

COMPASS Framework documentation

COMPASS Framework DCS team¹ CERN-IT-CO, COMPASS

v1.4, 24-November-2000

1 Purpose

This document describes the framework which was designed for the COMPASS experiment. This is for the users (physicists, engineers, students, etc.) who want to use the framework for building their DCS (Detector Control System)². It contains a guide on how to use the framework, a description of the facility for creating, modifying or deleting the instances of the framework element types. PVSS knowledge and vocabulary are required to understand this document. This document does not give any detailed information concerning the implementation of the PVSS scripts, it just gives the different functionalities of the views, the libraries and the elements types.

PVSS terminology is used in this document, e.g.: dp stands for data point, dpe or dpElement for data point element, etc. The term alert used in this document should be understood as alarm. A panel is a term used in PVSS to design a window where data are presented.

The Section 2, "Description" gives a description of the COMPASS DCS, the Section 3, "Framework elements" describes all the elements types of the COMPASS framework, the Section 4, "Framework Panels" describes all the panels, the Section 5, "Framework Libs" describes all the libs used by the panels and the Section 6, "Framework Scripts" describes all the scripts used in the production system.

The CAEN HV configuration tool (Section 4.1.41, "HVConfig") was developed by the JCOP framework team. Christian Schuegger, TU Munchen, developed the error handling facility for the function call of the library, the fwk_complibElementCommon.ctl library (Section 5.3, "fwk_complibElementCommon.ctl").

2 Description

The COMPASS DCS is based on a two layer architecture (Figure 1):

- on the top: the PVSS SCADA³ system: this is used for the supervision process and high level automated action
- on the bottom: DCS front-end equipment⁴ with control software, such as VME, PC, PLC, fieldbus equipment, etc.: this is used for interfacing to the hardware, for low level automated action.

¹ The COMPASS framework DCS team: Herve Milcent, IT-CO; Oliver Holme, IT-CO; Juha Rossi, EP-HL; Niko Markus Karlsson (left the team in August 2000).

² Also named Slow Control.

³ SCADA: Supervision Control And Data Acquisition.

⁴ A high voltage crate is also a front-end equipment but it not considered as a DCS front-end equipment, because it has to be interfaced to the DCS by a software.

Two PCs: pcompass04 (linux-SuSE 6.0) and pcompass03 (NT) are dedicated to the supervision. The Linux computer holds all the PVSS functionalities: database, archiving, etc. The NT computer is used for the connection to hardware via OPC. During normal operation, when the users want to operate their equipment, the PVSS system called PRODUCTION is used (located in /home/compass_prod). Up to 5 users can operate at the same time their equipments using 5 different graphical interfaces. A graphical interface can be started by a remote X window connection to the linux PC or by installing the PVSS software locally in the PC and configuring it to connect to the PVSS system running in the linux PC.

To avoid conflicts with the running system (deleting element, stop or start of the PVSS system, development of new panels, etc.) the development is done in another PVSS system called OFFLINE⁵ (located in /home/compass_offline). Up to 2 users can developed new PVSS panels or script at the same time in one OFFLINE system. Other OFFLINE systems can be created if needed. The framework development is done in the OFFLINE system and the new release are copied to the PRODUCTION system during shutdown time. The user development will have to be transferred to the PRODUCTION system to be used in operation time. There is no restriction on the number of OFFLINE system that can be created.

The differences between the PRODUCTION system and the OFFLINE systems are:

- it is not possible to delete elements from the PRODUCTION system
- alarm setup cannot be created or deleted in the PRODUCTION system

The front-end computers, PLCs or fieldbus components are shared between detectors to reduce the costs.

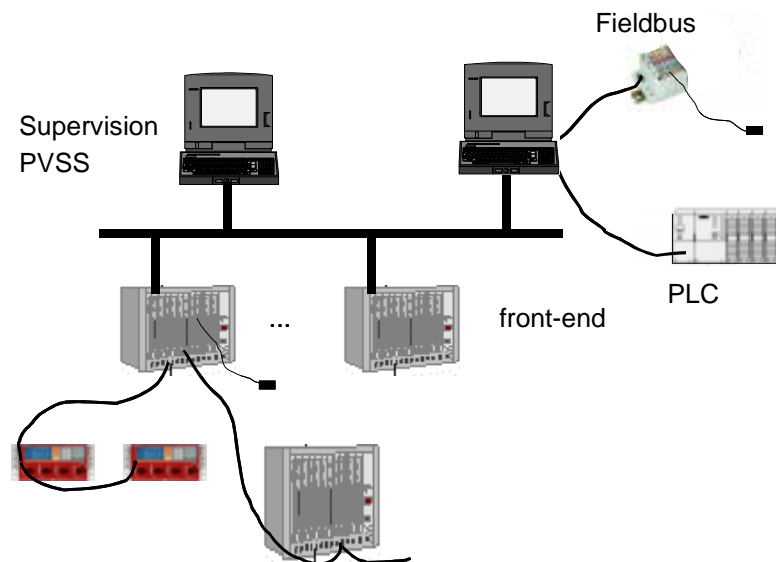


Figure 1: COMPASS DCS hardware architecture.

⁵ other offline PVSS system can be defined, but only one production system should exist.

To ease the development of the COMPASS DCS and to have a common representation of the common elements, a framework was designed. The PVSS implementation and the PVSS configuration is hidden to the developer. It is composed of a set of common elements types which are used in COMPASS (Section 3.1, "element list"). It provides a tree hierarchy, instances of the element types can be created, modified, deleted and placed in the tree by using panels based on PVSS. The CAEN HV behavior (Section •, "setTrip: set the trip time value to all the CAEN HV channels.") is implemented within the framework. The framework also provides a tool to browse the tree (Section 4.2, "Run-time panels"), to configure the tree hierarchy (Section 4.1, "Modification panels"). It is possible to put in the COMPASS tree a reference to an element which is not from the framework (Section 3.10, "interfacing new element type with the framework"). The actions described in Section 3.4, "action on the framework elements" are also supported by the framework. The framework can also be used with a front-end software and its configuration, the data can be imported within the framework (Section 4.1.39, "ImportHVConfig").

The elements of the DCS are organized in a tree hierarchy composed of nodes and leaves, a node is an element having children (nodes or leaves) and a leaf is a node without children. Usually leaves are connected to a hardware value, e.g.: a detector is a node and a value of a pressure sensor is a leaf. This tree hierarchy represents the way the elements are grouped, located inside the COMPASS detectors, to which detector they belong to, the way the user wants to operate them, etc. The first layer is the detector, like RICH, STRAW, etc. or a common service like RACK cooling, etc.

The detector level is divided into two sub-trees:

- the hardware tree: giving a hardware structure view of the elements inside a detector.
- the logical view: giving a logical view of the elements inside a detector.

The logical tree can be used to group the elements together and to operate them as a single entity. E.g.: the high voltage crate of the RICH detector will certainly power the Jura side and the Saleve side of the detector. It can be useful to separate the two high voltage systems in order to operate them independently. Therefore, two logical entities can be created, called HV_GROUP for example, and they will be located in the logical tree of the RICH detector. The operators on shift, the physicists can then operate the high voltage by using the HV_GROUP. The high voltage crate will be located in the hardware tree of the RICH and the HV_GROUP in the logical tree. The responsible of the DCS may be more interesting in the hardware view than in the logical view (Figure 2).

A service element can be used to represent a common service provided to the experiment, such as water cooling, rack control, gas system, etc. Data from different detectors, such as temperature, HV, etc. are read through the same software in order to reduce the cost, a service can be also used to represent this readout hierarchy (Section 3.12, "Element of a detector which are coming from service"). This can help to diagnose problems due to this common readout system, to take action on all the elements of this common readout. E.g.: in COMPASS, two PLCs are used for gas systems (MWPC, GEM, STRAW, RICH), these PLCs are connected to the DCS via one single entity: an OPC Server. So all the data are read by one single software, but each piece of data belongs to different detectors: that means that each detector wants the possibility to set the alarm limits, etc. to their elements and they don't want that another detector expert modify their configuration. But if the OPC Server or the PLC is not working correctly, it would be difficult to show that each element is not correct, and it could difficult for a fast diagnostic to locate where is the problem. Therefore

the PLC readout can be declared as a service, which reads elements from different gas systems belonging to different detectors. At operation time each detector operator is only aware of its data even if they are read by the service entity, but in case of doubt about the validity of the data, the dcs expert can easily look at the service state, and diagnose the reason of the incorrectness of the data, because the elements will belong both to the service and the detector. One can also imagine that the dcs expert could disable the whole service path because he has to repair the service readout, of course this implies an agreement with the detector expert.

Only one instance of each element exists in the tree but a reference to it can be in many node belonging to the same detector. It is also possible to get in a detector a reference of an element belonging to a service. Each leaf and node has an element representing its state. The state of the node is calculated (Section 3.3, "Alarms and states") from the state of all its child nodes or child leaves. At the node and leaf level, some actions can be initiated (Section 3.4, "action on the framework elements").

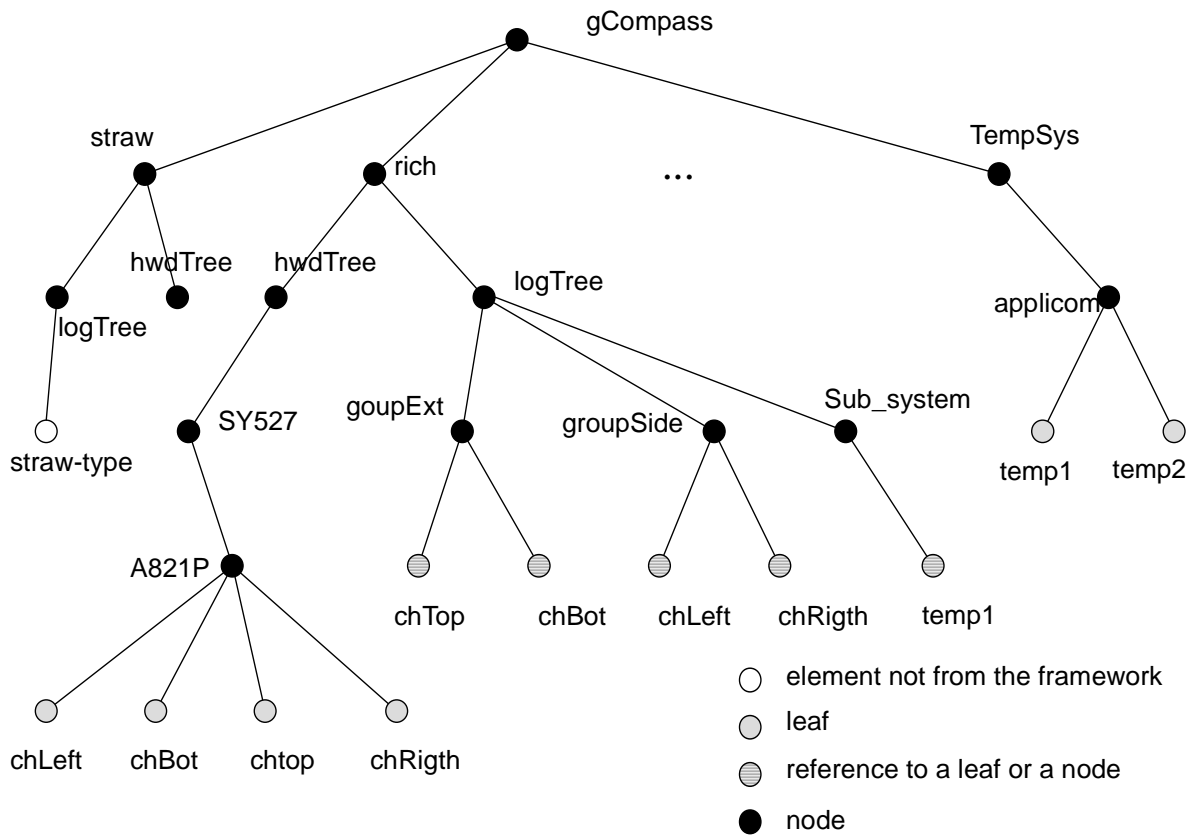


Figure 2: Tree hierarchy.

The COMPASS framework is composed of a set of PVSS panels, scripts, libraries, etc. The /home/compass_prod and /home/compass_offline have the following directory hierarchy:

- common: containing all the common elements of COMPASS
- Fwk-hv: containing all the high voltage elements used in the JCOP⁶ framework, these elements were re-used in the COMPASS framework.
- compass-hv: containing all the elements concerning COMPASS HV which are not in the JCOP framework or which were redefined, e.g.: it is possible to import the configuration of the IT-CO front-end software, this implementation is stored in this directory.
- dim-pvss: containing all the elements corresponding to DIM.
- new-release: containing all the new elements which have to be tested.

Two communication interfaces are supported by the framework: DIM and OPC. DIM Server and DIM Client can be connected to PVSS.

The tree hierarchy described above should not be mixed with the view hierarchy the users may design. One can imagine a developer doing a PVSS panel where data from different detectors are shown, e.g. in the RICH panel, one can put data from the RICH detector and from the temperature control system. Panels are linked together by mean of 'opening a panel from another one', however there is no link between them in term of alarm, etc. It means that from one panel, one would not be able to know in which sub-panels there is an alarm. On the other hand the tree hierarchy of the elements can be used to locate rapidly from which part of the COMPASS experiment are the problems because each node reflects the state of all its children nodes or leaves. It is also possible that the view hierarchy follows strictly the tree hierarchy of elements. Both approaches can also be mixed.

3 Framework elements

The COMPASS DCS is organized as a tree hierarchy, the top of the tree is called gCompass or dcs1:gCompass⁷. PVSS summary alarm are defined for the CAEN_HV_ch, CAEN_HV_ch_SP and FAN_TRAY elements type. A detector developer can use the data point type described in the Section 3.1, "element list" to build his detector hierarchy or can create his own data point type. In that latter case some rules (Section 3.10, "interfacing new element type with the framework") have to be followed to allow an integration in the COMPASS tree. The actions on the framework elements are described in Section 3.4, "action on the framework elements".

⁶ JCOP: Joint Control Project: <http://itcowww.cern.ch/jcop>.

⁷ dcs1 is the system name. In PVSS, the name of an element can be gCompass or dcs1:gCompass.

3.1 element list

Table 1 shows the list of common elements type used in COMPASS. Other datapoint are used internally by the COMPASS framework, e.g: OPC Server, DIM, etc.

Table 1: Element type

dp type	description
COMPASS	only one instance of this element type is used: gCompass, this is the startup point of the COMPASS DCS list of elements.
DETECTOR	This element type represents a detector like the RICH, STRAW, etc. or a service like water cooling, etc. The DETECTOR has two internal tree: the hardware tree used for the hardware location of the elements and the logical tree. The hardware tree represent the hardware view of the detector (list of HWD, high voltage crate, etc.) and the logical tree the view of the physicists operation. E.g.: the high voltage crate will be located in the hardware tree but the physicist could use HV_SYSTEM or HV_GROUP to handle the high voltage.
SERVICE	This element type represents a common readout path, or a service provided to the experiment: e.g.: a measurement system, gas system, etc. This element can be use to group in a hierarchical manner all the elements from different detectors which are read through a common path, or which are coming from a central service of the experiment.
SUB_SYSTEM	This element type represents a sub-system of a detector, it could be used to create a logical group of elements of type: AI, AO, AO_F, DI, DO, DO_F, SUB_SYSTEM, HV_GROUP or HV_SYSTEM. E.g. the Saleve side of the RICH detector, the logical group of all the temperature measured in the RICH detector, etc. A SUB_SYSTEM element is in the logical tree of a DETECTOR element.
HWD	this is a used to group element of type AI, AO, AO_F, DI, DO, DO_F and HWD, a kind of collection, it represent the hardware location of the measured elements. An element of this type can also be used to show the status of an OPC Server. E.g.: the RICH PLC for the gas system, the APPLICOM OPC Server providing all the information connected to the PROFIBUS fieldbus, the Jura node of the Trigger Hodoscope detector, etc.
AI	analogue input data, it can be connected to an OPC Item, to a DIM data or can be used just as an internal memory element. E.g.: a temperature sensor can be associated to a AI, the requested position of the Trigger Hodoscope detector, etc.
AO	analogue output data, it can be connected to an OPC Item, to a DIM data or can be used just as an internal memory element. E.g.: a valve opening set point (0 to 100%) can be associated to a AO, etc.
AO_F	analogue output and input data, it can be connected to an OPC Item, to a DIM data or can be used just as an internal memory element. E.g.: a valve opening set point (0 to 100%) and readout valve open position can be associated to a AO_F, etc. The AO_F can be used when the physical element it is connected to represents a single device. Another possible representation is to use a AI and a AO.
DI	digital input data, it can be connected to an OPC Item, to a DIM data or can be used just as an internal memory element. E.g.: a valve opened/closed position can be associated to a DI, the end switch of the Trigger Hodoscope detector motor, etc.

