The left mouse button is used to select any item. Appropriate on–line help is always available.

The XMCA application has the following attributes:

* The selection options are grouped into 3 categories: single channel, multiple channels, and whole list.

* The list of IOC channels is dynamically loaded into the memory through an external file. A user can load any IOC list or static database on the fly.

* The menu bar of the main window allows a user to load IOC channel names, select the work list, and obtain on–line help.

* The single channel category has five push buttons: *caGet* , *caPut*, *caProbe, caGetWF,* and *caGetDB*. Each push button results in a popup display window.

* The *Monitor Value* button pops up a multiple selection box and subsequent display windows.

* The whole list selection has two types of display windows: the alarm characteristics displaying window and the complete information window.

* Each popup window is managed separately by its own close button. Before bringing up another similar type of window the old one must be closed explicitly by the user since only one can be active.

* Multiple channels or whole list display windows always give the control operating ranges and information. If the size of the whole list is less than 50, a user can monitor the whole list.

* The whole list display window provides the option of generating a hard copy of the report from the scrolled window list.

2.2 XMCA Files

Four types of files are used by the XMCA. They are described as follows.

2.2.1 Input Static Database File

The input database file consists of the complete set of IOC channel names defined in a static database. The file name has ".database" as the suffix. Which is automatically created by the DCT (EPICS Database Configuration Tool) or the ATDB tool.

2.2.2 Input IOC List File

The input IOC list file specifies a list of the IOC channel names defined in a database. The IOC file name has ".ioc" as the suffix. The permission mode should be set to read only for the master IOC list file, this will avoid the user accidentally saving the selected sublist to the master file. Each line in the file should contain record type and name for a single process variable only. The list file should be sorted in alphabetic order. The XMCA saved sublist also has the type of ".ioc", but it has the write/read permission mode.

2.2.3 Backup File

The backup file contains a list of pairs of backup channel (process variable) name and saved value. The backup file name has ".bk" as the suffix. The backup file is an ASCII file that can be displayed on any terminal or sent to a line printer. The master backup file should be set to read only. This will prevent XMCA user accidentally rewrites the backup data to the master backup.

2.2.4 Output Report File

The output report file named " /tmp/xmca.InfoOut " is saved on the workstation where XMCA is invoked. This file contains the snapshot summary information report for the whole list. This file is formatted for 130–character–wide line printer output. It is automatically generated and printed by XMCA when the print option is selected.

2.3 Display Window Selection Types

The XMCA main window contains three display window selection types: *Single Channel*, *Multiple Channels*, and *Whole List*.

2.3.1 Single Channel Selection

The single channel selection consists of five push buttons: *caGet, caPut, caProbe, caGetWF,* and *caGetDB.* The *caGet* button pops up a get value selection and display window. The *caPut* button pops up a put value selection and display window. The *caProbe* button pops up a probe selection window which will automatically invoke the PROBE program. The *caGetWF* button pops up a wa-

veform record selection and display window. The *caGetDB* button pops up a database record selection and displaying window.

2.3.2 Multiple Channel Selection

The *Monitor Value* selection button pops up a multiple selection box, which allows a user to select more than one channel to be monitored. The selected monitored list can be saved as a new IOC list file, which can be loaded directly into XMCA later. The popup channel monitoring window displays detailed information about monitored channels. This monitor window also provides an option of popping up the multiple probe window which allows a user to get values from IOC or put values to IOC for the selected channels.

2.3.3 Whole List Selection

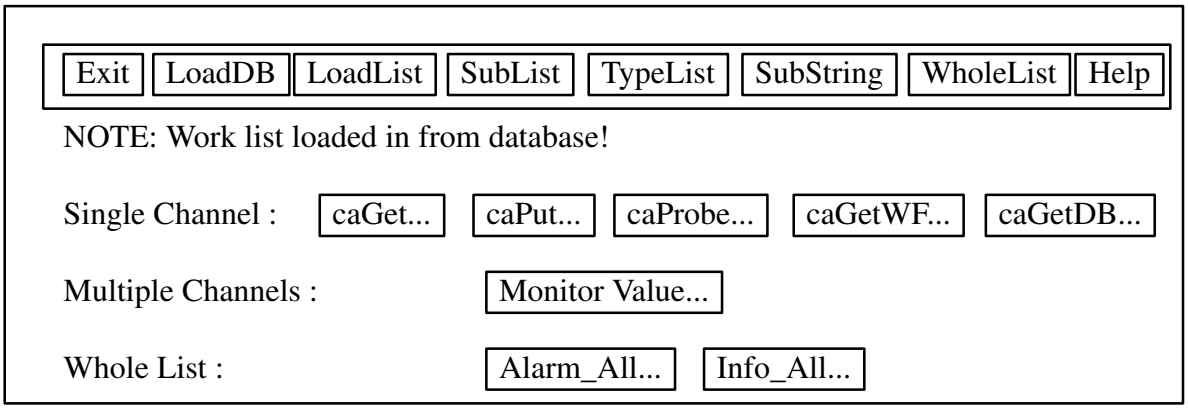
XMCA allows two types of whole list display windows of IOC channels. One is the alarm characteristics display window which displays current values, alarm status and severity, and alarm conditioning settings of all channels. The other is the information window which displays IOC control information (values, alarm status, alarm severity, units, operating ranges, and alarm conditioning settings) of all channels of the current loaded work list.

If the number of channels in the whole list is less than 50, this information window allows a user to monitor the whole list. This window also allows a user to generate a printed copy of the content of the scrolled window.

CHAPTER 3 XMCA MAIN WINDOW

3.1 Main Window Layout

The XMCA main window provides the user with various ways of obtaining detailed control information from channel access about any channel or process variable across the IOC network. XMCA allows a user flexibly to load any IOC list file or database file into the memory. By simple push button actions a user can quickly access IOC control operating information on the Unix system. The layout of the main window is shown below. This window contains a menu bar, a status of the work list information message, and three groups of selection buttons: *Single Channel, Multiple Channels,* and *Whole List*.



3.2 Main Window Menu Bar

The main window menu bar consists of eight menus. They are described in the following sections.

3.2.1 Exit Menu

The *Exit* menu allows the user to leave the XMCA application. Selecting the *Ok* button confirms the termination. It clears all the channel access events, closes all the windows, and stops the XMCA program.

3.2.2 LoadDB Menu

The *LoadDB* menu allows the user to load a database file into memory. XMCA uses the standard Motif file selection box to load the database file. Before querying static database fields the database process variable names must be loaded into memory by selecting this option. Normally this is the first button depressed by the user.

3.2.3 LoadList Menu

The *LoadList* menu allows the user to load an IOC list file into memory. XMCA starts with an empty list. The IOC list file allows mixed of process variables from different databases. Before using the scrolled list of channels the IOC process variable names must be loaded into memory by selecting this button. Normally *LoadDB* or *LoadList* is the first button depressed by the user.

3.2.4 SubList Menu

Sometimes related channels are named with same prefix group name. The *SubList* option allows the user to select channels with the same prefix group name. The total number of subgroups can not exceed 500.

This menu gives the user a choice of selecting a group of devices as the working list. The subgroup list is automatically generated by XMCA and depends on the IOC file loaded into memory. In order to use this option, the IOC device name has to use ' : ' as the group hierarchy separator and the list file should be sorted before loading into XMCA (as described in Chapter 8). The default work list is the whole list. If the SubList Menu is non–empty, the last entry is always ' WHOLE_LIST'. If the last entry is selected, the whole IOC list is used. If the other item is selected, only the devices matches the items are selected.

3.2.5 TypeList Menu

This menu allows the user termporarily to reduce the work list to the specified database record type. The *TypeList* menu consists of the items currently supported by the DCT database tool.

3.2.6 SubString Menu

This menu allows the user to select all the records which contain the user spcified search string as a new work list.

3.2.7 WholeList Menu

This menu allows XMCA to use the originally loaded in database or IOC list as the work list.

3.2.8 Help Menu

The help menu provides on–line help information about each push button of the main window.

3.3 Work List Message

A work list status message is shown on main window which reflects the current status of the work list. A message of "NOTE: No work list yet!" is displayed when XMCA is first invoked. This message will be updated according to the item selected from the menubar. For example, the message will be replaced by " NOTE: Work list loaded in from ioc file!" if *LoadList* item is selected from the menubar. The message will be replaced by "NOTE: Work list loaded in from database!" if *LoadDB* item is selected from the menubar.

3.4 Single Selection Buttons

For single channel selection, there are five push buttons: *caGet, caPut, caProbe, caGetWF,* and *caGetDB*.

3.4.1 caGet Button

To get the float or string value from a desired channel device use the *caGet* button. The *caGet* button pops up the get value window.

3.4.2 caPut Button

To put a real or string value to a desired channel device use the *caPut* button. The *caPu*t button pops up the put value window.

3.4.3 caProbe Button

To probe detailed information for a desired channel device use the *caProbe* button. The *caProbe* button pops up the probe channel dialog window.

3.4.4 caGetWF Button

To get array values for a desired waveform channel use the *caGetWF* button. The *caGetWF* button pops up the array value display dialog window.

3.4.5 caGetDB Button

To get database record information for a desired channel use the *caGetDB* button. The *caGetDB* button pops up the static database fields display dialog window.

3.5 Multiple Selection Button

XMCA allows a user to monitor multiple channels of IOC, get an array of values from IOC, or write new array of values to IOC.

3.5.1 Monitor Value Button

To select several channels to monitor / put / get use the *Monitor Value* button. The *Monitor Value* button first pops up the multiple channel selection box and then the monitored scrolled window for the selected channels. A *Cntl* menu of the monitored window can pop up the multiple probe control window which allows a user to get / put multiple values to IOC.

3.6 Whole List Buttons

For a whole list selection, there are two push buttons: *Alarm_All* and *Info_All*.

3.6.1 Alarm_All Button

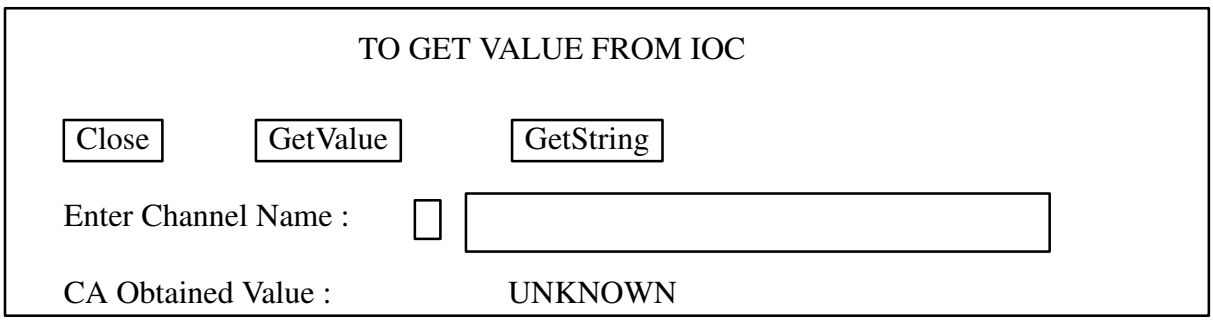
To list the alarm characteristics for all the channels in the current work list use the *Alarm_All* button. The *Alarm_All* button pops up an alarm information window for all channels.

3.6.2 Info_All Button

To list complete information for all the channels from the current work list use the *Info_All* button. The *Info_All* button pops up an information window and displays all the channels from the current working list.

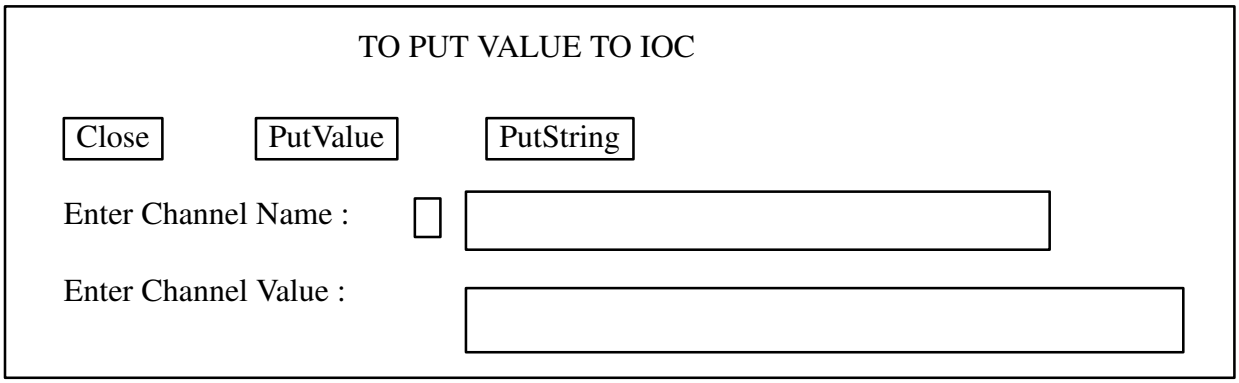
CHAPTER 4 SINGLE SELECTION WINDOWS

4.1 CaGet Window Layout



The caGet window layout is shown above. It contains four push buttons: *Close*, *GetValue*, *GetString* and an unlabeled channel selection button. It also contains a text field for entering the channel name. A user can type in the channel name directly or use the channel selection button, located at the left–hand side of the text field, to choose the name from the work list. Depressing the *GetValue* button accepts the name in the text field, gets the current value form the IOC, and displays the obtained value in engineering form. Depressing the *GetString* button returns the value in a string form instead of engineering number form. A value of 'UNKNOWN' is displayed for device which is not connected. Pressing the *Close* button closes this window layout.

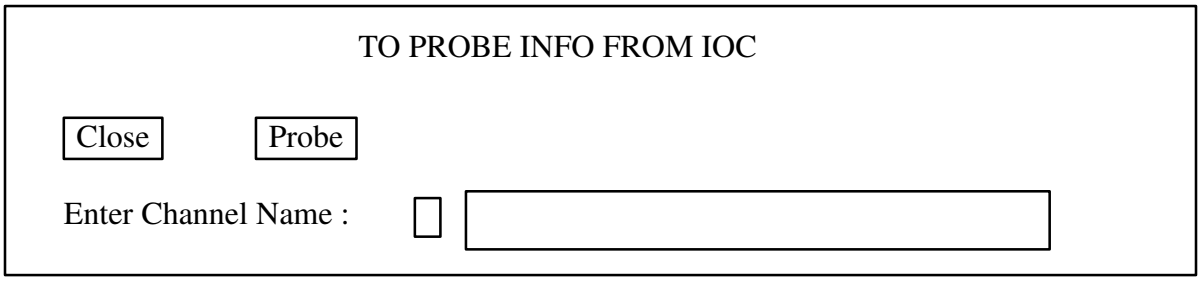
4.2 CaPut Window Layout



The caPut window layout is shown above. It contains 4 push buttons: *Close*, *PutValue*, *PutString*, and an unlabeled channel selection button. It also contains 2 text fields for entering name and new value. A user can type in the device channel name directly or use the selection button, located at the

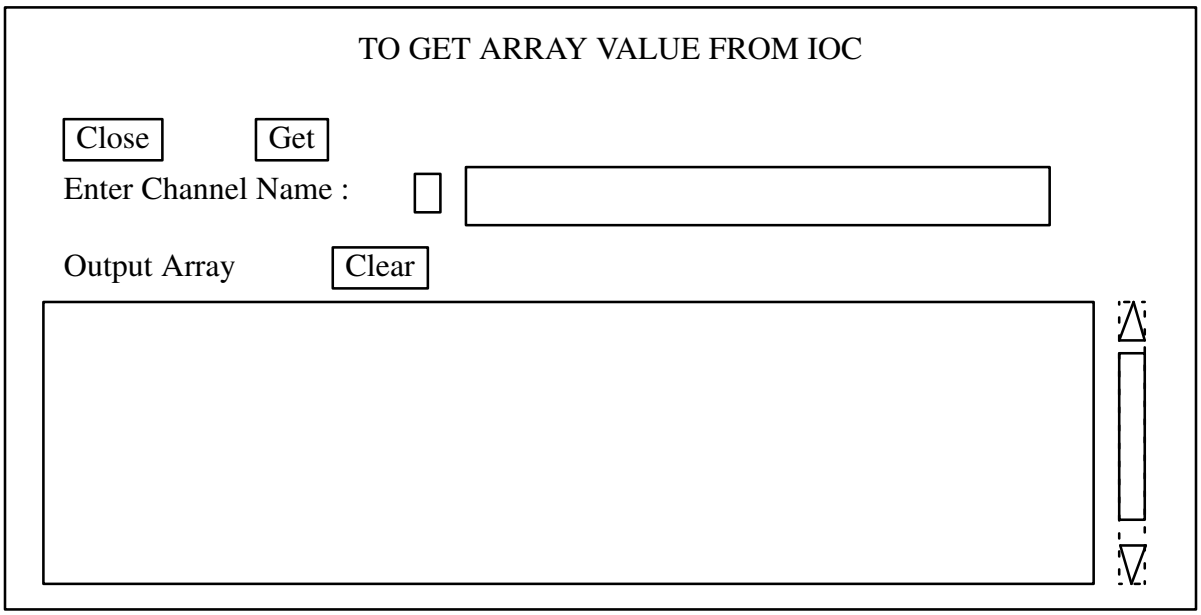
left–hand side of the channel name text field, to choose the name from the loaded list. Depressing the *PutValue* button accepts the name and value in the text fields and puts the value to the IOC. For some device, the native value field defined as string type, the *PutString* button should be used for putting a string value to the specified field. Pressing the *Close* button closes this window layout.