Table 1: Element type

dp type	description
DO	digital output data, it can be connected to an OPC Item, to a DIM data or can be used just as an internal memory element. E.g.: open or close a valve can be associated to a DO, the start motor of the Trigger Hodoscope detector, etc.
DO_F	digital output and input data, it can be connected to an OPC Item, to a DIM data or can be used just as an internal memory element. E.g.: open or close a valve and the opened/closed position of the valve can be associated to a DO_F, etc. The DO_F can be used when the physical element it is connected to represents a single device. Another possible representation is to use a DI and a DO.
SY127	CAEN SY127 high voltage crate, collection of CAENHVBoard
SY527	CAEN SY527 high voltage crate, collection of CAENHVBoard
SY403	CAEN SY403 high voltage crate, collection of CAENHVBoard
CAEN_HV_ch	CAEN high voltage channel read periodically
CAEN_HV_ch_SP ^a	CAEN high voltage channel read in the spill and out of the spill
CAENHVBoard	CAEN high voltage board, collection of CAEN_HV_ch or CAEN_HV_ch_SP
PROCS	element representing an external process to be monitored: e.g: to monitor if the PVSS00dim is running correctly, to monitor that the IT-CO front-end system is running correctly, etc. The value of this element should be changed in a time less than 40 sec, otherwise an alarm is generated.
HV_GROUP	collection of high voltage channel: CAEN_HV_ch, CAEN_HV_ch_SP
HV_SYSTEM	high voltage system: collection of HV_GROUP, HV_SYSTEM, CAEN_HV_ch or CAEN_HV_ch_SP
FAN_TRAY	a WIENER fan tray.
FAN_TRAY_SP ^b	a WIENER fan tray read in the spill and out of the spill.

a. not supported by the IT-CO front-end software in version v1.4.

b. not implemented in version v1.4

Each instance of an element type has a unique name in the PVSS system. The naming convention of the element instances, except for DETECTOR, PROCS and COMPASS element type, is the following: systName:DET_TYPE_eltName where:

- systName: is the DCS system name usually dcs1
- DET: is the detector name or the service name to which the element belongs to, this is selected by the user when it creates the instance of an element type.
- TYPE: is the type of the dp type from Table 1. selected by the user. For the CAEN_HV_ch and CAEN_HV_ch_SP element type, TYPE is equal to CCH, for SY127, SY527 and SY403 TYPE is equal to CCR, for CAENHVBoard TYPE is equal to CB and for FAN_TRAY TYPE is equal to FT, this is to avoid long name.
- eltName is the name given by the user, this could be the logical location of the element or the hardware location.

It is better to use as eltName the logical location instead of the hardware characteristic (Section 3.2, "Restrictions"). E.g.: for a CAEN_HV_ch element type which represents the

high voltage channel of the RICH detector, it is better to give as eltName: HVTopPlex rather than HVChannel2Crate3, later this will allow to change easily the characteristics of the channels such as the channel number or the crate number without renaming the instance of the element.

An instance of an element of type DETECTOR or PROCS has the following name: systName:eltName where:

- systName: is the DCS system name usually dcs1
- eltName is the name given by the user by the user, this is the detector name for DETECTOR element type or the name of the process to monitor for the PROCS element type.

An instance of an element of type SERVICE has the following name: systName:eltName where:

- systName: is the DCS system name usually dcs1
- eltName is the name given by the user by the user, this is the service name for SERVICE element type.

There is only one instance of the COMPASS element type named: systName:gCompass. where systName: is the DCS system name usually dcs1.

3.2 Restrictions

The name of the elements has to be unique⁸ within the DCS. Once the element is created it is not possible to rename it, one can create a new instance with the new name and delete the old one. It is not possible to move an element from a detector sub-tree to another detector sub-tree or to get a reference of an element belonging to another detector. Not having a reference on an element means not being able to control, operate or modify it. However it is possible to show in the same panels data from different detectors because the graphical view which can be designed as a tree hierarchy can be a different one than the tree hierarchy of the elements.

It is possible in a detector to have a reference of an element which is in a service hierarchy only if the element was declared at creation to be a member of the detector. Having this elements in the detector hierarchy implies that a detector expert operate/send actions to/modify the configuration of this element.

For all the elements of the framework, the alarms are defined only on the value connected to input data, there are no alarm on output data⁹.

3.3 Alarms and states

Alarms are used to show an anomaly on an element, to show that an element is in a bad state, to diagnose rapidly where are the problems. An alarm is generated if the value is out a range. Alarms are usually defined on a leaf, it can also be a summary alarm.

⁸ This is PVSS constraint.

⁹ in release v1.4

Each alarm will generate a message in the AlertScreen (Section 4.2.29, "Alert panel"). E.g.: if a high voltage channel trips, its read voltage will drop down and its read state will be off, therefore up to 4 alarms messages can appear in the AlertScreen: a message for the alarm on the read voltage, a message on the alarm of the trip status, a message for the alarm on the state and a message on the summary alarm of the channel. Acknowledging an alarm is an action initiated by the operator to show that he has taken in account the alarm, of course appropriate actions on the element have to be done in order that the alarm disappears.

Each leaf and each node have a state. A leaf's state is calculated from its alarm settings, a node's state is calculated from its child leaves and child nodes state. The state represents the status of the element: OK, Warning, Fault or Fatal. This state is propagated from the bottom to the top of the tree. It is possible to temporary stop this propagation by masking it at a node level or leaf level. The state (if it is not masked) can be used to locate where the problems are.

Masking is an action which blocks the propagation of the state to the upper layer. It set the state of the element to OK. Un-masking will un-block this propagation.

The state calculation is done in a PVSS script: Section 6.4, "dcsMainStatus.ctl".

3.4 action on the framework elements

Some actions are supported by the framework. These actions can be initiated at different levels of the compass tree. As an element may belong to two different nodes inside the same detector or inside a service, this could lead to some inconsistencies¹⁰, e.g.: disabling a HV channel which is reference in a HV crate and in a HV group may be misinterpreted by an operator, because this element will not be operable. There are two kinds of group of actions: the general purpose actions (Section 3.4.1, "General purpose actions") and the HV actions (Section 3.4.2, "HV actions").

3.4.1 General purpose actions

The general purpose actions supported by all the elements of the framework, they are triggered by using the panel described in Section 4.2.12, "actionPanel".

¹⁰ The framework v1.4 does not support locking of elements.

The actions which are allowed are:

- mask: mask the element, stop the propagation of the state to the upper layer, e.g. mask the HWD element will mask the commsError alarm if it is defined and set the state of the HWD to OK.
- unMask: un-mask the element, re-start the propagation of the state to the upper layer, e.g. un-mask the HWD element will un-mask the commsError alarm if it is defined and re-calculate the state of the HWD from its children's state and commsError alarm.
- acknowledge¹¹: acknowledge any alarms of this element if they exist, e.g. acknowledge the alarms on the HWD element will acknowledge the commsError alarm if it is defined.
- disable¹²: disable the element, the alarms are masked if there is any and the element is not read on the hardware¹³.
- enable¹⁴: enable the element, the alarms are un-masked¹⁵ if there is any.
- mask all: mask the element and mask all the child elements if there is any.
- unMask all: un-mask the element and un-mask all the child elements if there is any.
- acknowledge all: acknowledge the alarm of the element if there is any, and acknowledge all the alarms of its child element if there is any.
- disable all: disable the element and disable all its child elements.
- enable all: enable the element and enable all its child elements.

Most of these actions are executed in a PVSS script: Section 6.1, "dcsMain-HVMain.ctf", by setting the enable and the methodName data point element of the corresponding element or inside the panel "actionPanel", Section 4.2.12, "actionPanel".

Table 2, Table 3, Table 4 and Table 5 summarize the possible actions on the different framework elements implemented by the panel "actionPanel" or by the PVSS script. The action which are allowed are marked as s: executed by the PVSS script dcsMain-HVMain.ctf, Section 6.1, "dcsMain-HVMain.ctf", p: executed inside the

¹¹ it is not possible in PVSS to acknowledge a summary alarm.

¹² some data point element may be de-activated (refer to the PVSS documentation), therefore the PVSS Ev manager will reject any change on the data point element, this will generate errors in the log file.

¹³ this depends on the hardware used, this will be implemented by the IT-CO front-end software in the next release. This is not possible with OPC. However the elements are de-activated, this is generating a lot of error messages in the log file, the data read by the OPC Server are not written in the PVSS data base.

¹⁴ some data point element may be activated (refer to the PVSS documentation), therefore the PVSS Ev manager will accept any change on the data point element, the data point element is automatically set to invalid by the PVSS Ev manager when it is activated, the data point element will be set to valid as soon as the PVSS Ev manager receives a new value on this elements either by a driver, a PVSS script or a graphical interface panel.

¹⁵ alarm will be automatically un-masked when the enable is triggered even if it was masked before.

actionPanel panel itself, Section 4.2.12, "actionPanel", the cell for an action which not implemented is left empty.

Table 2: action summary

action	COMPASS	DETECTOR	SUB_SYSTEM	HWD	AI	AO	AO_F	DI
mask		s	s	s	s		s	s
unMask		s	s	s	s		s	s
acknowledge				s	s		s	s
disable		s	s	s	s	p	s	s
enable		s	s	s	s	p	s	s
mask all	s	s	s	s				
unMask all	s	s	s	s				
acknowledge all	s	s	s	s				
disable all		s	s	s				
enable all		s	s	s				

Table 3: action summary (cont)

action	DO	DO_F	SY127	SY527	SY403	CAEN_HV_ch	CAEN_HV_ch_SP
mask		s	s	s	s	s	s
unMask		s	s	s	s	s	s
acknowledge		s	s	s	s	s	s
disable	p	s	s	s	s	s	s
enable	p	s	s	s	s	s	s
mask all			s	s	s		
unMask all			s	s	s		
acknowledge all			s	s	s		
disable all			s	s	s		
enable all			s	s	s		

Table 4: action summary (cont)

action	CAENHVBoard	PROCS	HV_GROUP	HV_SYSTEM	SERVICE
mask		s	s	s	s
unMask		s	s	s	s
acknowledge		s			
disable	s	s	s	s	s
enable	s	s	s	s	s

Table 4: action summary (cont)

action	CAENHVBoard	PROCS	HV_GROUP	HV_SYSTEM	SERVICE
mask all	s		s	s	s
unMask all	s		s	s	s
acknowledge all	s		s	s	s
disable all	s		s	s	s
enable all	s		s	s	s

Table 5: action summary (cont)

action	FAN_TRAY	FAN_TRAY_SP
mask	s	
unMask	s	
acknowledge	s	
disable	s	
enable	s	
mask all		
unMask all		
acknowledge all		
disable all		
enable all		

3.4.2 HV actions

The HV actions are supported for the framework element of type: HV_GROUP, HV_SYSTEM, SY127, SY403 and SY527. They are triggered by using the panel described in Section 4.2.13, "HV_ActionPanel". These actions are propagated to all the HV elements which are underneath of the HV node element, e.g.: a setState will be propagated to all the HV_GROUP within a HV_SYSTEM.

The actions which are allowed are:

- setState: set all the CAEN HV channels to ON or OFF.
- setRup: set the ramp up speed value to all the CAEN HV channels.
- setRdown: set the ramp down speed value to all the CAEN HV channels.
- setVmax: set the Vmax value to all the CAEN HV channels.
- setTrip: set the trip time value to all the CAEN HV channels.
- setV0: set the settings.v0 value to all the CAEN HV channels.

These actions are executed in a PVSS script: Section 6.1, "dcsMain-HVMain.ctl", by setting the methodName and methodParameter data point element of the corresponding element or inside the panel "HV_ActionPanel", Section 4.2.13, "HV_ActionPanel".

Table 6 summarizes the possible actions on the different framework elements implemented by the panel "HV_actionPanel" or by the PVSS script. The action which are allowed are marked as s: executed by the PVSS script dcsMain-HVMain.ctf, Section 6.1, "dcsMain-HVMain.ctf", p: executed inside the actionPanel panel itself, Section 4.2.13, "HV_ActionPanel", the cell for an action which not implemented is left empty.

Table 6: HV action summary

action	HV_SYSTEM	HV_GROUP	SY127	SY403	SY527
setState	s	s	s	s	s
setRup		s			
setRdown		s			
setVmax		s			
setTrip		s			
setV0		s			

3.5 CAEN HV behavior

The HV elements channel (CAEN_HV_ch or CAEN_HV_ch_SP element type) connected to the IT-CO front-end software have a pre-defined unit. The unit of the readout voltage (vMon) of the readback settings: v0, v1 and the set points: v0 and v1 is set to V (volt). The unit of the readout current iMon, the readback settings: i0, i1 and the set points: i0 and i1 is set to uA (micro-amp). The hardwareAlarm dpElement can have different value:

- none: there is no hardware alarm on that channel.
- ovv: a over voltage hardware alarm was detected on the channel.
- uvv: a under voltage hardware alarm was detected on the channel.
- trip: the channel has tripped.
- xxxx: the value returned by the CAEN interface is not known.

An automatic masking and modification of the alarm limits defined on the vMon, iMon, isOn, hardwareAlarm and isOverCurrent alarm dpElement is also done. Every time the v0 or the on/off state of the CAEN HV channel is modified, the alarm limits of vMon, iMon and isOn is modified according to the v0 or on/off. The alarm are masked during the ramp up or ramp down time and then automatically un-masked after a delay calculated from the the ramping speed and the voltage to reach, 10 sec is added to this calculated delay: this covers the time to send to the hardware and the time to read back the data. This automatic action is re-entrant, e.g.: if an operator change again the voltage during the ramping time, the delay is extended. User masking or un-masking is always a pre-empted action over this automatic behavior: if an operator masks an alarm of a high voltage channel during ramping up or ramping down time, the alarm will remain masked (even after the end of ramping time), and if an operator un-mask the alarm on a high voltage channel, this alarm will be automatically un-mask: this could of course generate alarm messages. If the HV channel is ON, when the v0 is changed or when the on/off is set to ON the vMon alarm limits are set to $v0 \pm \text{threshold}$ and the iMon alarm limits are set to lowIlimit and to highIlimit; otherwise the vMon alarm limits are set to $\pm \text{threshold}$ and the iMon alarm limits are set to 0 and highIlimit. The threshold, lowIlimit and highIlimit are configured in

Section 4.1.24, "CAEN_HV_chAlertConfig" or in Section 4.1.27, "CAEN_HV_ch_SPAlertConfig". If the alarm for these dpElement is set and the config is not defined or corrupted, the threshold is 50 volts, the lowLimit is 0 and the highLimit is 1000 uA. The CAEN HV behavior is implemented by the PVSS script dcsHVAlarm.ctf (Section 6.3, "dcsHV-FTAlarm.ctf"). For the CAEN_HV_ch and CAEN_HV_ch_SP elements all the voltage value are expressed in V (volt) and all the current value are expressed in uA (micro amperes).

3.6 FAN TRAY behavior

An automatic masking defined for temperatureAir, temperaturePS, temperatureBin, fanOk and ftOnOff dpElement and modification of the alarm limits defined on ftOnOff dpElement is done. Every time the settings.ftOnOff, the alarm limit of the readout.ftOnOff is modified according to the settings.ftOnOff and all the alarm defined for the FAN_TRAY element are masked during 10 sec: this covers the time to send to the hardware and the time to read back the data. This automatic action is re-entrant, e.g.: if an operator change again the ftOnOff during the sleeping time, the delay is extended. User masking or un-masking is always a pre-empted action over this automatic behavior: if an operator masks the fan-tray during the sleeping time, the alarm will remain masked (even after the end of sleeping time), and if an operator un-mask the fan-tray during the sleeping time, this alarm will be automatically un-mask: this could of course generate alarm messages. The FAN TRAY behavior is implemented by the PVSS script dcsHVAlarm.ctf (Section 6.3, "dcsHV-FTAlarm.ctf").