4.3 CaProbe Window Layout



The caProbe window layout is shown above. It contains three push buttons: *Close*, *Probe*, and an unlabeled channel name selection button. It also contains a text field for entering a channel name. A user can type in the name directly or use the selection button, located at the left–hand side of the name text field, to choose the name from the loaded list. Depressing the *Probe* button accepts the name from the text field and then starts a child process **probe**. A user can start as many probe processes as desired. Pressing the *Close* button closes this window dialog. The user interface for the **probe** program is described in Chapter 10.

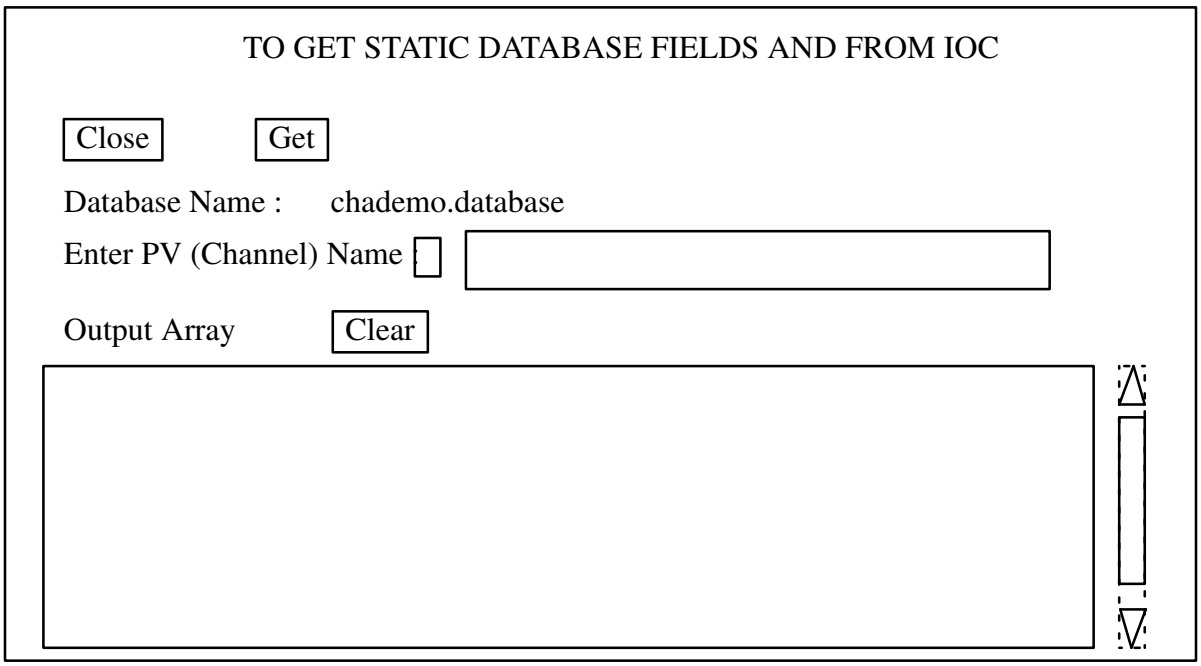
4.4 CaGetWF Window Layout



The caGetWF window layout is shown above. It contains four push buttons: *Close*, *Get, Clear*, and an unlabeled channel name selection button. It also contains a text field for entering a channel name

and a scrolled text window for displaying the array of output values obtained from the IOC. A user can type in name directly or use the selection button, located at the left–hand side of the name text field, to choose the name from the loaded list.

4.5 CaGetDB Window Layout



The caGetDB window layout is shown above. It contains four push buttons: *Close*, *Get, Clear*, and an unlabeled channel name selection button. It also contains a text field for entering a channel name and a scrolled text window for displaying the output static database field values obtained from the IOC. A user can type in name directly or use the selection button, located at the left–hand side of the name text field, to choose the name from the loaded list.

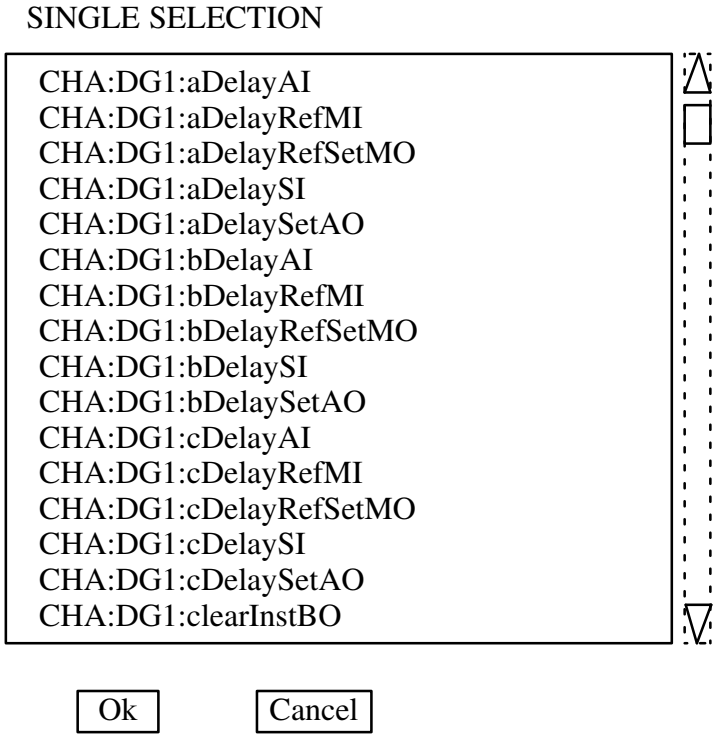
The number of fields returned depends on the type of the record defined in database. Depressing the *Get* button accepts the name from the text field and then outputs the obtained values in the scrolled window area. For each DCT configurable field a value is returned. For the field name marked with '^' indicates that field can not be accessed by the channel access. Channel access returned values which are different from the static database values are also listed. Depressing the *Clear* button clears the output scrolled window. Pressing the *Close* button closes this window dialog.

4.6 Single Channel Selection Window Layout

The device selection button pops up a single channel selection window. This window contains a scrollable selection list and two buttons: *Ok*, and *Cancel*. The single channel selection window layout is shown below.

The *Cancel* button closes the window and aborts the selection dialog.

The selected item is highlighted. The *Ok* button accepts the highlighted item as the selection, closes the selection dialog, and writes the selected item name into the Enter Channel Name field.



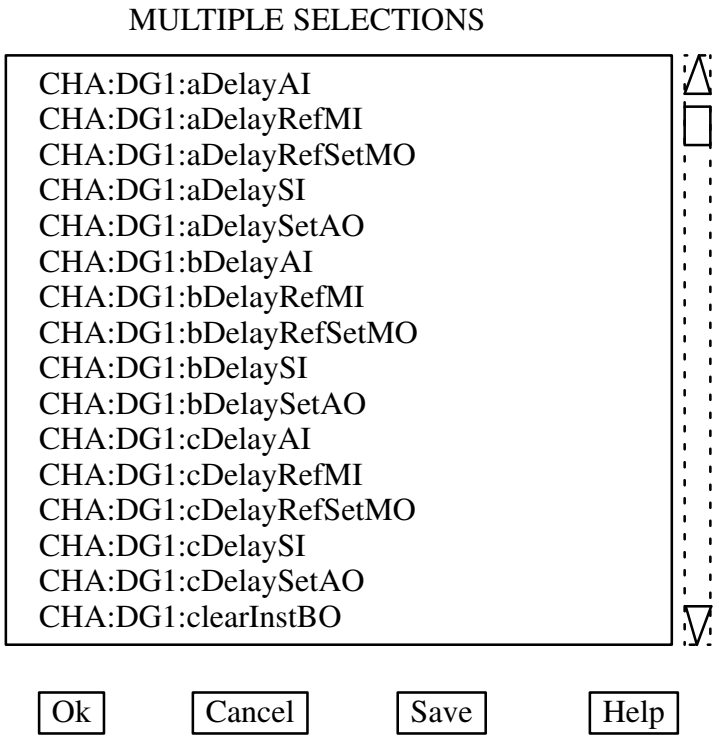
The operation of scrolling a list is controlled by the up arrow, the down arrow, and the scroll bar. The size of the scroll bar indicates the ratio of items displayed to the whole list of items. The position of the scroll bar indicates the position of the first item in the window relative to the first item of the whole list. If the scroll bar is at the top, it indicates that the first item in the window is the first item of the list. Conversely, if the scroll bar is at the bottom, it indicates that the last item displayed is the last item of the list. Dragging the scroll bar up or down scrolls the list up or down continuously. Clicking on the up arrow or down arrow scrolls one line up or down.

CHAPTER 5 MULTIPLE SELECTION WINDOWS

There are two windows associated with the *Monitor Value* button: *Multiple Channel Selection* and *Selected Channels Info*.

5.1 Multiple Channel Selection Window

The *Monitor Value* button pops up the multiple channel selection window. This window contains a scrollable selection list and 4 buttons: *Ok*, *Cancel, Save* and *Help*.



The operation of scrolling a list is controlled by the up arrow, the down arrow, and the scroll bar. The size of the scroll bar indicates the ratio of items displayed to the whole list of items. The position of the scroll bar indicates the position of the first item in the window relative to the first item of the whole list. If the scroll bar is at the top, it indicates that the first item in the window is the first item of the list. Conversely, if the scroll bar is at the bottom, it indicates that the last item displayed is the last item of the list. Dragging the scroll bar up or down scrolls the list up or down continuously. Clicking on the up arrow or down arrow scrolls one line up or down.

5.1.1 Help Button

The *Help* button pops up a help information window which describes the function of multiple item selection.

5.1.2 Cancel Button

The *Cancel* button closes the window and aborts the selection dialog.

5.1.3 Save Button

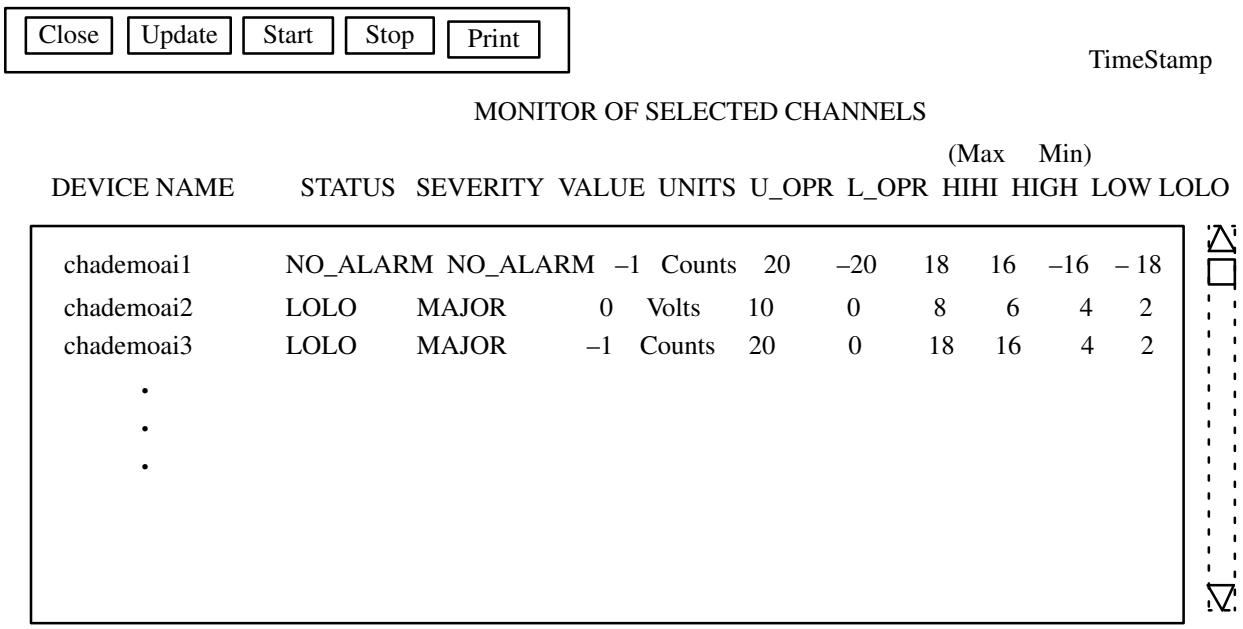
The *Save* button pops up a file selection box which in turns allows the user to save the highlighted items as a new IOC list file. This allows the new saved list (a subset of IOC) later to be loaded into the memory instead of the whole IOC list.

5.1.4 Ok Button

All the selected items are highlighted. The *Ok* button accepts the highlighted items as selected, closes the selection dialog, and pops up the *Selected Channels Information* window which displays the information of all selected channels and provides the capability of monitoring the selected channels.

5.2 Selected Channels Info Window

The selected channels Info window contains a menu bar, a labeled scrollable channel information window, and a time stamp. The menu bar consists of five buttons: *Close, Update, Start, Stop,* and *Print.*



5.2.1 Scrolled Window Contents

The scrolled window contains a list of information for the monitored channels. Each line displays the channel information obtained from IOC: channel name, status, severity, value, units, upper operating value (uoper), lower operating value (loper) , hihi, high, low, and lolo for each channel. The

string representation of the value field is displayed if the record type is either DBR_ENUM or DBR_STRING. If the upper and lower operating ranges are the same, the fields uoper and loper will be left blank for clarity. If hihi is same as lolo for a given channel, the fields hihi, high, low, and lolo will be left blank in the scrolled window. A typical example is shown in Table 1 on page 35.

The scrolled window content is slightly different when the monitoring starts. The value field is always displayed as a real number. The upper and lower operating ranges are always displayed. The fields hihi, high, low, and lolo will be replaced by the maximum operating value and minimum operation value. Both the maximum and minimum operating values are enclosed in parentheses. A typical example is show in Table 2 on page 36.

5.2.2 Close

The *Close* button will close the window and stop the monitoring of the selected channels.

5.2.3 Update

The *Update* button updates the contents of the scrollable text window. The *Update* button allows a user to get a new, time–stamped snapshot of the operation values of the selected channels from the IOC. The *Update* button updates the following fields of all selected channels: value, status, severity, alarm limits, upper and lower operating range, and the units.

5.2.4 Start

The *Start* button begins monitoring of the value change or alarm status change for all the selected channels. The time stamp, which marks the beginning of monitoring, is updated by the *Start* button. The channel information of the scrollable text window is updated according to the value change event. In addition to value, status, severity, operating ranges, and units, the maximum value and minimum value are also updated for a channel whenever there is a value change occurring.

5.2.5 Stop

The *Stop* button stops the monitoring process. It removes all the monitoring events for all the selected channels. The updating of scrolled window contents is stopped immediately.

5.2.6 Print

A file " /tmp/xmca.InfoOut " is automatically created by the *Print* button on the host machine where XMCA is running. This file contains the snapshot of the text contents of the scrolled window. Then XMCA automatically generates a printer copy of this file to the default printer.

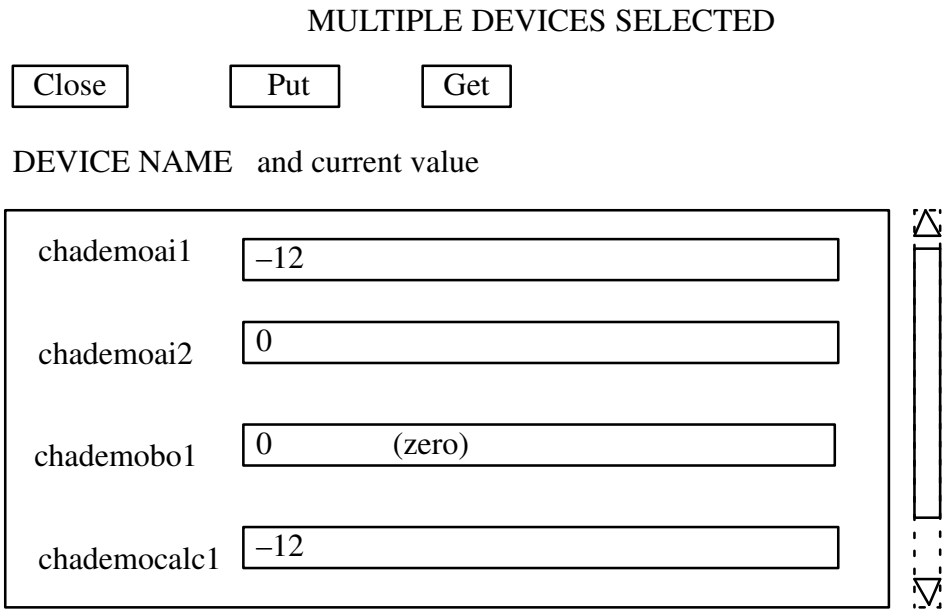
5.2.7 Cntl

The *Cntl* button pops up the multiple probe window. It allows the user to put new values to IOC for the monitored channels.