3.7 OPC

The OPC connection is supported by the compass framework, OPC Server and OPC Groups are pre-defined inside the framework. A HWD element can be associated to an OPC Server, in that case the 'commsError' is used for the communication errors with the OPC Server. This can be useful to raise an alarm when the OPC Server is disconnected or when the PVSS OPC Client is not running. When PVSS detects that the OPC Server is not connected the 'commsError' of the HWD is set to 1 (OPC Server not connected). When the PVSS OPC Client is not connected, it is set to 2 (PVSS OPC client not connected), otherwise it is set to 0. The data value of the element of type AI, AO, AO_F, DI, DO and DO_F which are connected to the OPC Server will be set to invalid¹⁶ when the 'commsError' dpElement is set to 1 or 2 or when the OPC Item which was entered by the user is not supported by the OPC Server. The mechanism is handle by the PVSS Script dcsDrvMain.ctf (Section 6.2, "dcsDrvMain.ctf"). The OPC Client sets also the data value of an element when the quality¹⁷ attribute of the OPC Item, the element is connected to, is bad. In that latter case the OPC Client maps the quality data to the dpElement user bits, one can look at them via the PARA module. The PVSS OPC client driver resets this invalid bit when it writes new data value in the PVSS database. The compass framework is supporting the connection to the APPLICOM OPC Server and to the Schneider OPC Server¹⁸.

¹⁶ Unfortunately PVSS is not able to get the address configuration if the driver is not started, therefore at startup if the PVSS OPC Client is not connected when the dcsDrvMain.ctf script is started, the elements will not be set to invalid.

¹⁷ Refer to OPC documentation for more detail or to the JCOP web page: <http://itcowww.cern.ch/jcop/subprojects/OPC/NewOPC/Welcome.html>

3.8 DIM

The elements of the framework can be connected to data through the DIM¹⁹ protocol via:

- DIM Client interface using:
 - DIM Service: subscription to a piece of data published by a DIM Server, any change on this data will be given to the framework element. A default value can be returned by DIM when the DIM Client loses the connection with the DIM Server. If the default value is not specified then no default value will be written in PVSS. In both cases the dpElement is set to invalid in the PVSS database and the PVSS user_bit8 config is set. The data can be subscribed by the DIM Client on change or on change and periodically. A time stamp and the quality can be read from the DIM Service, the time stamp and the quality are internally set by DIM in the DIM Server library, but the DIM Server can also set them. This quality is written in the PVSS user_bit1 config of the dpElement. The dpElement is also set to invalid in the PVSS database if the quality is different from 0. One can look at the user bits via the PARA module.
 - DIM Command: subscription to a data published by a DIM Server, a change on the framework element will be transmitted to the DIM Server publishing it.
- DIM Server interface using:
 - DIM Service: publishing of a piece of data, any change on the value of the framework element will be transmitted to all the DIM Clients which have subscribed to that DIM service.
 - DIM Command: publishing of a piece of data, any change on that piece of data by a DIM Client subscribing to it will be transmitted to the framework element.

E.g.: the IT-CO front-end software is able to read and set CAEN high voltage channel value. The readout value such as vMon, iMon, etc. are interfaced to PVSS as DIM Client, DIM Service. The setting value such as v0, i0, etc. are interfaced to PVSS as DIM Client, DIM Command. The IT-CO front-end software is a DIM Server. The enable facility is interfaced to PVSS as DIM Server, DIM Service, the IT-CO front-end software is a DIM Client.

DIM is integrated within PVSS by mean of a PVSS API manager: PVSS00dim, it supports all the DIM element types: integer, float, string, complex structure, etc. The DIM configuration²⁰ can be entered via the PARA module or via the framework configuration panels. A file, which contains the list of DIM Services, DIM commands for a DIM Server or Client interface, has to be created and given to the PVSS00dim process. One DIM interface program: PVSS00dim is dedicated to interface the IT-CO front-end software. This configuration file can be created from the framework panel: Section 4.1.31.3, "#addRmv". The PVSS00dim manager has to be re-started to handle new DIM configuration: Section 4.1.12, "COMPASSmodification". It is possible to use the same program to interface

¹⁸ PVSS is supporting any OPC connection, but in the current COMPASS installation, there is only a connection to APPLICOM and Schneider software via OPC. However, new OPC connection can be easily implemented, the script dcsDrvMain.ctf (Section 6.2, "dcsDrvMain.ctf") should be modified to handle a new added OPC connection: Section 4.1.12, "COMPASSmodification".

¹⁹ refer to the DIM documentation for more detail: [http://delonline.cern.ch/d\\$onl/communications/dim/doc/www/dim.html](http://delonline.cern.ch/d$onl/communications/dim/doc/www/dim.html)

²⁰ It is not possible to attach to a dpElement a DIM complex structure, this feature is not supported by the PARA module and the compass framework configuration panels, however the PVSS00dim manager supports it, and this has to be configured manually by defining it in the startup file of the manager.

other front-end software or to start another DIM-PVSS interface: PVSS01dim (a binary copy of PVSS00dim) with another driver id.

3.9 monitoring the activity of external process

A PROCS element can be used to monitor the activity of an external process, e.g.: this could be the activity of the IT-CO front-end system, the activity of the surveys inside the front-end software²¹, the PVSS00dim process, the activity of the RICH PLC, etc. This is handled by the PVSS script dcsDrvMain.ctl (Section 6.2, "dcsDrvMain.ctl"). The PROCS element should be connected to a data which is changed periodically in a time less than 40 sec²². An alarm is generated if the value of the PROCS element has not changed within the 40 sec. If desired, the dcsDrvMain.ctl script can also kill the PVSS API process manager if this value has not changed, the process should run in the Linux PC. The script is configured to automatically kill whatever PVSS manager²³, e.g. if the PVSS00dim is defined then the script can kill it if it has not refresh the value in PVSS.

3.10 interfacing new element type with the framework

A detector user can create his own data point type and integrate them within the tree in order to received actions or to use common features. The data point type should have the two following data point element:

- enable (PVSS bit type): TRUE (0) means that the element is enable, FALSE (0) means that the element is disable
- methodName (PVSS text type): the following value could be written by framework element or by the "actionPanel":
 - enableAllElement: enable or disable (according to the enable bit²⁴) the element and all its child elements.
 - enable: enable or disable (according to the enable bit²⁵) the element.
 - maskAll: mask the element and its child elements if there is any.
 - mask: mask the element and its alarm if there is any.
 - unMaskAll: un-mask the element and its child elements if there is any.
 - unMask: un-mask the element and its alarm if there is any.
 - acknowledgeAllAlarm: acknowledge all the alarms of the element and its child elements if there is any.
 - acknowledge: acknowledge all the alarms of the element if there is any.

Scripts shall also be provided to handle the functions described above and the state calculation. The run-time panels and configuration panels shall also be done by the user to configure and operate the new elements. These elements should also be connected to the framework elements. These elements can be integrated at the DETECTOR level in the logTree hierarchy, or at the SUB_SYSTEM level. These user defined elements should be

²¹ refer to the documentation on the IT-CO front-end software for the meaning of survey.

²² The 40sec is fixed value, hard-coded in the PVSS script. It can be easily changed.

²³ In the current release v1.4, the script can only killed the PVSS00dim process, it can be easily modified to be able to kill other processes.

²⁴ The enable bit and the methodName are set at the same time.

²⁵ The enable bit and the methodName are set at the same time.

removed from the DETECTOR log tree list or SUB_SYSTEM list before it is deleted, otherwise reference errors will be incorporated within the tree hierarchy²⁶ (the framework libraries Section 5, "Framework Libs" can be used to add or remove elements from a list or from a summary alarm list). If a user element is an alarm summary, one element should be entered in the alarm list otherwise the alarm summary is the summary alarm of all the data point if the PVSS system²⁷. The gDummyAlert can be used if one does not know the content of the list.

3.11 Customized user panel

The detector developer can create his own panel, he can if he wishes incorporate the reference panels (Section 4.3, "Reference panels"), call the modification panels (Section 4.1, "Modification panels") or run-time panels (Section 4.2, "Run-time panels"). The same panels can be used during maintenance or run-time phase, and it could be useful to incorporate in all the panels a banner showing the status of all the detectors even if this panel will be used during the maintenance phase. This will allow later on to modifying the banner reference without modifying all the user panels. Whatever reference panel, incorporated in a user panel, can be modified, e.g. the behavior can be changed, the scripts can be removed, etc. All the libraries described in Section 5, "Framework Libs" can also be used. New PVSS scripts can be created, they should be started by PVSS0xctrl, where x is the driver number. The functions of the libraries cannot be re-used, they have to be copied in the script²⁸. This driver number should be greater than 5. The number 1 to 5 are reserved for the framework.

3.12 Element of a detector which are coming from service

Data of some detectors may come from a central service: e.g. gas system, rack cooling, etc. or may be read by a common software: e.g. IT-CO front-end software, OPC, etc. However even if these data are coming from a central or common entity, each detector may want to operate, control or modify them. The SERVICE element type can be used to represent this common service or readout. Elements can be created in the SERVICE²⁹ sub-tree and can be placed later within the DETECTOR logical tree hierarchy. This will allow a detector expert to modify, operate or control this element even if its hardware connection is coming from a SERVICE. The element has to be created by giving the name of the DETECTOR element it could belong to. The element name will therefore starts with the DETECTOR name and will be placed in the SERVICE tree.

4 Framework Panels

The most important part of the panels are explained: they are referenced as #xyz.

²⁶ The reference errors can be corrected via the PARA module by editing the list of elements and the alarm list.

²⁷ This is PVSS bug.

²⁸ In the PVSS version 2.10.1.3 used in COMPASS it is not possible to share a library between scripts started by PVSS00ctrl and panels started by PVSS00ui, therefore the common functions have to be duplicated in every scripts.

²⁹ It is not possible to use a DETECTOR element in that case because it is not allowed to have a reference of the elements from another detector. Not having a reference means not being able to operate it.

Most of scripts behind the panel elements (push buttons, text fields, etc.) are implemented behind 'working function' e.g: dpConnect() which means that the value(s) of this element is refreshed automatically according to the changes of the datapoint element. The panel is refreshed automatically when a new value is received by PVSS. Some panel elements may contain script for controlling other panel element(s).

There are three kinds of panels:

- modification: used by the operator to create, delete instances of the elements type (Section 3, "Framework elements"), to change the tree hierarchy and to add, delete or modify the configuration of the of the elements: Section 4.1, "Modification panels". They can also be used in user defined panels. The PVSS implementation is hidden to the operator.
- run-time: used at run-time by the operators to view the data value: Section 4.2, "Run-time panels", to browse the hierarchical tree or to operate the elements. The run-time panel can called other run-time panels. They can also be used in user defined panels.
- reference also called symbol: are ready made "components" which can be incorporated in a user defined panel: Section 4.3, "Reference panels". They show the most important parameter(s) of the element, they generally call run-time panel.

Alarm color convention for the leaf elements or state color for the node elements, PVSS dp connection are implemented within the run-time and reference panels. The developer has just to select the data point he wants to use with these panels.

4.1 Modification panels

The modification panels are used to create new instances of the framework element type, to modify the configuration of the existing elements, to delete them or to change the tree hierarchy.

The elements can be created within the detector sub-trees or within the service tree. It is better to create³⁰ the element in the service tree if they are provided by a common service or read by a common readout software. If the element should be referenced in a detector later on, a detector should be selected instead of the service name, this will allow the detector expert to operate the element. E.g.: two PLCs are used in COMPASS for different gas system, the collection of the data is done via a common service: an OPC Server. A service can be created which represents the common OPC readout and each element belonging to each detector can be placed in this service element. This could be helpful for providing a fast diagnostic in case of problems on a common service when data are spread in different detectors. An state different from OK at the service quality level will warn the operator that the quality of the data coming from that service might not be good, this a way to be sure that the data are correct. It is not possible to set the quality of the data for element which are commonly provided to the experiment, this will put an overload of the system.

The DCS system name is shown in most of the modification panels, some actions on the alarm are not allowed in the PRODUCTION system (refer to Section 2, "Description" for more detail).

³⁰ In the version v1.4, it is only possible to create in a SERVICE element of type HWD and PROCS.

Only one panel per kind of element can be opened at the same time to avoid to have too many panels on the screen. There is no need to close the panel to open the modification panel of an element of the same type, this is done by PVSS, the content of the panel is modified to show the new selected element.

The modification panels are called by the COMPASSmodification panels, the DETECTORmodification panels, etc. However it is possible to call them from user customized panels, the appropriate \$-parameter has to be given to the panel.

Alarms

The user can add an alarm (Section 3.3, "Alarms and states") feature for an observed value via a modification panel. The category³¹ of the alarm can be one of next three different possibilities:

- Warning: W
- Fault: F
- Fatal: FTL

Warning is the lowest level of alarm, the short sign for warning is 'W'. One level higher category from Warning is Fault with a short sign 'F'. Fatal is the highest category with a short sign 'FTL'. Alarms are acknowledgeable only on the transition from good (or ok) to WARNING, FAULT or FATAL: this is the 'CAME PVSS' terminology. The AlertScreen (Section 4.2.29, "Alert panel") can be used to see the active alarms, different filters can be applied based on the category. An alarm will appear only once in the AlertScreen even if it is oscillating inside and outside the alarm limits³².

The following arguments have to be defined for an analogue value alarm:

Alert category.	Category of the generated alarm.
Upper limit.	Upper treshold value. If the value increases above this value then an alarm with the defined alert category will be generated.
Lower limit.	Lower treshold value. If the value decreases below this value then an alarm with the defined alert category will be generated.
Upper bad-range text.	Text which appears when value goes above the defined upper limit.
ok-range text.	Text which appears when value is between the upper and lower limits.
Lower bad-range text.	Text which appears when value goes under the defined lower limit.

The following arguments have to be defined for a digital value alarm:

Alert category.	Category of the generated alarm.
Valid range.	Position of "dot" tells which one (either TRUE or FALSE) is the good range. E.g. if the position of "dot" is on TRUE choice, the

³¹ The category of the alarm is stored in an PVSS alert class, alarm class colors can be easily changed.

³² This is a PVSS feature. In the current release this is configured within the alert class used by the framework, this can be easily changed to show the old alert. However in that latter case, the old alert will remain active in the AlertScreen until the operator acknowledge them, and this can be confusing.

alarm will be generated when value equals FALSE. TRUE is equivalent to 1 and FALSE is equivalent to 0.

True text. Text which appears when value equals TRUE.

False text. Text which appears when value equals FALSE.

Peripheral address

The PVSS peripheral address is the way of defining to which source is connected the dpElement. The following possible choice are supported by the framework elements:

None. no peripheral address, the element is used as an internal data and the dpElement can be changed by the user.

OPC. the dpElement is connected to an OPC Item published by an OPC Server (Figure 3). The OPC Server should be selected, and the OPC Item name should be entered. The OPC group and driver number are automatically pre-selected depending on the type of the element: input or output³³.

DIM. the dpElement is connected to a data provided by the DIM interface (Figure 4, Figure 5 and Figure 6). The driver number should be entered, it corresponds to the PVSS00dim manager number. Different possibilities are available: DIM Server, DIM Client, DIM Service, DIM Command (Section 3.8, "DIM")³⁴. Leave the 'default value' box empty when one does not want any default value when the connection is lost between the PVSS DIM Client interface and the DIM Server providing the DIM Service data. The time stamp and quality for the DIM Client interface, DIM Service data is selected with the 'time stamp' check box (Figure 6). If the data can be read by the IT-CO front-end software (use default FrontEnd check box), e.g.: CAEN HV channel, etc., the DIM configuration is pre-selected depending on the type of element (Figure 6).