5.3 Multiple Probe Window

The multiple probe window contains a scrollable row–column of selected channels and three buttons: *Close, Put,* and *Get*. The scrollable row–column lists each selected channel name followed

by the text field containing the corresponding value obtained from IOC. A blank value field indicates that the device is either not connected or not found. A user can enter a new value by typing directly into the text field.



5.3.1 Put Button

The *Put* button accepts the values from the scrollable text fields window as input to IOC, and calls the channel access function to write the new values to the IOC. A user has to be aware of that he is actually writing into IOC. A sensible value should be entered for real case.

5.3.2 Get Button

The *Get* button calls the channel access function to get new values of all the monitored channels, and displays the new values. For the enumerate type of record the value is followed by the string representation which is enclosed within a pair of open parentheses.

5.3.3 Close

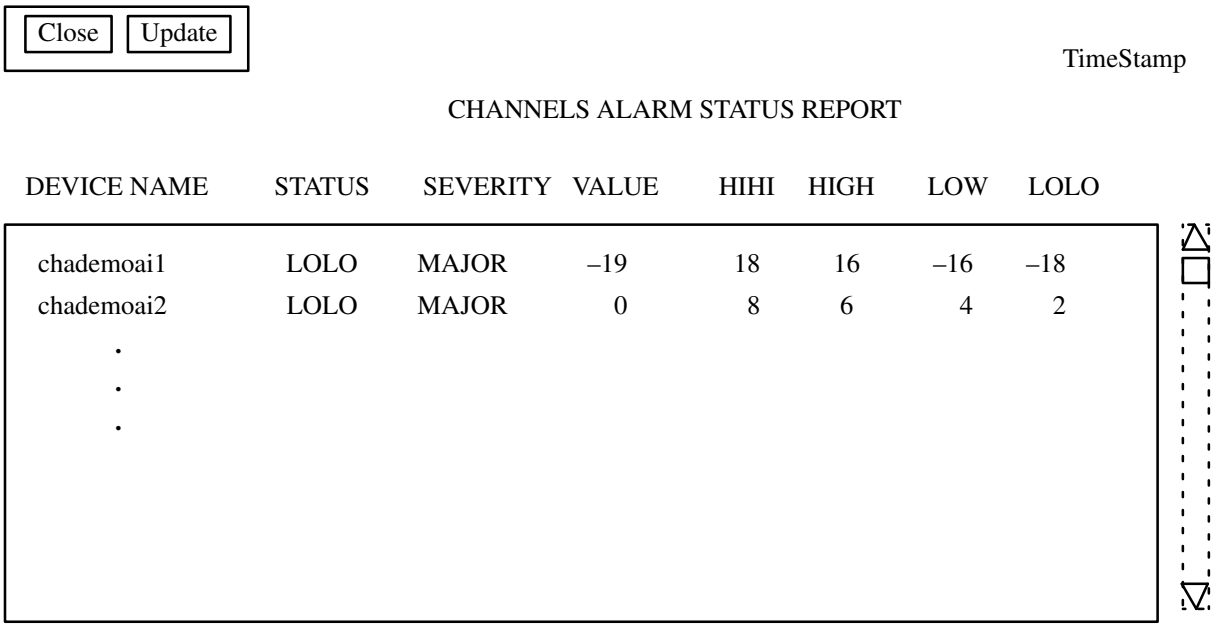
The *Close* button closes this multiple probe window.

CHAPTER 6 WHOLE LIST WINDOWS

There are two types of whole list information windows: the whole list *Alarm Info* window and the whole list *Info All* window.

6.1 Whole List Alarm Info Window

The *Alarm_All* button pops up the whole list *Alarm Info* window. This window contains a menu bar and a scrollable text window with time stamp. The menu bar contains two buttons: *Close,* and *Update*. The layout of this window is shown below.



6.1.1 Scrolled Alarm Window Contents

The scrolled window contains a list of alarm information for the whole list. Each line displays the channel information obtained from IOC. It contains channel name, status, severity, value, hihi, high, low, and lolo for each channel. If the hihi is same as the lolo for a given channel, the fields hihi, high, low, and lolo will be left blank in the scrolled window for clarity.

6.1.2 Close

The *Close* button closes the *Alarm Info* window.

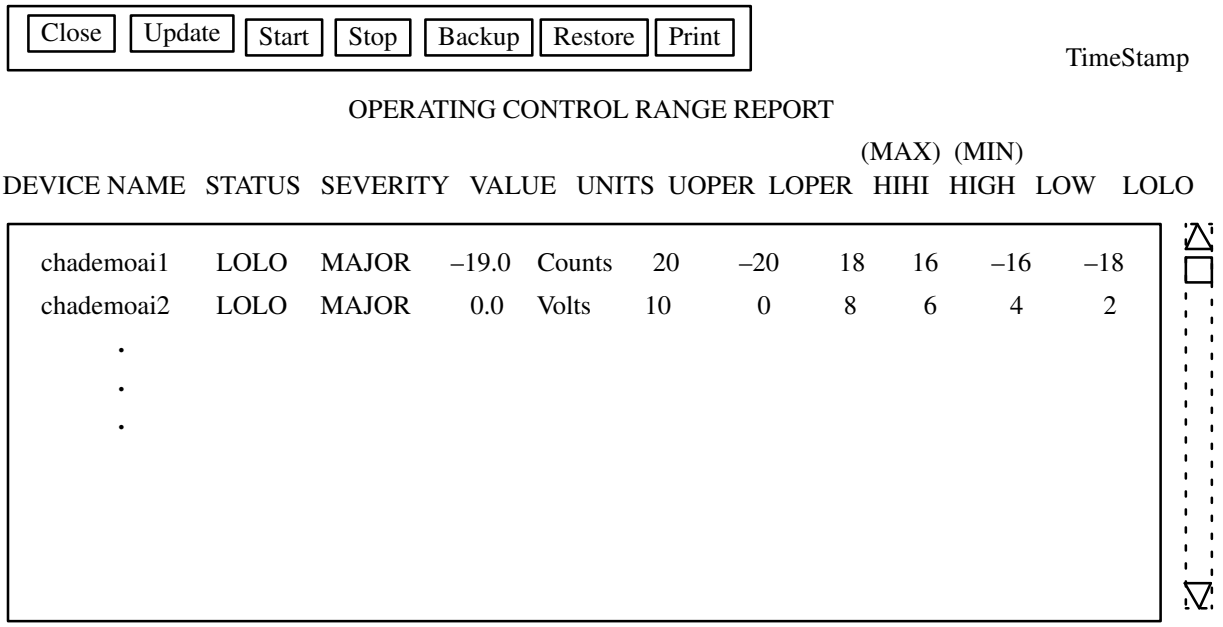
6.1.3 Update

The *Update* button gets a new snap–shot of the alarm characteristics defined for the whole list. It updates the scrollable text window and the time stamp. Sometimes the text window indicates that

the contents of the whole list are "*** not found / not connected ***" . This means that the timeout occurred before channel access was able to return the whole list. A user can simply press the *Update* button to refresh the contents of the whole window.

6.2 Whole List Info All Window

The *Info_All* button pops up the whole list *Info All* window. This window contains a menu bar and a scrollable text window. The menu bar contains 7 buttons: *Close, Update, Start, Stop, Backup, Restore,* and *Print*. The layout of this window is given below. If the size of the whole list exceeds 50, the *Start* and *Stop* buttons will not appear on the menu bar.



6.2.1 Scrolled Info Window Contents

The scrolled window contains a list of information for the whole list. Each line displays the channel information obtained from IOC: channel name, status, severity, value, units, upper operating value (uoper), lower operating value (loper) , hihi, high, low, and lolo for each channel. The string representation of the value field is displayed if the record type is either DBR_ENUM or DBR_STRING. If the upper and lower operating ranges are the same, the fields uoper and loper will be left blank for clarity. If hihi is same as lolo for a given channel, the fields hihi, high, low, and lolo will be left blank in the scrolled window. A typical example is shown in Table 1 on page 35.

The scrolled window content is slightly different when the monitoring starts. The value field is always displayed as a real number. The upper and lower operating ranges are always displayed. The fields hihi, high, low, and lolo will be replaced by the maximum operating value and minimum operation value. Both the maximum and minimum operating values are enclosed in parentheses. A typical example is show in Table 2 on page 36.

6.2.2 Close

The *Close* button closes the *Info All* display window.

6.2.3 Update

The *Update* button get a new snap–shot of the channel operating information for the whole list. It updates the scrollable text window and the time stamp. Sometimes the text window initially indicates that the contents of the whole list are "*** not found / not connected ***" . This means that the timeout occurred before channel access was able to return the whole list. A user can simply press the *Update* button to refresh the contents of the whole window. The fields updated by the *Update* button include status, severity, value, units, upper and lower operating ranges, and alarm settings hihi, high, low, and lolo.

6.2.4 Start

The presence of the *Start* button on the menu bar depends upon the size of the whole list. When the size is less than 50, this window allows a user to monitor the whole list through use of the *Start* button. Pressing the *Start* button begins monitoring of the value change events for the whole list. The time stamp, which marks the start of monitoring, is updated by this button as is the scrollable text window. The fields updated by the *Start* button include status, severity, value, units, upper and lower operating ranges, maximum value, and minimum value. The maximum and minimum operating values are displayed instead of the alarm conditioning settings hihi, high, low, lolo.

6.2.5 Stop

The presence of the *Stop* button on the menu bar depends upon the presence of the *Start* button; the *Start* and *Stop* buttons appear on the menu bar as a set. Pressing the *Stop* button immediately halts the value change monitoring for the whole list.

6.2.6 Backup

The *Backup* button allows a user to save the current values for the whole work list. It is assumed that each channel has a single value; therefore, backup of the waveform record is not supported by XMCA. The *Backup* button pops up a file selection box which allows a user to enter or select a file name for saving the current values. The assumed default file type of the backup file is ".bk".

6.2.7 Restore

The *Restore* button allows a user to restore saved values to IOC from a backup file. It is assumed that each channel has a single value; therefore, restoration of the waveform record is not supported by XMCA. The *Restore* button pops up a file selection box which allows a user to enter or select a backup file to restore. The assumed default file type of the backup file is ".bk".

6.2.8 Print

A file " /tmp/xmca.InfoOut " is automatically created by the *Print* button on the host machine where XMCA is running. This file contains the snapshot of the text contents of the scrolled window. Then

XMCA automatically generates a printer copy of this file to the default printer.

CHAPTER 7 INVOKING XMCA

7.1 Setup Requirement

To access the EPICS official released unix tools the following set up is required from an application directory:

$epics/Unix/sun4/bin/getrel $epics

where environment variable **$epics** should be set to the EPICS official release root directory**.**

For easy file management it is better to create an *ioc* subdirectory where various IOC list files or database links will reside. At the application directory make appropriate soft links to the ioc directory.

7.2 Running XMCA

From the application directory enter the XMCA command:

xmca

7.3 Remote Usage

To run XMCA on a remote machine, a user has to enter the following commands before invoking XMCA:

setenv DISPLAY host:0 (on remote machine)

xhost + (on host machine)

CHAPTER 8 XMCA FILES

8.1 Default File Type Settings

The database files created by DCT tool can be loaded into XMCA directly through the *LoadDB* option. The EPICS database file has ".database" as a suffix.

An IOC list file can contain combination of mixed channel names from different databases.

It is assumed that all the list files for a given system / subsystem reside in the same directory. All IOC channel list files should have ".ioc" as a suffix, and all backup files should have ".bk" as a suffix. When creating a new channel list file or sublist file a user should use ".ioc" as the suffix. When creating a new backup file a user should use ".bk" as the suffix.

The standard Motif File Selection Dialog Box is used for selecting or entering a file name. The length of a file name, including the path, should not exceed 60 characters.

XMCA can be invoked from any application directory. For easy file management, all the IOC channel list files, sublist files, and backup / restore files should reside in the same working directory (e.g. the ioc directory).
The user must have the privilege of reading from and writing to the directory. The master IOC list or backup file should have read permission only, to prevent a XMCA user from accidentally writing to the master file.

8.2 XMCA List File Description

The list file can be prepared via any text editor. It contains any combination of IOC channels which a user wants to get information or place monitoring. The list file just contains lines of channel names (i.e. process variable names) which are defined in any database and loaded into any IOC on the same network. Each line contains two fields: the process vairable type and process variable name from a database.

If the sublist option is desired from XMCA, the naming convention for process variables should use ':' as a subname qualifier in the DCT database creation (e.g. CHA:WP1:pumpOnCmdBO, CHA:WP2:pumpOnCmdBO), and the list should be sorted in the file before loading into XMCA. The number of lines in a list file should not exceed 2000 and the group names can not exceed 500.

8.2.1 Creating Complete IOC List File

The example of getting a complete list of process variables from a database is given below:

In the user application directory assumed there exists a DCT database "chademo.database". To create an "chademo.ioc" file in the *ioc* subdirectory by the Unix tool **dbl** command.:

mkdir ioc *only if directory ioc does not exist yet*

dbl chademo.database 1 > ioc/chademo.ioc
creates IOC list file chademo.ioc

8.2.2 Example List File

A typical list file (chademo.ioc) is given below. In this case, since no '**:**' is used in process variable names, no sublist will be generated by XMCA.

ai	chademoai1
ai	chademoai2
ai	chademoai3
bi	chademobi1
bi	chademobi2
bo	chademobo
calc	chademocalc1
calc	chademocalc2
calc	chademocalc3
calc	chademocalc4
calc	chademocalc5
calc	chademocalc6
calc	chademomask1
calc	chademomask2
compress	chademocompress1
compress	chademocompress2
compress	chademocompress3
compress	chademocompress4
mbbi	chademosevr1
stringin	chademostringin
waveform	chademowf1
waveform	chademowf2
waveform	chademowf3
waveform	chademowf4
waveform	chademowf5
waveform	chademowf6
waveform	chademowf7
waveform	chademowf8

8.3 XMCA Backup File Description

The backup file is automatically generated by the *Backup* button of XMCA, which records the current channel value for the whole work list. Currently only one value is recorded for each channel; therefore, the backup and restore of the waveform record is not supported by XMCA. The backup file consists of a list of channel names and corresponding values.

The backup file generated by XMCA contains the same number of lines as in the work list. Each line contains the process variable name and its value obtained from the IOC.

8.3.1 Example Backup File

A list of sample backup file is given below.

```
chademoai1                      12.0000000000
chademoai2                      0.0000000000
chademoai3                      12.0000000000
chademobi1                      0.0000000000
chademobi2                      0.0000000000
chademobo                       0.0000000000
chademocalc1                    12.0000000000
chademocalc2                    2.0000000000
chademocalc3                    16.0000000000
chademocalc4                    243.7873840332
chademocalc5                    -0.9509963393
chademocalc6                    0.3088211417
chademomask1                    4.0000000000
chademomask2                    0.0000000000
chademocompress1                13.0000000000
chademocompress2                0.5878771544
chademocompress3                0.8089018464
chademocompress4                3.0000000000
chademosevr1                    0.0000000000
chademostringin                 This is a test
chademowf1                      0.0000000000
chademowf2                      0.0000000000
chademowf3                      0.0000000000
chademowf4                      0.0000000000
chademowf5                      0.0000000000
chademowf6                      0.0000000000
chademowf7                      0.0000000000
chademowf8                      0.0000000000
```

8.4 XMCA Output Report

XMCA can generate a printed report of the operating control range report from the contents of the whole list information window. The printer output just gives a snapshot of the scrolled window. Table 1 (page 35) shows the snapshot of the system operating condition of the whole list. Table 2 (page 36) shows the report generated after a monitoring period.

CHAPTER 9 SYSTEM REQUIREMENTS

XMCA runs as a separate application on an Operator Interface (OPI). It uses channel access to communicate with the Input/Output Controllers (IOCs), and multiple XMCA processes can run on the same or different OPIs. Each XMCA process can operate on any combination of IOC channels. Thus XMCA can be used for displaying operating conditions for test stands, or any subsystems.

9.1 Pre And Post Conditions

The control system network needs to be operating properly in order for the XMCA to display information from all parts of the system. For each IOC channel not connected, the message "*** not found / not connected ***" will be displayed for the channel.

9.2 Channel Access Monitor Events

The IOC is responsible for detecting the value change event. The XMCA sends a ca_add_event request to the IOC for each database channel to be monitored. The IOC sends an event to XMCA whenever the associated channel has a value change.

9.3 Channel Access Get / Put Values

The XMCA's single and multiple selection windows can be used to get or set values for related fields in the database. XMCA sends a corresponding ca_get or ca_put to the IOC for each specified database channel. The IOC is responsible for getting and putting the value to specified channels.

9.4 Special Hardware Required

XMCA runs on a distributed network which is equipped with X–window system workstations and/or X terminals. They require a monitor, keyboard, and mouse.