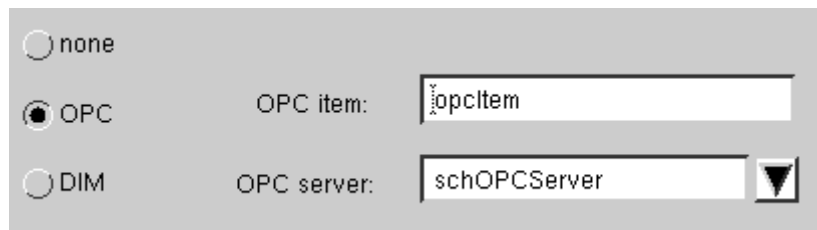


Figure 3: OPC configuration.

³³ PVSS has a limitation in the quantity of element connected as an output. A PVSS driver can only accept one dpElement connected to an output reference. A PVSS OPC Client driver can only accept one dpElement connected to an output OPC Item per OPC group. In the framework, the OPC groups are pre-defined. There are two groups per OPC Server, the names are _OPCServerNameIn for input elements and _OPCServerNameOut for output elements, where OPCServerName is the name of the PVSS OPC Server data point element.

³⁴ DIM Client, DIM Command are considered to be output data, therefore it is not possible to have the same DIM Command connected to more than one dpElement on the same driver.

none Server DIM command:
 OPC Client drvNr:
 DIM DIM service
 DIM command

Figure 4: DIM configuration.

none Server use default FrontEnd
 OPC Client
 DIM DIM service DIM service:
 DIM command
 drvNr:

Figure 5: DIM configuration (cont.).

none Server use default FrontEnd
 OPC Client
 DIM DIM service DIM service:
 DIM command
 On change
 At regular intervals Secs
 default value: time stamp
 drvNr:

Figure 6: DIM configuration (cont.).

Archive

Element of the framework can be archive. The following choice are supported by the framework elements:

- no.** the data is not archived.
- yes with old/new.** the data is archived if the newValue is different that the oldArchivedValue.

yes with old/new and time dependent. the data is archived if the newValue is different than the oldArchivedValue or periodically.

yes with deadband. the data is archived if the $\text{abs}(\text{newValue} - \text{oldArchivedValue}) > \text{deadband}$. This configuration is not used for data point element of type PVSS bit, even if the yes with deadband is selected (e.g. for a CAEN_HV_ch element, the PVSS bit dpElement archive config is set to yes with old/new).

yes with deadband and time dependent. the data is archived if the $\text{abs}(\text{newValue} - \text{oldArchivedValue}) > \text{deadband}$ or periodically. This configuration is not used for data point element of type PVSS bit, even if the yes with deadband is selected (e.g. for a CAEN_HV_ch element, the PVSS bit dpElement archive config is set to yes with old/new and time dependent).

Conversion

The data can be automatically converted. This feature is supported by the elements of type AI, AO and AO_F connected to an OPC Server³⁵. The peripheral address should be created before adding the conversion config.

None. no conversion.

polynomial. the conversion is a polynomial curve.

curve. a series of x and y coordinate value have to be entered, a linear extrapolation between each pair of x, y coordinates is done to convert the raw value.

Pv_range

The pv_range is used to set a minimum and maximum allowed value that the user can enter for the dpElement. This feature is supported by the elements of type CAEN_HV_ch and CAEN_HV_ch_SP. A value outside the pv_range will not be set³⁶.

Smoothing

The smoothing is used to filter the data to avoid too many changes in the PVSS data base: e.g. to avoid a change of 0.01 degC of a pt100 sensor. This feature is supported by the elements of type AI, AO_F connected to an OPC Server³⁷.

None. no smoothing.

yes with deadband. the data is send to the Ev manager if the $\text{abs}(\text{newValue} - \text{oldSentValue}) > \text{deadband}$.

yes with deadband and time dependent. the data is send to the Ev manager if the $\text{abs}(\text{newValue} - \text{oldSentValue}) > \text{deadband}$ or periodically.

yes with old/new. the data is send to the Ev manager if the newValue is different than the oldSentValue.

³⁵ This config is only supported by PVSS driver, e.g. OPC. In the release v1.4, the PVSS DIM Client interface is not a PVSS driver, it is a PVSS API manager.

³⁶ The root user is a particular user in PVSS, he has all the rights. PVSS is not checking the pv_range for the root user. However the framework panels checks that the value is inside the pv_range.

³⁷ This config is only supported by PVSS driver, e.g. OPC. In the release v1.4, the PVSS DIM Client interface is not a PVSS driver, it is a PVSS API manager.

yes with old/new and time dependent. the data is send to the Ev manager if the newValue is different than the oldSentValue or periodically.

4.1.1 conversionPoly

Two kinds of conversion functions (Figure 7, Section , "Conversion") can be used in the framework, refer to the PVSS documentation for more information on conversion configuration parameter. This configuration can be used when a driver is giving raw data to the Event manager, e.g.: the APPLICOM OPC is reading analogue value and returning the read value as ADC count, a conversion is then needed to have the value in engineering units. The conversion config is only supported for elements connected to data via an OPC Server. Two type of conversion are supported by the compass framework:

- polynomial up to the fourth degree.
- curve: up to 5 x and y coordinates can be entered, a linear extrapolation between each pair of x, y coordinates is done to convert the raw value.



Figure 7: conversionPoly panel.

4.1.2 archiveSet

Four kinds of archiving are supported (Figure 8), Section , "Archive", for the analogue data, only two kind of archiving are supported for boolean data. This panel takes care of the type of the dp and set the archive configuration to the dpElement of the data point.



Figure 8: archiveSet panel.

4.1.3 smoothSet

Four kinds of smoothing are supported (Figure 9), Section , "Archive", for the analogue data, only two kind of smoothing are supported for boolean data.

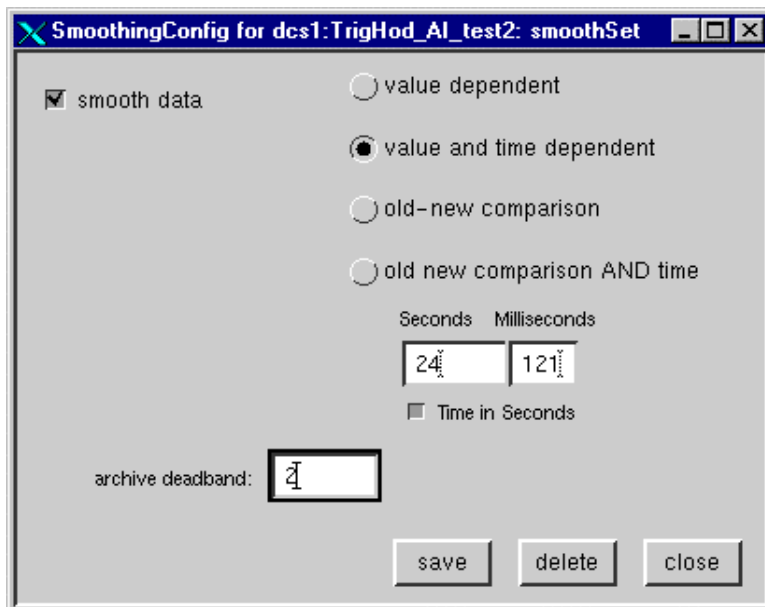


Figure 9: smoothSet panel.

4.1.4 Almodification

This panel (Figure 10 and Figure 11) is called from the COMPASSmodification, HWDmodification and DETECTORmodification panel. It is used to configure the AI element.

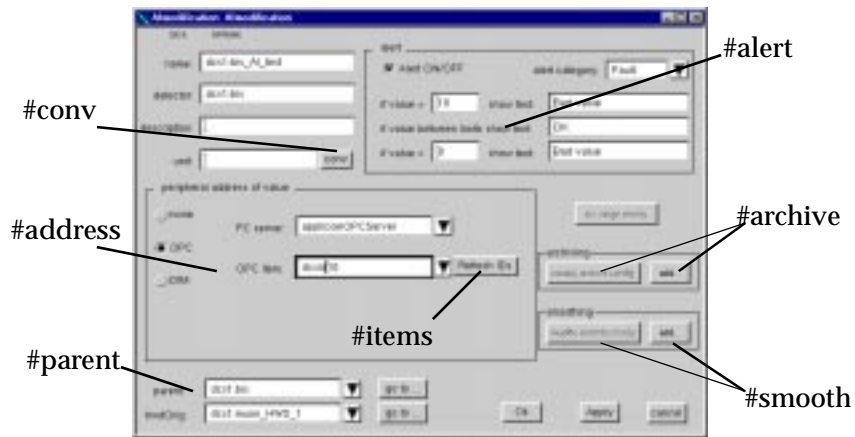


Figure 10: Almodification panel: OPC config.

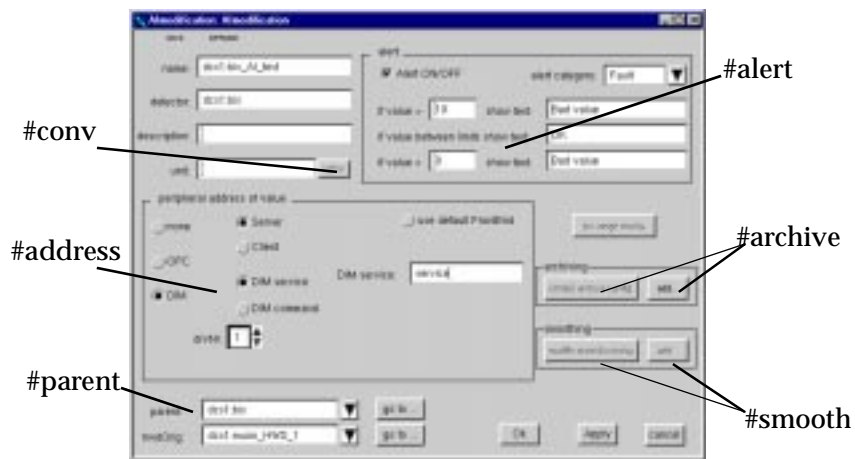


Figure 11: Almodification panel DIM config.

4.1.4.1 #alert

Alarm can be added or delete in the OFFLINE system by selecting or un-selecting the check box. Alarm limit and category can be changed.

4.1.4.2 #conv

A conversion config (Section 4.1.1, "conversionPoly") can be added or deleted, this is only allowed in the case of a connection via OPC.

4.1.4.3 #address

The peripheral address config can be added, this defines the source of the data (Section , "Peripheral address"). The drvId is only used for the DIM Connection.

4.1.4.4 #archive

The archive config of the 'value' dpElement can be added deleted or modified (Section 4.1.2, "archiveSet").

4.1.4.5 #parent

There are two kinds of parent for the AI element:

- logical location of the AI element.
- hardware location of the AI element.

4.1.4.6 #smooth

A smoothing config for the 'value' dpElement can be added deleted or modified (Section 4.1.3, "smoothSet"), this is only allowed in the case of a connection via OPC.

4.1.4.7 #items

The 'Refresh IDS' button triggers the OPC Server to refresh the list of OPC Items. The list is show in the combo box: OPC Items.

4.1.5 AOmodification

This panel (Figure 12) is called from the COMPASSmodification, HWDmodification and DETECTORmodification panel. It is used to configure the AO element.

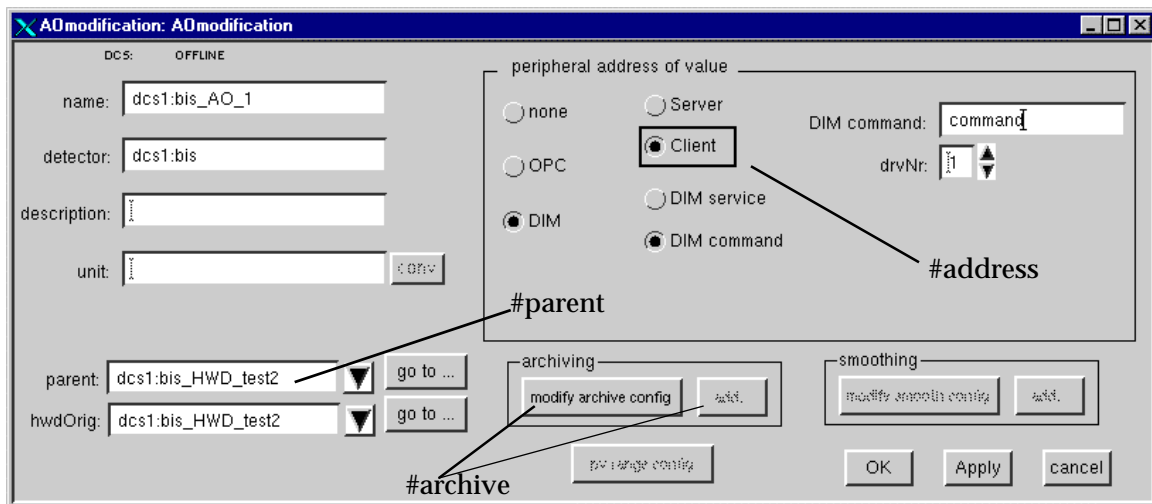


Figure 12: AOmodification panel.

4.1.5.1 #archive

The archive config of the 'value' dpElement can be added deleted or modified (Section 4.1.2, "archiveSet").

4.1.5.2 #address

The peripheral address config can be added, this defines the source of the data (Section , "Peripheral address"). The drvId is only used for the DIM Connection.

4.1.5.3 #parent

There are two kinds of parent for the AO element:

- logical location of the AO element.
- hardware location of the AO element.

4.1.6 AO_Fmodification

This panel (Figure 13) is called from the COMPASSmodification, HWDmodification and DETECTORmodification panel. It is used to configure the AO_F element.

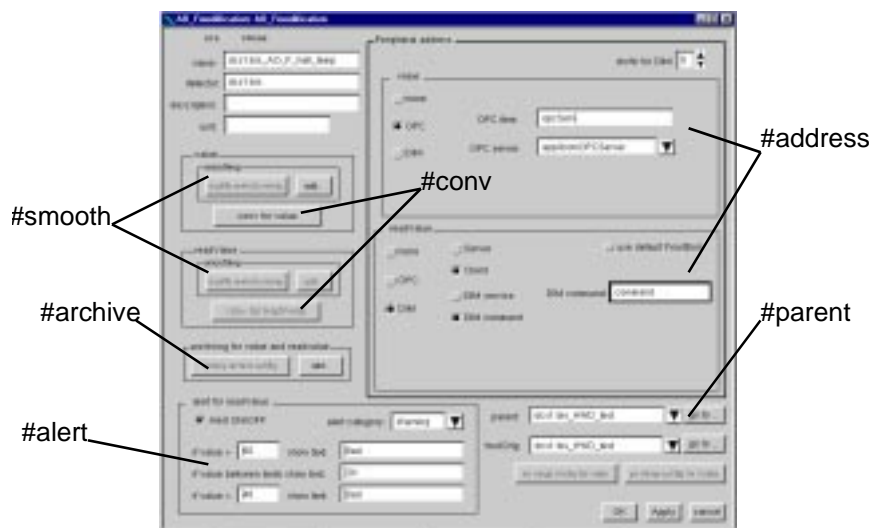


Figure 13: AO_Fmodification panel.

4.1.6.1 #alert

Alarm of the readValue dpElement can be added or delete in the OFFLINE system by selecting or un-selecting the check box. Alarm limit and category can be changed.

4.1.6.2 #conv

A conversion config (Section 4.1.1, "conversionPoly") can be added or deleted, this is only allowed in the case of a connection via OPC, for both value³⁸ and readValue dpElement.

³⁸ in version 1.1, this feature is not supported.

4.1.6.3 #address

The peripheral address config can be added, this defines the source of the data (Section , "Peripheral address"). The drvId is only used for the DIM Connection and is the same for value and readValue dpElement.

4.1.6.4 #archive

The archive config of the 'value' and 'readValue' dpElement can be added deleted or modified (Section 4.1.2, "archiveSet").

4.1.6.5 #parent

There are two kinds of parent for the AO_F element:

- logical location of the AO_F element.
- hardware location of the AO_F element.

4.1.6.6 #smooth

A smoothing config for the 'value' and 'readValue' dpElement can be added deleted or modified (Section 4.1.3, "smoothSet"), this is only allowed in the case of a connection via OPC.

4.1.7 DImodification

This panel (Figure 14) is called from the COMPASSmodification, HWDmodification and DETECTORmodification panel. It is used to configure the DI element.