9.5 Special Software Required

XMCA is an X–window application which uses the Motif Toolkit. The software requirement includes:

X Toolkit Library:

Xlib	X11
Xt Intrinsics	Xt
Motif widget set	Xm

CHAPTER 10 PROBE USER INTERFACE

10.1 What is Probe

Probe is a channel access utility with Motif user interface running under X–window. The idea of the probe is to provide a tool for the application engineer to debug his or her database on any IOC (Input Output Controller) from any location across the network.

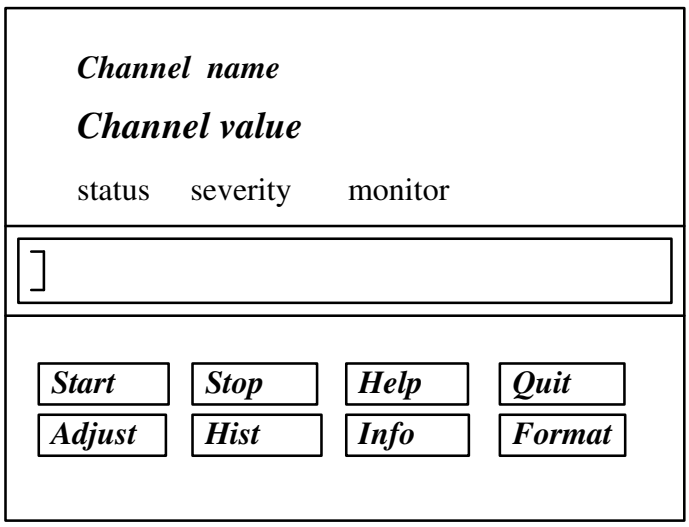
10.2 Getting Started

To start probe, type

probe

Then a small window with the title ‘‘probe” will pop up.

10.3 Layout of the Probe



Probe consists of one display area, one text entry box and eight function buttons. The display area occupies the top portion of the window and has three lines of text on it. The first line of the text shows the current connected process variable name; the second line shows the value of the process variable; the last line shows the alarm and severity of the process variable and whether or not the process variable is monitored. The text entry box in the center of the window is used for entering the name of the process variable name. The eight function buttons located on the bottom portion of the window

are used to select the different modes and functions of **probe**.

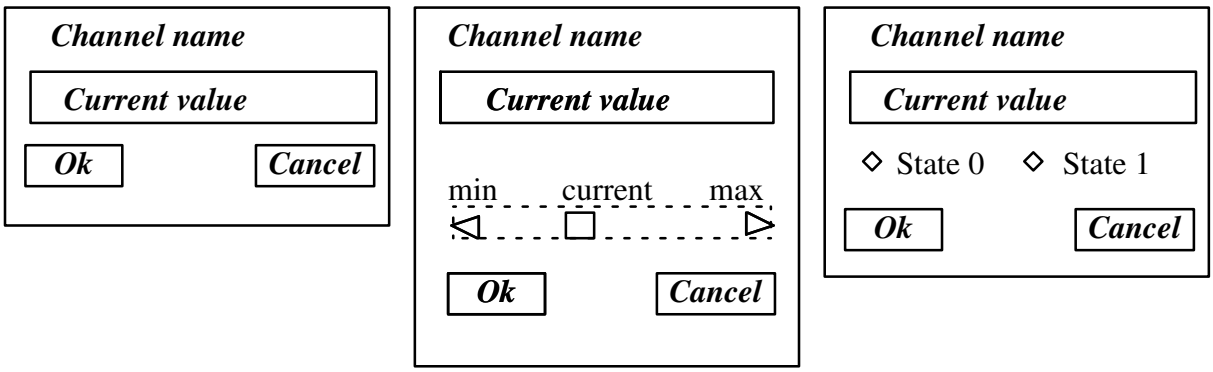
10.4 Connecting a Process Variable

To enter a process variable name, place the mouse pointer over the text entry box and click the left mouse button once. You should see the box highlighted and a flashing cursor. Type in the process variable name and press the <return> key. If everything goes right, you should see the process variable name you entered appear on the first line of the display and a value appear on the second line. What you just did is asking **probe** to set up a connection to your database and take a snapshot on the value and status of the process variable you specified. If an "*Unknown channel name !*" message appears instead, check the spelling of the process variable and try again.

10.5 Monitoring a Process Variable

If you want to read the value and status of the process variable continuously, you have to tell **probe** to monitor the process variable. To monitor the process variable place the mouse pointer over the *Start* button and click the left mouse button once. The word "*monitor*" will appear on the status line. **Probe** will update the change in the value and status of the process variable. If you want to stop monitoring, click the *Stop* button once. The word "*monitor*" will disappear from the status line.

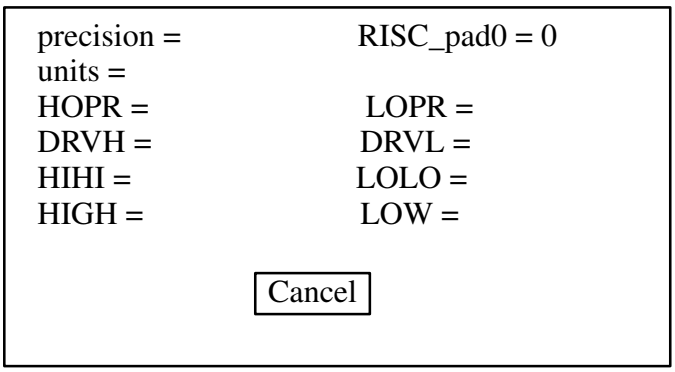
10.6 Changing the Value of a Process Variable



You can use **probe** to change the value of a process variable. By clicking the *Adjust* button once, an adjust window containing one or two of the control elements, such as text entry box, slider, button panel, will pop up. Which control elements are included in the window depends on the data type of the process variable. **Probe** classifies the process variable into four data types: text, integer, floating point number, and state. For text data type, the adjust window contains a text entry box only; for integer and floating point number types, it contains a text entry box and a slider; and for state data type, it contains a text entry box and button panel. For example, if you want to change the value of an integer type process variable, you can either use the slider or enter the value directly into the text entry box on the adjust window. For a state type process variable, you have the choice of entering the state name, and number or clicking one of the buttons on the button panel.

Use the *Ok* button to accept the new setting, and use the *Cancel* button to close the window.

10.7 Getting the Record Setting

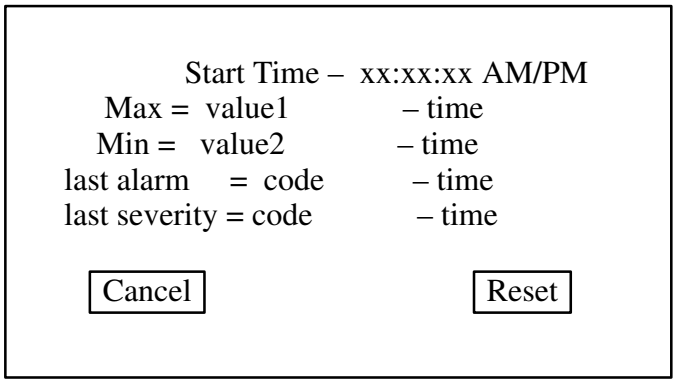


Probe provides two types of information: process variable setting and run–time history. By clicking the *Info* button, a window containing the process variable setting will pop up. The number of fields shown also varies from data type to data type. Listed below are the possible settings which may be displayed and their explanation:

precision	Number of decimal point of a floating point number will maintain in the display.
RISC_pad0	Ignore this one at the moment.
units	Units of the process variable.
HOPR	*H*igh *OP*erator *R*ange, information used by the slider, bar graph, and charts to specify the upper limit in the display manager.
LOPR	*L*ow *OP*erator *R*ange, information used by the slider, bar graph, and charts to specify the lower limit in the display manager.
DRVH	*DR*i*V*e limit *H*igh, the limit of the highest output value.
DRVL	*DR*i*V*e limit *L*ow, the limit of the lowest output value.
HIHI	*HI*gh *HI*gh alarm triggering level.
HIGH	*HIGH* alarm triggering level.
LOW	*LOW* alarm triggering level.
LOLO	*LO*w *LO*w alarm triggering level.

To close the window, click the *Cancel* button once.

10.8 Getting the Run Time History



By clicking the *Hist* button, a window with the label "History_popup" will appear if the process variable is monitored. The first line of the text will give the time when the monitor on the process variable was started. The last two line of the text will give what the alarm and severity are, and when the change occurred. The information in the middle of the text is also data type–dependent. For the text data type, the information includes the current string, the last string, and their time stamp; for integer and floating point numbers, the maximum and minimum value and their time stamps are included; for the state type record, the last state, current state, and their time stamps are shown. There are also two buttons, *Cancel* and *Reset* at the bottom portion of the window. The *Cancel* button is used to close the history window. The *Reset* button is used to restart the monitor. If the *Reset* button is hit, all time stamps will equal the current time. Also, the current string and last string for the text data type, current state or last state for state data type, or maximum and minimum for the integer and floating point data type will be set to the current value of the process variable.