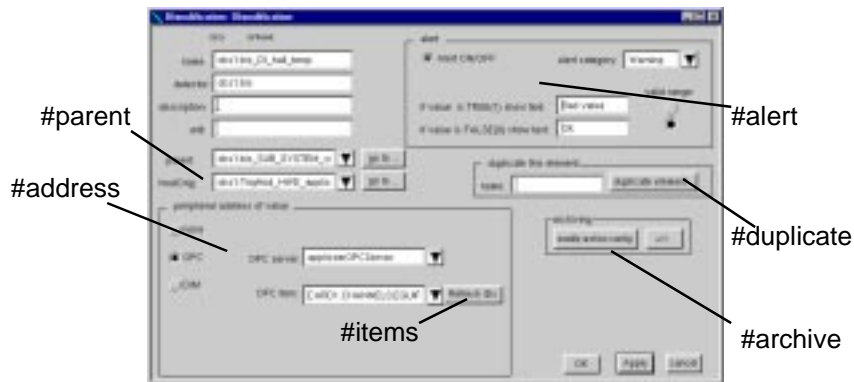


Figure 14: DImodification panel.

4.1.7.1 #alert

Alarm can be added or delete in the OFFLINE system by selecting or un-selecting the check box. Alarm category and valid state alarm can be changed.

4.1.7.2 #archive

The archive config of the 'value' dpElement can be added deleted or modified (Section 4.1.2, "archiveSet").

4.1.7.3 #parent

There are two kinds of parent for the DI element:

- logical location of the DI element.
- hardware location of the DI element.

4.1.7.4 #address

The peripheral address config can be added, this defines the source of the data (Section , "Peripheral address"). The drvId is only used for the DIM Connection.

4.1.7.5 #items

The 'Refresh IDS' button triggers the OPC Server to refresh the list of OPC Items. The list is show in the combo box: OPC Items.

4.1.7.6 #duplicate

The duplicate button creates a clone of this element (same configs).

4.1.8 DOmodification

This panel (Figure 15) is called from the COMPASSmodification, HWDmodification and DETECTORmodification panel. It is used to configure the DO element.

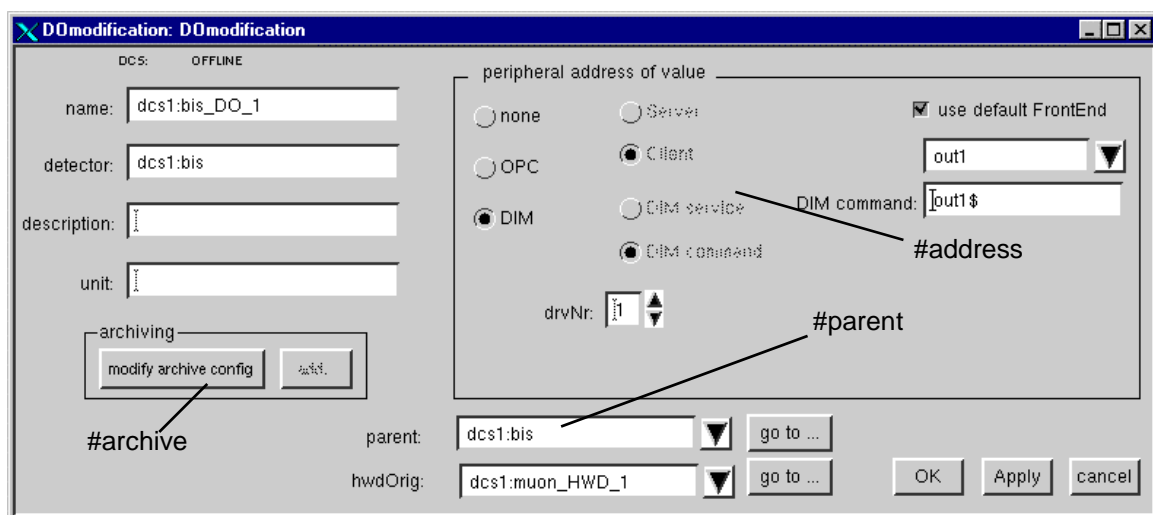


Figure 15: DOmodification panel.

4.1.8.1 #parent

There are two kinds of parent for the DO element:

- logical location of the DO element.
- hardware location of the DO element.

4.1.8.2 #address

The peripheral address config can be added, this defines the source of the data (Section , "Peripheral address"). The drvId is only used for the DIM Connection.

4.1.8.3 #archive

The archive config of the 'value' dpElement can be added deleted or modified (Section 4.1.2, "archiveSet").

4.1.9 DO_Fmodification

This panel (Figure 16) is called from the COMPASSmodification, HWDmodification and DETECTORmodification panel. It is used to configure the DO_F element.

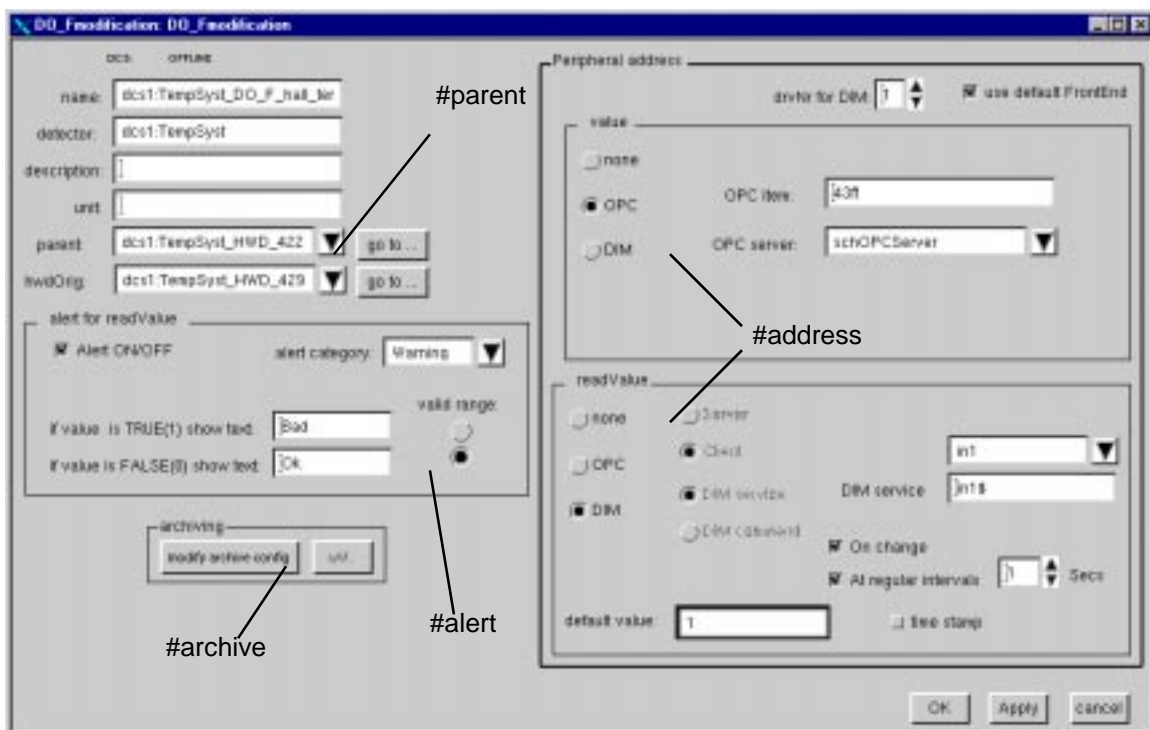


Figure 16: DO_Fmodification panel.

4.1.9.1 #parent

There are two kinds of parent for the DO_F element:

- logical location of the DO_F element.
- hardware location of the DO_F element.

4.1.9.2 #alert

Alarm of the readValue dpElement can be added or delete in the OFFLINE system by selecting or un-selecting the check box. Alarm category and valid state alarm can be changed.

4.1.9.3 #address

The peripheral address config can be added, this defines the source of the data (Section , "Peripheral address"). The drvId is only used for the DIM Connection and is the same for value and readValue dpElement.

4.1.9.4 #archive

The archive config of the 'value' and 'readValue' dpElement can be added deleted or modified (Section 4.1.2, "archiveSet").

4.1.10 HWDmodification

This panel (Figure 17) is called from the COMPASSmodification, HWDmodification and DETECTORmodification panel. It is used to configure the HWD element.

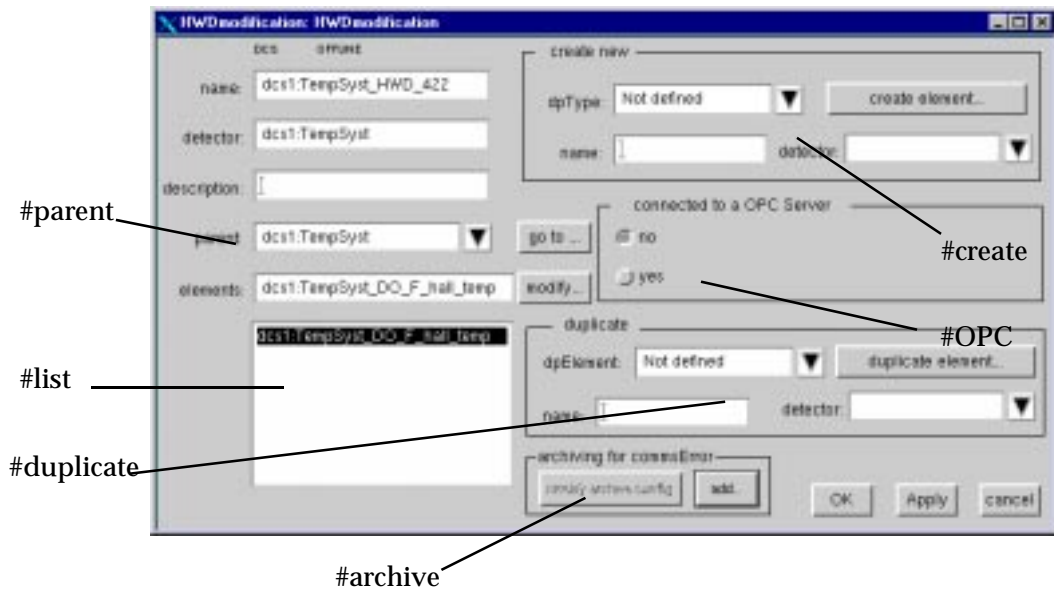


Figure 17: HWDmodification panel.

4.1.10.1#OPC

A HWD element can be associated to an OPC Server, in that case the 'commsError' is used for the communication errors with the OPC Server (Section 3.7, "OPC").

4.1.10.2#parent

A HWD element has only one parent: a HWD or a detector. This represents the hardware location of the element.

4.1.10.3#list

This is the list of child elements of the HWD, a double click on one line opens the corresponding modification panel.

4.1.10.4#create

New elements can be created from the HWD panel, automatically the hwdOrig is assigned to the HWD from which is initiated the creation. A detector or a service has to be selected. The name of the newly created elements will start with the detector or service name.

4.1.10.5#duplicate

This is to clone an element belonging to the hwd. A detector or a service has to be selected. The name of the newly created elements will start with the detector or service name.

4.1.10.6#archive

The archive config of the 'commsError' dpElement can be added deleted or modified (Section 4.1.2, "archiveSet").

4.1.11 DETECTORmodification

This panel (Figure 18) is called from the COMPASSmodification panel. It is used to configure the DETECTOR element. The parent of a detector is gCompass (top of the hierarchical tree) and it cannot be changed. The DETECTOR has two internal tree: the hardware tree used for the hardware location of the elements and the logical tree. The hardware tree represent the hardware view of the detector (list of HWD, high voltage crate, etc.) and the logical tree the view of the physicists operations.

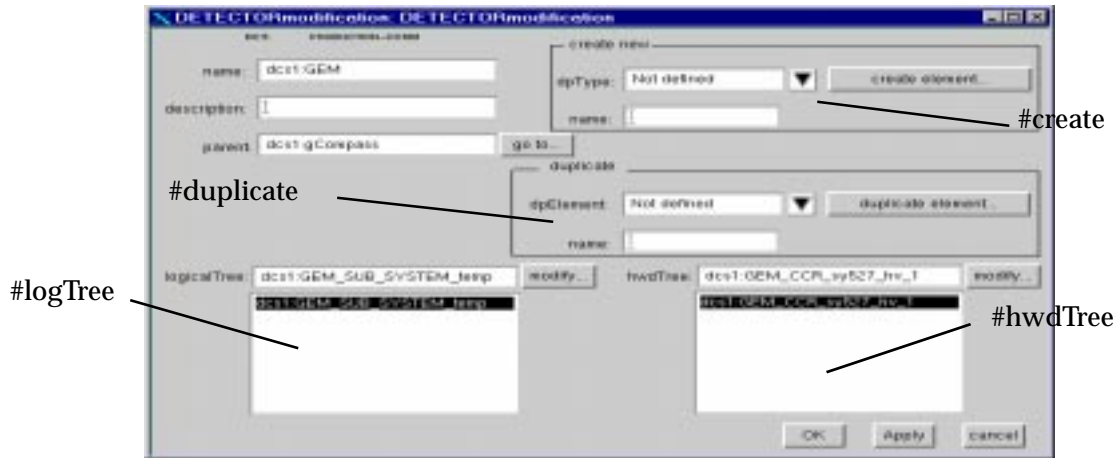


Figure 18: DETECTORmodification panel.

4.1.11.1#logTree

This is the list of the elements of the logical tree. A double click on a line of this list opens the corresponding modification panel.

4.1.11.2#hwdTree

This is the list of the elements of the hardware tree. A double click on a line of this list opens the corresponding modification panel.

4.1.11.3#create

New elements can be created from the DETECTOR panel, automatically the parent is assigned to the DETECTOR from which is initiated the creation. The elements of type AI, AO, AO_F, DI, DO, DO_F, SUB_SYSTEM, HV_SYSTEM and HV_GROUP are placed in the logical tree and the elements of type HWD are placed in the hardware tree. SY127, SY403, SY527, CAEN_HV_ch and CAEN_HV_ch_SP cannot be created from the DETECTORmodification, the ImportHVConfig or HVConfig button (Section 4.1.39, "ImportHVConfig" and Section 4.1.41, "HVConfig") has to be used instead.

4.1.11.4#duplicate

This is to create a clone element.

4.1.12 COMPASSmodification

This panel (Figure 19) can be called from other panels. It is used to configure the elements and the tree hierarchy.

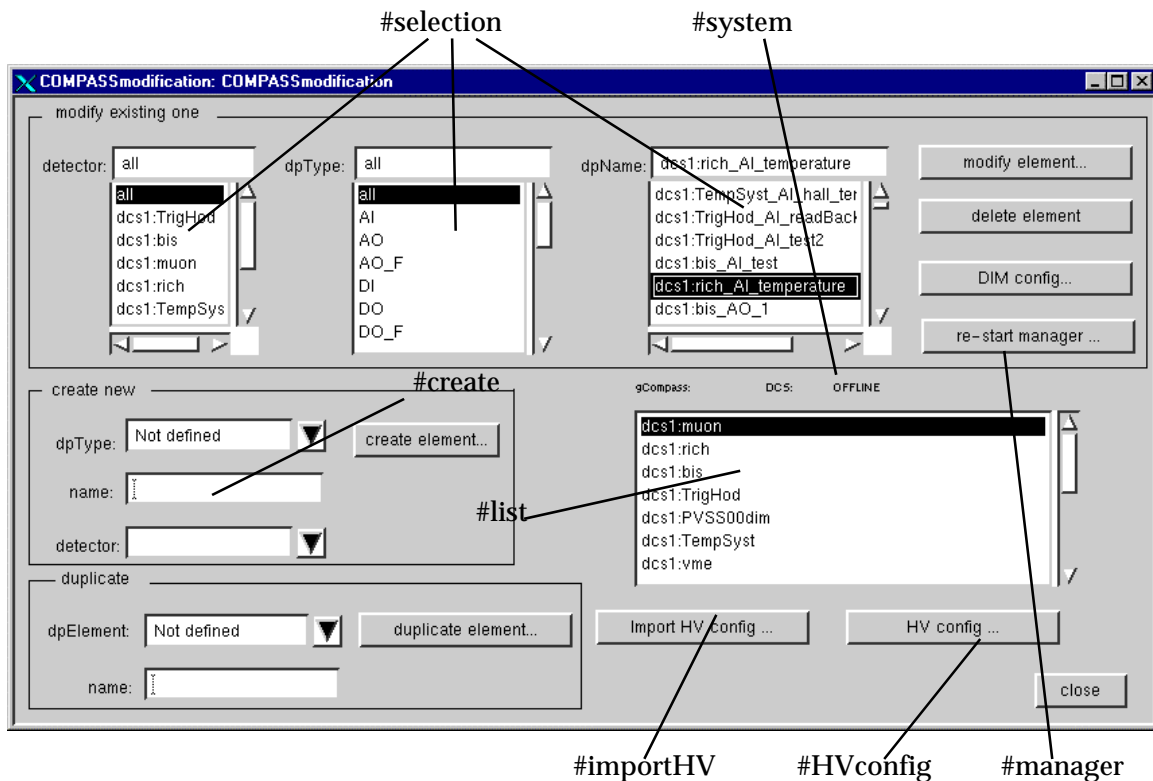


Figure 19: COMPASSmodification panel.

4.1.12.1#create

New elements can be created the parent and/or the hwdOrig³⁹ are set to empty. SY127, SY403, SY527, CAEN_HV_ch and CAEN_HV_ch_SP cannot be created from the DETECTORmodification, one has to use instead the ImportHVConfig or HVConfig button (Section 4.1.39, "ImportHVConfig" and Section 4.1.41, "HVConfig").

4.1.12.2#selection

Different filters are support to allow a fast access to the data.

4.1.12.3#manager

Whenever the peripheral address config is changed or the elements are deleted or modified the DIM config should be created and edited (Section 4.1.31.3, "#addRmv"). Clicking on this button will stop all the managers.

4.1.12.4#list

This is the list of child elements of the gCompass top tree hierarchy.

³⁹ This depends on the type of the element, e.g. a HWD element has not a hwdOrig link, it just has a parent link.

4.1.12.5#system

This is the system name. When this panels is called within the PRODUCTION system, the 'delete element' button is not accessible.

4.1.12.6#importHV

HV configuration already done for the IT-CO front-end software can be imported within PVSS (Section 4.1.39, "ImportHVConfig").

4.1.12.7#HVconfig

The HV can also be configured by using the JCOP framework elements (Section 4.1.41, "HVConfig").

4.1.13 PROCSmodification

This panel (Figure 20) is called from the COMPASSmodification panel. It is used to configure the PROCS element.

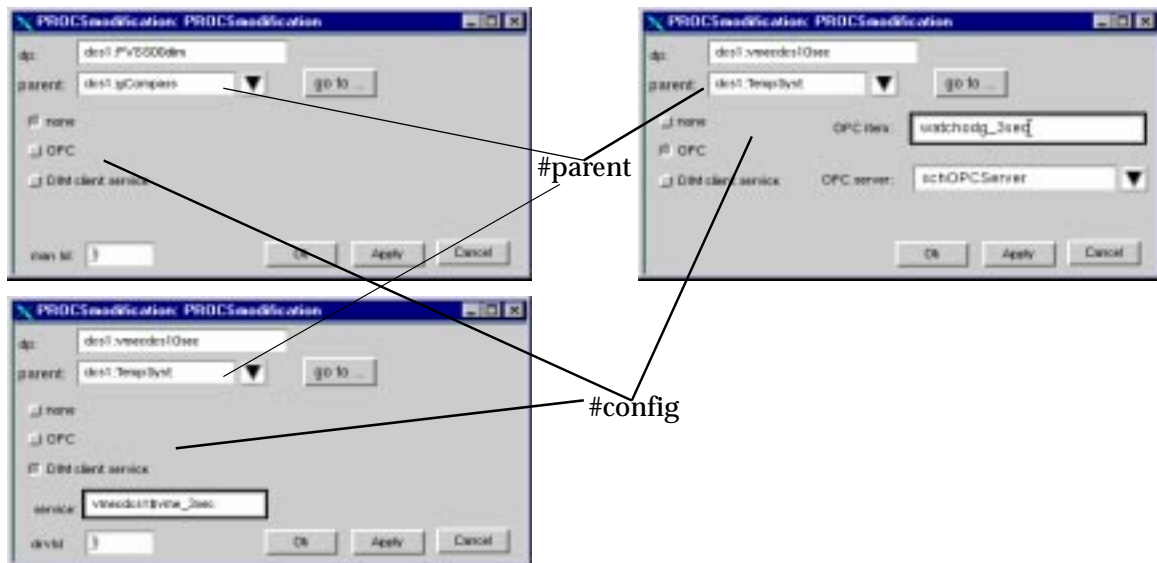


Figure 20: PROCSmodification panel.

A PROCS element can be used to monitor the activity of an external process (Section 3.9, "monitoring the activity of external process"), e.g.: this could be the activity of the IT-CO front-end system, the activity of the surveys inside the front-end software⁴⁰, PVSS00dim, etc. This is handled by the PVSS script dcsDrvMain.ctf (Section 6.2, "dcsDrvMain.ctf"). The #config is used to set from which source is generated the data. The manId is used for the PVSS00dim process⁴¹, it represents the API manager driver id, and it has to be greater than

⁴⁰ refer to the documentation on the IT-CO front-end software for the meaning of survey.

⁴¹ In the current release v1.4, the script can only killed the PVSS00dim process, it can be easily modified to be able to kill other processes.

0, for the other PROCS elements the manId should be 0. The #parent is a SERVICE element or gCompass.

4.1.14 SY127modification

This panel (Figure 21) is called from the COMPASSmodification panel. It is used to configure the SY127 peripheral address and alarm config of the SY127 element read by the IT-CO front-end software. An #archive config can be added or modified (Section 4.1.2, "archiveSet") to the errorCount dpElement of the SY127 crate. The errorCount is the quantity of consecutive errors while reading data from the SY127 crate via the CAENnet bus. The #alert calls the SYAlertConfig panel (Section 4.1.16, "SYAlertConfig") to set an alarm on the errorCount.

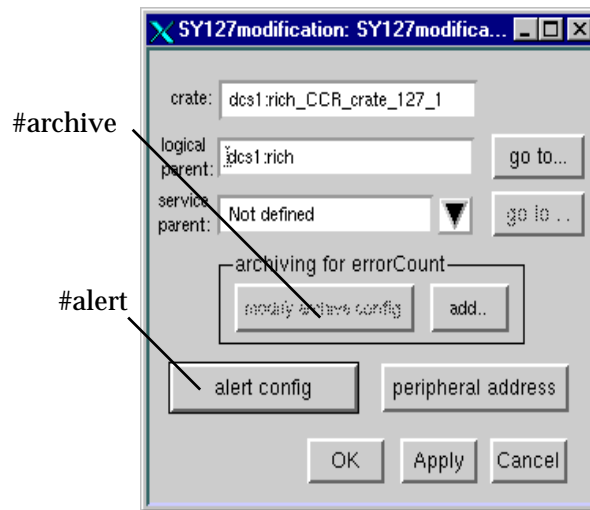


Figure 21: SY127modification panel.

4.1.15 SY127PeriphConfig

This panel (Figure 22) is called from the SY127modification panel (Figure 21). It is used to configure the peripheral address config of the SY127 crate when it is read by the IT-CO front-end software. The drvId is used in the case of DIM. OPC is not supported

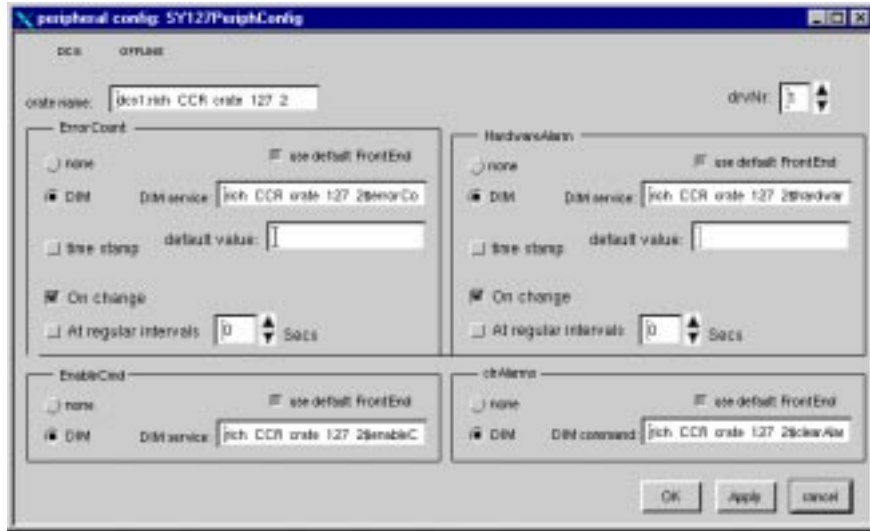


Figure 22: SY127PeriphConfig panel.

4.1.16 SYAlertConfig

This panel (Figure 23) is used to configure the alarm config of the errorCount of the SY127, SY403 and SY527 crate when it is read by the IT-CO front-end software. The errorCount is the quantity of consecutive errors while reading data from the SY127, SY403 or SY527 crate via the CAENnet bus, an alarm can be set if desired.

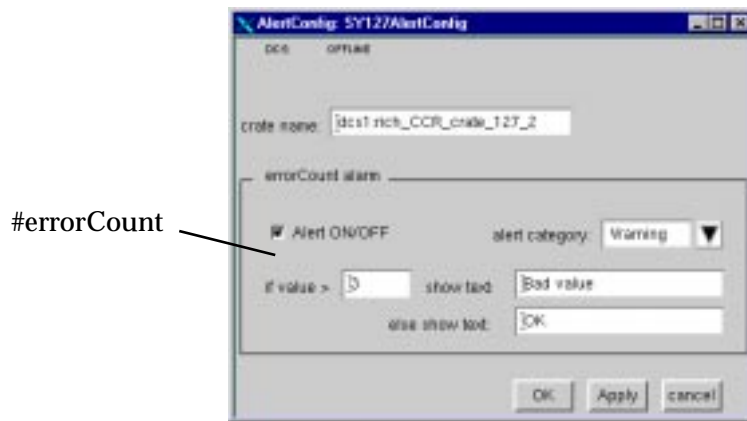


Figure 23: SYAlertConfig panel.

4.1.17 SY403modification

This panel (Figure 24) is called from the COMPASSmodification panel. It is used to configure the SY403 peripheral address and alarm config of the SY403 element read by the IT-CO front-end software. An #archive config can be added or modified (Section 4.1.2, "archiveSet") to the errorCount dpElement of the SY403 crate. The errorCount is the quantity of consecutive errors while reading data from the SY403 crate via the CAENnet

bus. The #alert calls the SYAlertConfig panel (Section 4.1.16, "SYAlertConfig") to set an alarm on the errorCount.

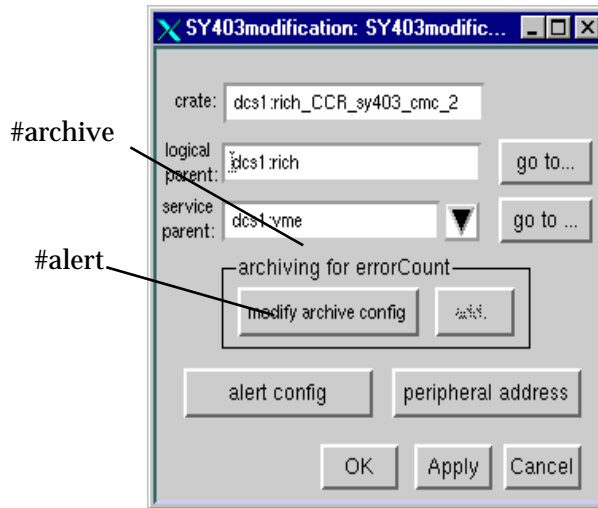


Figure 24: SY403modification panel.

4.1.18 SY403PeriphConfig

This panel (Figure 25) is called from the SY403modification panel (Figure 26). It is used to configure the peripheral address config of the SY403 crate when it is read by the IT-CO front-end software. The drvId is used in the case of DIM. OPC is not supported.

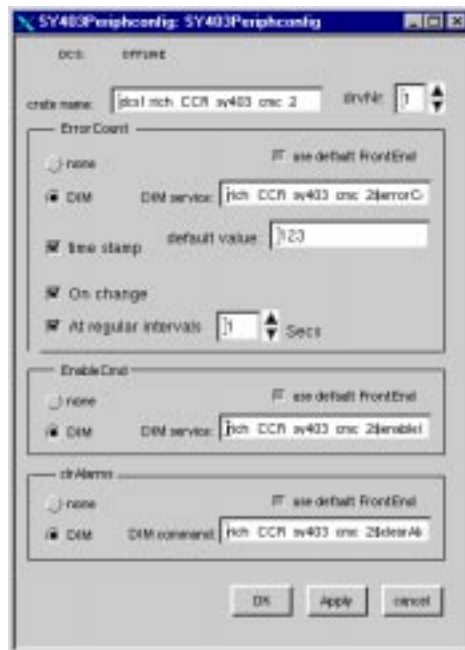


Figure 25: SY403PeriphConfig panel.

4.1.19 SY527modification

This panel (Figure 26) is called from the COMPASSmodification panel. It is used to configure the SY527 peripheral address and alarm config of the SY527 element read by the IT-CO front-end software. An #archive config can be added or modified (Section 4.1.2, "archiveSet") to the errorCount dpElement of the SY527 crate. The errorCount is the quantity of consecutive errors while reading data from the SY527 crate via the CAENnet bus. The #alert calls the SYAlertConfig panel (Section 4.1.16, "SYAlertConfig") to set an alarm on the errorCount.

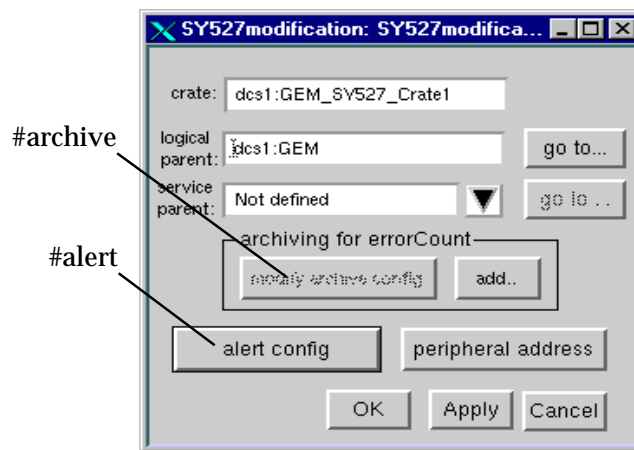


Figure 26: SY527modification panel.

4.1.20 SY527PeriphConfig

This panel (Figure 27) is called from the SY527modification panel (Figure 26). It is used to configure the peripheral address config of the SY527 crate when it is read by the IT-CO front-end software. The drvId is used in the case of DIM. OPC is not supported.

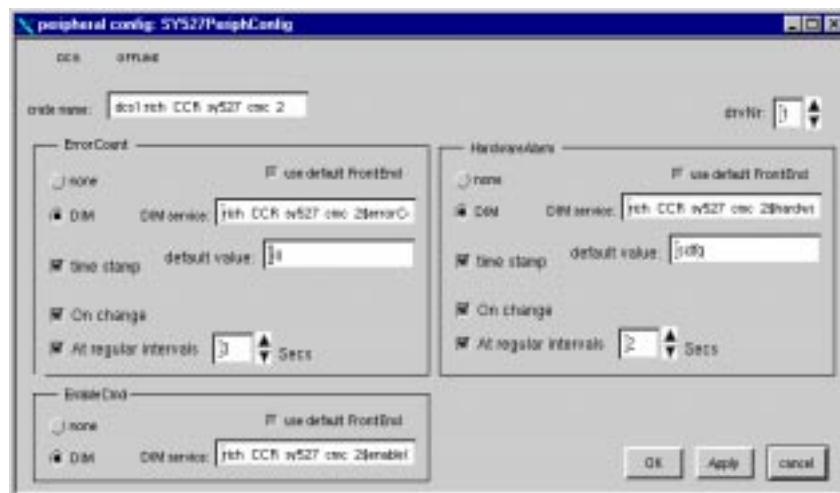


Figure 27: SY527PeriphConfig panel.