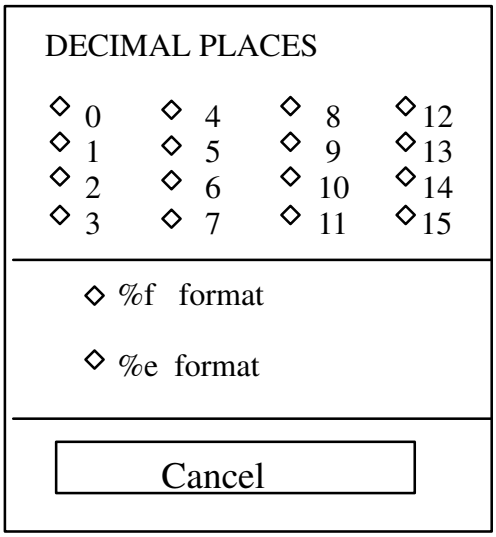
10.9 Changing the Display Format

The display format is only valid for the floating point data type. By clicking the *Format* button, a format window will come up. The user can select the number of decimal places to show and the data format to use. The decimal places are in the range of 0 to 15 and the data format can be either engineering notation, "%f format" or scientific notation, "%e format". For example, the value is 19.9999 and you want two decimal places. In engineering notation , the result is "19.99" and in scientific notation "1.99e+1" is displayed. Use the *Cancel* button to close this window.

Note : Every time a new process variable is connected, the default decimal places will be the precision value in the process variable setting.

10.10 Exiting the Program

Just click the *Quit* button to quit the program at any stage of the program.


DECIMAL PLACES
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
%f format
%e format
Cancel

APPENDIX

Table 1

Fri Sep 4 15:36:59 1992

	DEVICE_NAME	STATUS	SEVERITY	VALUE	UNITS	H_OPR	L_OPR	(MAX) HIHI	(MIN) HIGH	LOW	LOLO
1	chademoai1	NO_ALARM	NO_ALARM	-7.00000	Counts	20.000	-20.000	18.000	16.000	-16.000	-18.000
2	chademoai2	LOLO	MAJOR	0.00000	Volts	10.000	0.000	8.000	6.000	4.000	2.000
3	chademoai3	LOLO	MAJOR	-7.00000	Counts	20.000	0.000	18.000	16.000	4.000	2.000
4	chademobi1	NO_ALARM	NO_ALARM	State 0							
5	chademobi2	NO_ALARM	NO_ALARM								
6	chademobo	NO_ALARM	NO_ALARM	zero							
7	chademocalc1	NO_ALARM	NO_ALARM	-6.00000	Counts	20.000	-20.000	16.000	12.000	-12.000	-16.000
8	chademocalc2	HIGH	MINOR	7.00000	Counts	10.000	0.000	8.000	6.000	4.000	2.000
9	chademocalc3	NO_ALARM	NO_ALARM	9.00000	Counts	20.000	0.000	16.000	12.000	8.000	4.000
10	chademocalc4	NO_ALARM	NO_ALARM	244.41570							
11	chademocalc5	NO_ALARM	NO_ALARM	0.95112	Counts	20.000	-20.000	16.000	12.000	-12.000	-16.000
12	chademocalc6	NO_ALARM	NO_ALARM	0.95099	Counts	20.000	-20.000	16.000	12.000	-12.000	-16.000
13	chademocompress1	NO_ALARM	NO_ALARM	-5.00000	Integer	20.000	-20.000				
14	chademocompress2	NO_ALARM	NO_ALARM	0.30912		2.000	-2.000				
15	chademocompress3	NO_ALARM	NO_ALARM	0.95100		2.000	-2.000				
16	chademocompress4	NO_ALARM	NO_ALARM	8.00000	Integer	10.000	0.000				
17	chademomask1	NO_ALARM	NO_ALARM	0.00000							
18	chademomask2	NO_ALARM	NO_ALARM	0.00000							
19	chademosevr1	NO_ALARM	NO_ALARM	NO_ALARM							
20	chademostringin	NO_ALARM	NO_ALARM	This is a test							
21	chademowf1	NO_ALARM	NO_ALARM								
22	chademowf2	NO_ALARM	NO_ALARM	0.00000		10.000	0.000				
23	chademowf3	NO_ALARM	NO_ALARM	0.00000		1.000	-1.000				
24	chademowf4	NO_ALARM	NO_ALARM	0.00000		20.000	0.000				
25	chademowf5	NO_ALARM	NO_ALARM	0.00000		20.000	-20.000				
26	chademowf6	NO_ALARM	NO_ALARM	0.00000		20.000	0.000				
27	chademowf7	NO_ALARM	NO_ALARM	0.00000		2.000	-2.000				
28	chademowf8	NO_ALARM	NO_ALARM	0.00000		2.000	-2.000				

Table 2

Fri Sep 4 15:38:07 1992

	DEVICE_NAME	STATUS	SEVERITY	VALUE	UNITS	H_OPR	L_OPR	(MAX) HIHI	(MIN) HIGH	LOW	LOLO
1	chademoai1	NO_ALARM	NO_ALARM	-1.00000	Counts	20.000	-20.000	(19.000)	(-19.000)		
2	chademoai2	LOLO	MAJOR	0.00000	Volts	10.000	0.000	(0.000)	(0.000)		
3	chademoai3	LOLO	MAJOR	-1.00000	Counts	20.000	0.000	(19.000)	(-19.000)		
4	chademobi1	NO_ALARM	NO_ALARM	0.00000		0.000	0.000	(0.000)	(0.000)		
5	chademobi2	NO_ALARM	NO_ALARM	0.00000		0.000	0.000	(0.000)	(0.000)		
6	chademobo	NO_ALARM	NO_ALARM	0.00000		0.000	0.000	(0.000)	(0.000)		
7	chademocalc1	NO_ALARM	NO_ALARM	0.00000	Counts	20.000	-20.000	(19.000)	(-20.000)		
8	chademocalc2	LOLO	MAJOR	0.00000	Counts	10.000	0.000	(9.000)	(0.000)		
9	chademocalc3	LOLO	MAJOR	-4.00000	Counts	20.000	0.000	(19.000)	(-20.000)		
10	chademocalc4	NO_ALARM	NO_ALARM	239.07500		0.000	0.000	(314.787)	(0.000)		
11	chademocalc5	NO_ALARM	NO_ALARM	-0.58763	Counts	20.000	-20.000	(1.000)	(-1.000)		
12	chademocalc6	NO_ALARM	NO_ALARM	0.95112	Counts	20.000	-20.000	(1.000)	(-1.000)		
13	chademocompress1	NO_ALARM	NO_ALARM	1.00000	Integer	20.000	-20.000	(19.000)	(-20.000)		
14	chademocompress2	NO_ALARM	NO_ALARM	0.95109		2.000	-2.000	(1.000)	(-1.000)		
15	chademocompress3	NO_ALARM	NO_ALARM	0.80913		2.000	-2.000	(1.000)	(-1.000)		
16	chademocompress4	NO_ALARM	NO_ALARM	1.00000	Integer	10.000	0.000	(9.000)	(0.000)		
17	chademomask1	NO_ALARM	NO_ALARM	0.00000		0.000	0.000	(0.000)	(0.000)		
18	chademomask2	NO_ALARM	NO_ALARM	0.00000		0.000	0.000	(0.000)	(0.000)		
19	chademosevr1	NO_ALARM	NO_ALARM	0.00000		0.000	0.000	(0.000)	(0.000)		
20	chademostringin	NO_ALARM	NO_ALARM	This is a test							
21	chademowf1	NO_ALARM	NO_ALARM								
22	chademowf2	NO_ALARM	NO_ALARM	0.00000		10.000	0.000	(0.000)	(0.000)		
23	chademowf3	NO_ALARM	NO_ALARM	0.00000		1.000	-1.000	(0.000)	(0.000)		
24	chademowf4	NO_ALARM	NO_ALARM	0.00000		20.000	0.000	(0.000)	(0.000)		
25	chademowf5	NO_ALARM	NO_ALARM	0.00000		20.000	-20.000	(0.000)	(0.000)		
26	chademowf6	NO_ALARM	NO_ALARM	0.00000		20.000	0.000	(0.000)	(0.000)		
27	chademowf7	NO_ALARM	NO_ALARM	0.00000		2.000	-2.000	(0.000)	(0.000)		
28	chademowf8	NO_ALARM	NO_ALARM	0.00000		2.000	-2.000	(0.000)	(0.000)		