4.1.21 CAEN_HV_chmodification

This panel (Figure 28) is called from the COMPASSmodification panel. It is used to configure the CAEN_HV_ch peripheral address and alarm config of a CAEN_HV_ch element read by the IT-CO front-end software. An #archive config (Section 4.1.2, "archiveSet") can be added or modified for the all the readout element (actual PVSS structure): vMon, iMon, isOn, hardwareAlarm, rampingDirection and isOvc to all the settings: v0, v1, i0, i1, onOff, rUp, rDwn and tripTime to all the readBackSettings: v0, v1, i0, i1, rUp, rDwn and tripTime dpElement of the CAEN_HV_ch element. The pv_range of the settings.v0, settings.v1, settings.i0 and settings.i1 can be modified (Section 4.1.22, "pvRangeModification").

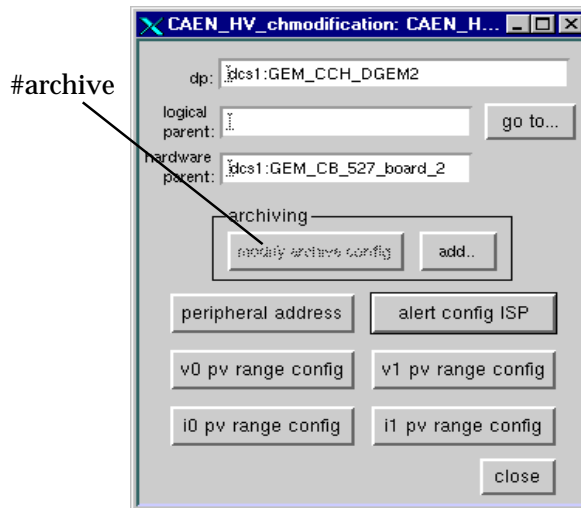


Figure 28: CAEN_HV_chmodification panel.

4.1.22 pvRangeModification

This panel (Figure 30) is called from modification and run-time panel, this is for changing the minimum and maximum of the allowed value.



Figure 29: pvRangeModification panel.

4.1.23 CAEN_HV_chPeriphConfig

This panel (Figure 30) is called from the CAEN_HV_chmodification panel (Figure 28). It is used to configure the peripheral address config of the CAEN_HV_ch element when it is read by the IT-CO front-end software. The drvId is used in the case of DIM.

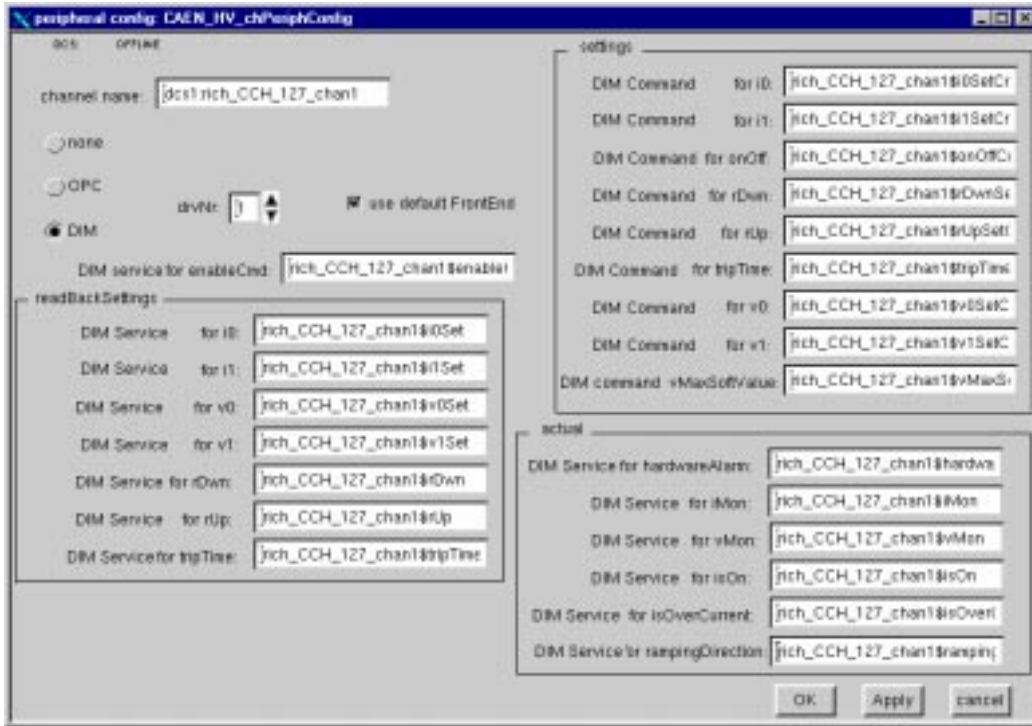


Figure 30: CAEN_HV_chPeriphConfig panel.

4.1.24 CAEN_HV_chAlertConfig

This panel (Figure 31) is called from the CAEN_HV_chmodification panel (Figure 28). It is used to configure the alarm config of the CAEN_HV_ch element when it is read by the IT-CO front-end software. An alarm can be set when the channel is tripping by using the hardwareAlarm dpElement.

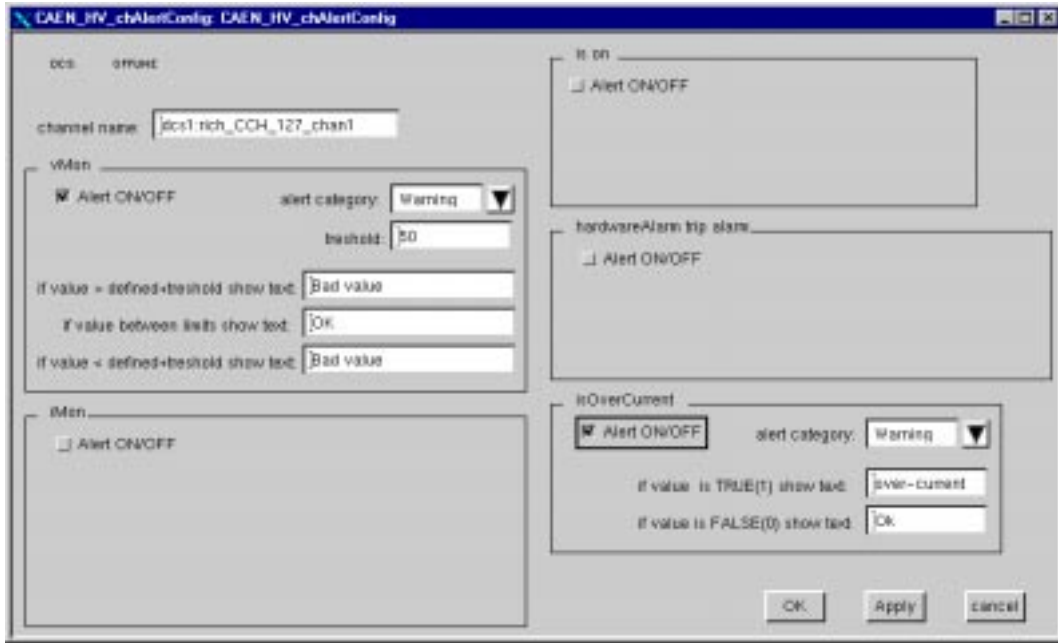


Figure 31: CAEN_HV_chAlertConfig panel.

4.1.25 CAEN_HV_ch_SPmodification

This panel (Figure 32) is called from the COMPASSmodification panel. It is used to configure the peripheral address and alarm config of a CAEN_HV_ch_SP element read by the IT-CO front-end software. The panel CAEN_HV_chAlertConfig (Figure 31) is called to configure the alarms of the in spill readout. An #archive config (Section 4.1.2, "archiveSet") can be added to the all the in spill and out of spill readout element (actual and actual_OSP PVSS structure): vMon, iMon, isOn, hardwareAlarm, rampingDirection and isOvc to all the settings: v0, v1, i0, i0, onOff, rUp, rDwn and tripTime to all the readBackSettings: v0, v1, i0, i1, rUp, rDwn and tripTime dpElement of the CAEN_HV_ch_SP element. The pv_range of the settings.v0, settings.v1, settings.i0 and settings.i1 can be modified (Section 4.1.22, "pvRangeModification").

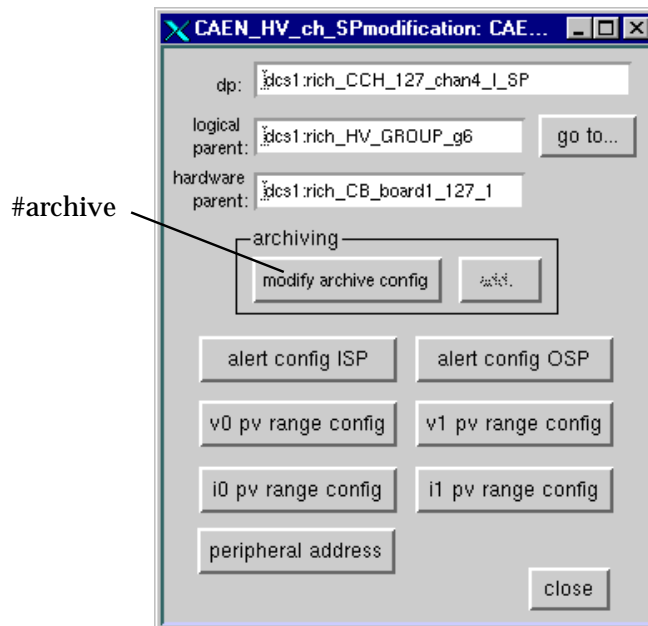


Figure 32: CAEN_HV_ch_SPmodification panel.

4.1.26 CAEN_HV_ch_SPPeriphConfig

This panel (Figure 33) is called from the CAEN_HV_ch_SPmodification panel (Figure 32). It is used to configure the peripheral address config of the CAEN_HV_ch_SP element when it is read by the IT-CO front-end software. The drvId is used in the case of DIM.

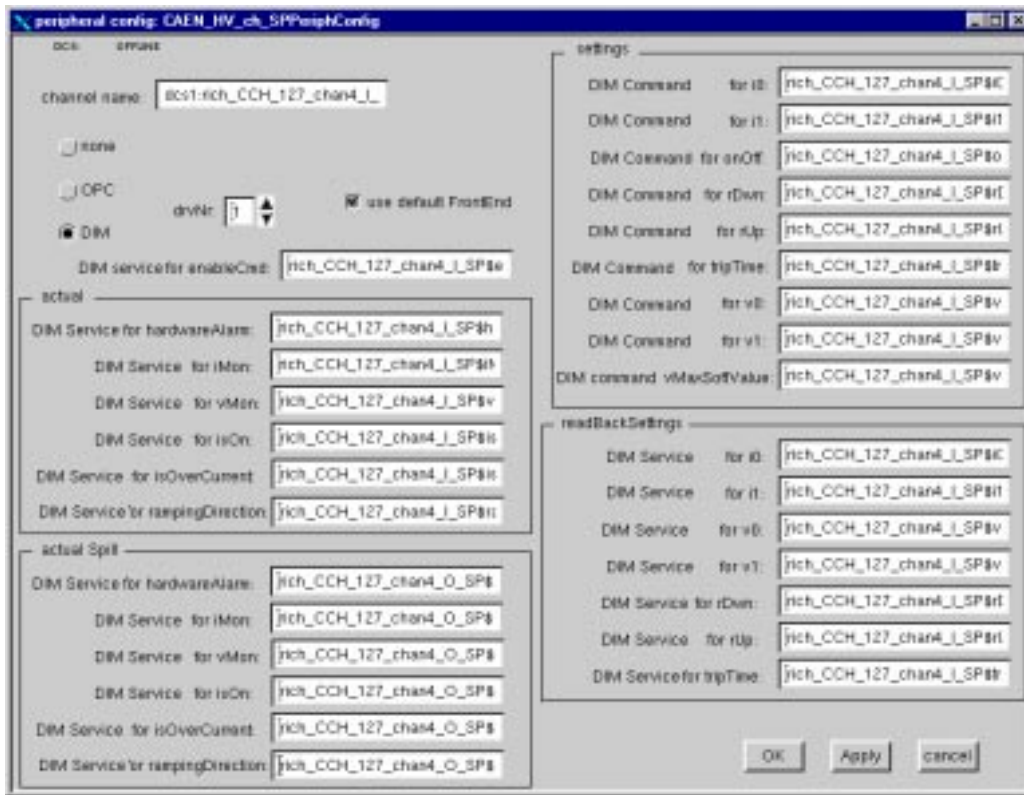


Figure 33: CAEN_HV_ch_SPPeriphConfig panel.

4.1.27 CAEN_HV_ch_SPAAlertConfig

This panel (Figure 34) is called from the CAEN_HV_ch_SPmodification panel (Figure 32). It is used to configure the alarm config of the out of spill readout data of the CAEN_HV_ch_SP element when it is read by the IT-CO front-end software. An alarm can be set when the channel is tripping by using the hardwareAlarm dpElement. The in spill readout alarm are configured by using the CAEN_HV_chAlertConfig panel (Figure 31) which is called from the CAEN_HV_ch_SPmodification panel (Figure 32).

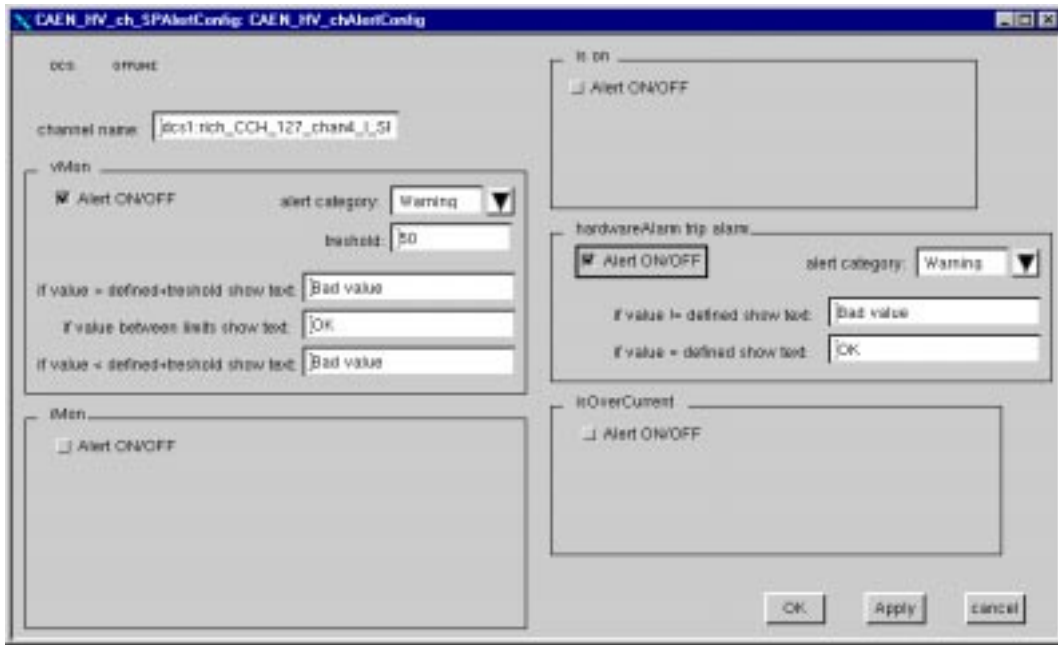


Figure 34: CAEN_HV_ch_SPAlertConfig panel.

4.1.28 HV_GROUPmodification

This panel (Figure 35) is called from the COMPASSmodification it is used to configure the HV_GROUP element

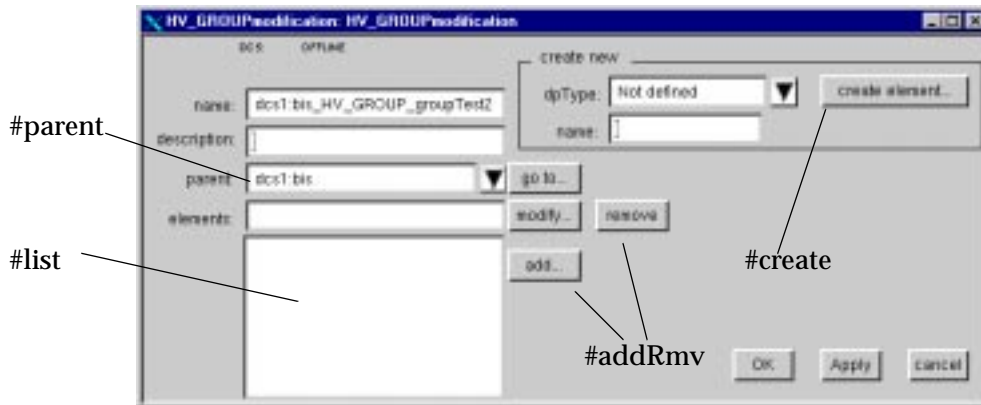


Figure 35: HV_GROUPmodification.

4.1.28.1#parent

The parent of the HV_GROUP element can be one of the following types: HV_SYSTEM, HV_GROUP, SUB_SYSTEM or DETECTOR. This represents the logical location of the element.

4.1.28.2#list

This is the list of child elements of the HV_GROUP, a double click on one line opens the corresponding modification panel.

4.1.28.3#create

New elements of type HV_GROUP can be created.

4.1.28.4#addRmv

The remove button removes the selected element from the list. The HV_GROUP element which is removed is added in the logTree list of the detector it belongs to. The add button calls the addElement panel (Figure 36) in order to add in the list element of types: HV_GROUP, CAEN_HV_ch and CAEN_HV_ch_SP.



Figure 36: addElement panel.

4.1.29 HV_SYSTEMmodification

This panel (Figure 37) is called from the COMPASSmodification, HV_SYSTEMmodification and DETECTORmodification panel. It is used to configure the HV_SYSTEM element.

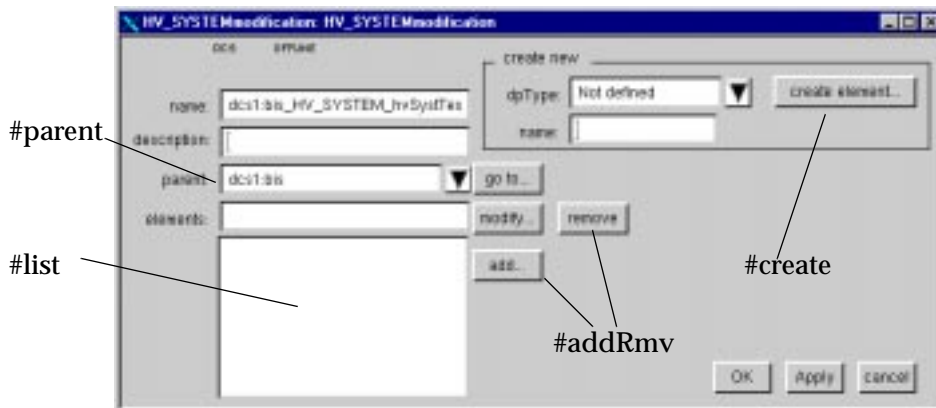


Figure 37: HV_SYSTEMmodification panel.

4.1.29.1#parent

The parent of the HV_SYSTEM element can be one of the following types: HV_SYSTEM, SUB_SYSTEM or DETECTOR. This represents the logical location of the element.

4.1.29.2#list

This is the list of child elements of the HV_SYSTEM, a double click on one line opens the corresponding modification panel.

4.1.29.3#create

New elements of type HV_GROUP or HV_SYSTEM can be created.

4.1.29.4#addRmv

The remove button removes the selected element from the list. The HV_SYSTEM, HV_GROUP element which is removed is added in the logTree list of the detector it belongs to. The add button calls the addElement panel (Figure 36) in order to add in the list element of types: HV_GROUP, HV_SYSTEM, CAEN_HV_ch and CAEN_HV_ch_SP.

4.1.30 SUB_SYSTEMmodification

This panel (Figure 38) is called from the COMPASSmodification, HV_SYSTEMmodification and DETECTORmodification panel. It is used to configure the HV_SYSTEM element.

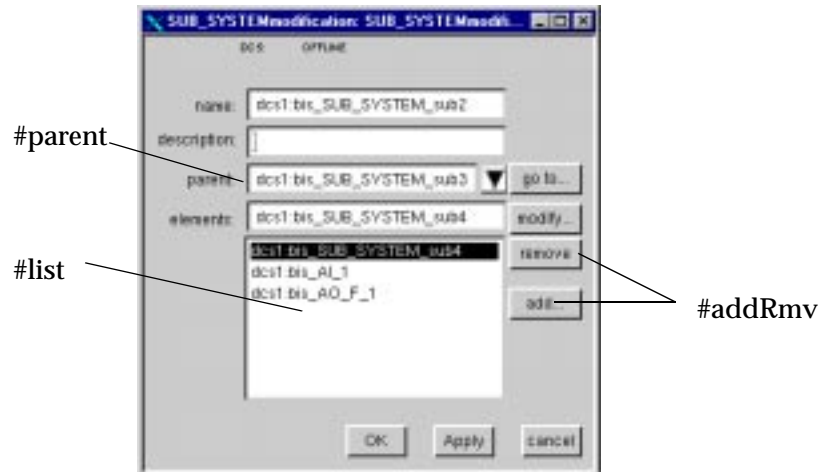


Figure 38: SUB_SYSTEMmodification.

4.1.30.1#parent

The parent of the SUB_SYSTEM element can be one of the following types: SUB_SYSTEM or DETECTOR. This represents the logical location of the element.

4.1.30.2#list

This is the list of child elements of the SUB_SYSTEM, a double click on one line opens the corresponding modification panel.

4.1.30.3#addRmv

The remove button removes the selected element from the list. The SUB_SYSTEM element which is removed is added in the logTree list of the detector it belongs to. The add button calls the addElement panel (Figure 36) in order to add in the list element of types: AI, AO, AO_F, DI, DO, DO_F, HV_GROUP, HV_SYSTEM and SUB_SYSTEM.

4.1.31 SERVICEmodification

This panel (Figure 38) is called from the COMPASSmodification panel. It is used to configure the SERVICE element.

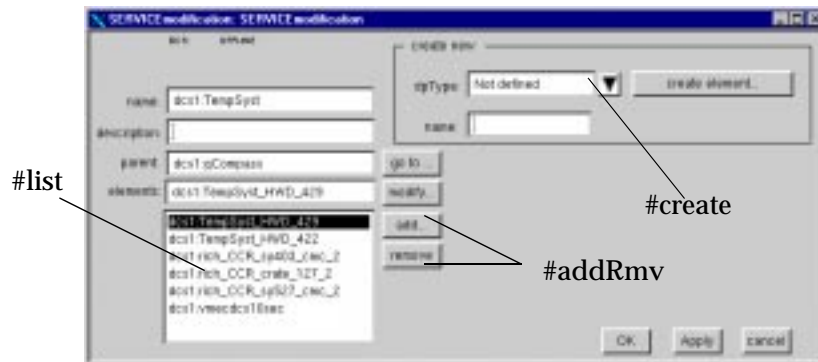


Figure 39: SERVICEmodification panel.

4.1.31.1#list

This is the list of child elements of the SERVICE, a double click on one line opens the corresponding modification panel. New elements can be added or removed.

4.1.31.2#create

New element of type HWD and PROCS can be created.

4.1.31.3#addRmv

The remove button removes the selected element from the list. The add button calls the addElement panel (Figure 36) in order to add in the list element of types: PROCS, SY127, SY403, SY527 and FAN_TRAY.

4.1.32 FAN_TRAYmodification

This panel (Figure 40) is called from the COMPASSmodification panel. It is used to configure the WIENER fan-tray peripheral address (#address) and alarm config of the FAN_TRAY element read by the IT-CO front-end software. An #archive config can be added or modified (Section 4.1.2, "archiveSet") to the errorCount, to all the readout: ftVoltage,

ftCurrent, temperatureAir, temperaturePS, temperatureBin, fanSpeed, ftOnOff, fanOk, overVoltageAlarm, underVoltageAlarm, overCurrentAlarm, underCurrentAlarm, all the readBackSettings: fanTripOn, errorTripOn, requestedVoltage, requestedVmax, requestedVmin, requestedImax, requestedImin, the settings: ftOnOff, fanSpeedRequired, resetVme, tripOn dpElement of the FAN_TRAY. The errorCount is the quantity of consecutive errors while reading data from the FAN_TRAY crate via the CAN bus. The #range is used to set the pv_range of the required speed. The #alert calls the FAN_TRAYAlertConfig panel (Section 4.1.34, "FAN_TRAYAlertConfig") to set an alarm on the errorCount. The #address calls the FAN_TRAYPeriphConfig (Section 4.1.33, "FAN_TRAYPeriphConfig").



Figure 40: FAN_TRAYmodification panel.

4.1.33 FAN_TRAYPeriphConfig

This panel (Figure 41) is called from the FAN_TRAYmodification panel (Figure 40). It is used to configure the peripheral address config of the FAN_TRAY element when it is read by the IT-CO front-end software. The drvId is used in the case of DIM.

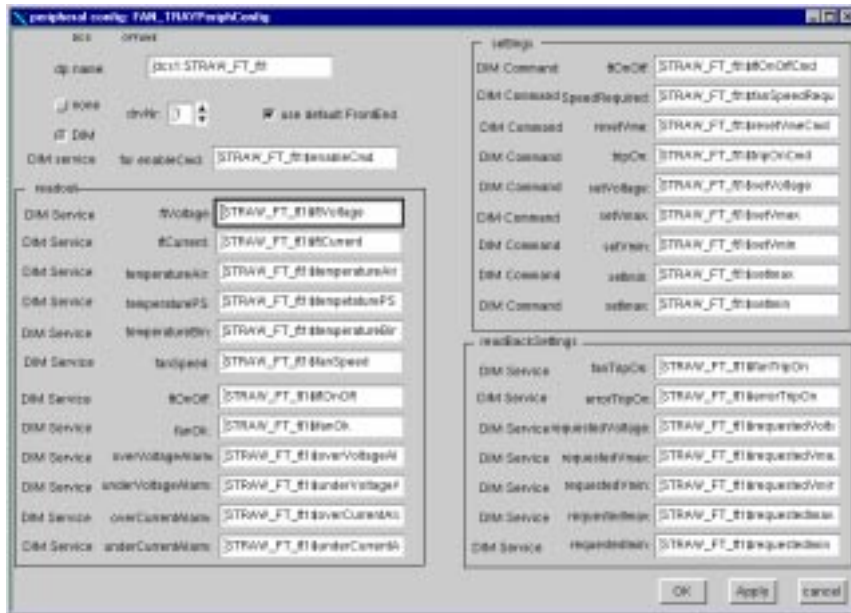


Figure 41: FAN_TRAYPeriphConfig panel.

4.1.34 FAN_TRAYAlertConfig

This panel (Figure 42) is called from the FAN_TRAYmodification panel (Figure 40). It is used to configure the alarm config of the FAN_TRAY element when it is read by the IT-CO front-end software. An alarm can be set on the errorCount and the readout element: ftOnOff, temperatureAir, temperatureBin, temperaturePS and fanOk.

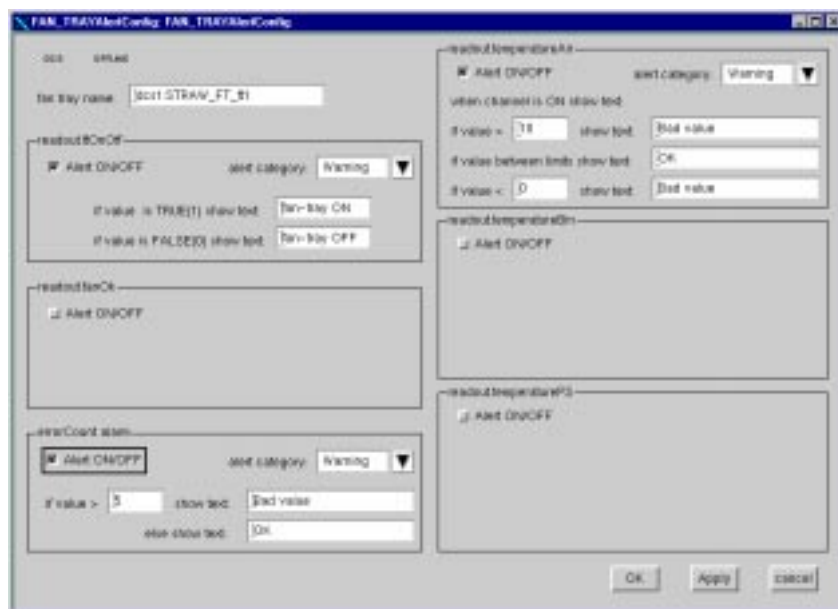


Figure 42: FAN_TRAYAlertConfig panel.

4.1.35 dimConfig



Figure 43: dimConfig panel.

The dimConfig panel (Figure 43) is used to create the configuration file for the PVSS00dim (interface PVSS-DIM). The configuration file can be edited. The PVSS00dim API manager should be re-started (Section 4.1.12, "COMPASSmodification"). When the create DIM config file is activated, the #statusBar is activated, the #close button is disabled and the work in process is displayed in the #list box.

4.1.36 okCancelButton

The okCancelButton panel (Figure 44) is pop-up when one wants to get a confirmation on an operation on an element of the framework. Pressing the Ok button will proceed with the operation, pressing Cancel will cancel the operation.

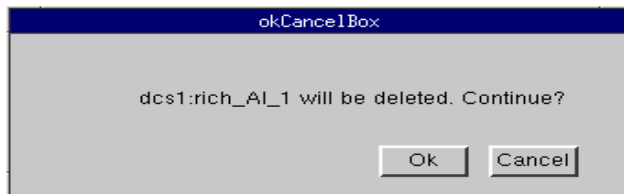


Figure 44: okCancelButton panel.

4.1.37 deleteBox

The deleteBox panel (Figure 45) is pop-up when one wants to delete an element from the framework. Pressing the delete button will continue the operation, pressing the Cancel button will cancel the operation.

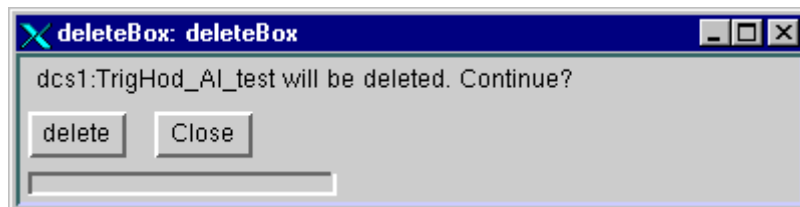


Figure 45: deleteBox panel.

4.1.38 errorDisplay

The errorDisplay panel (Figure 46) is pop-up when an error was detected while modifying the configuration of an element of the framework. The PVSS log file should also be edited to find the error⁴².

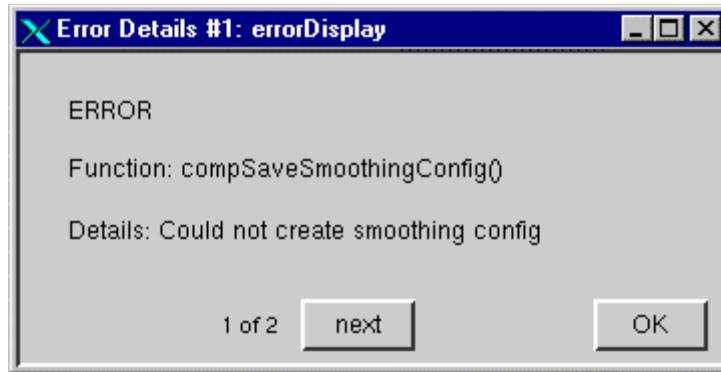


Figure 46: errorDisplay panel.

4.1.39 ImportHVConfig

The ImportHVConfig panel (Figure 47, Figure 48 and Figure 49) are used to import the IT-CO front-end software configuration of the CAEN-HV system already defined in the FileMaker Front-end database (refer to the appropriate documentation to create such a configuration).

4.1.39.1 ImportHVConfig: CAEN crate

This panel is used to import the CAEN HV crate already configured for the IT-CO front-end software.

⁴² One has to look at the stderr and stdout of the PVSS00ui program and at the PVSS_II.log file located in the folder /home/compass_offline/compass_dcs_off/log for the OFFLINE system and for the PRODUCTION system in the folder /home/compass_prod/compass_dcs_prod/log.

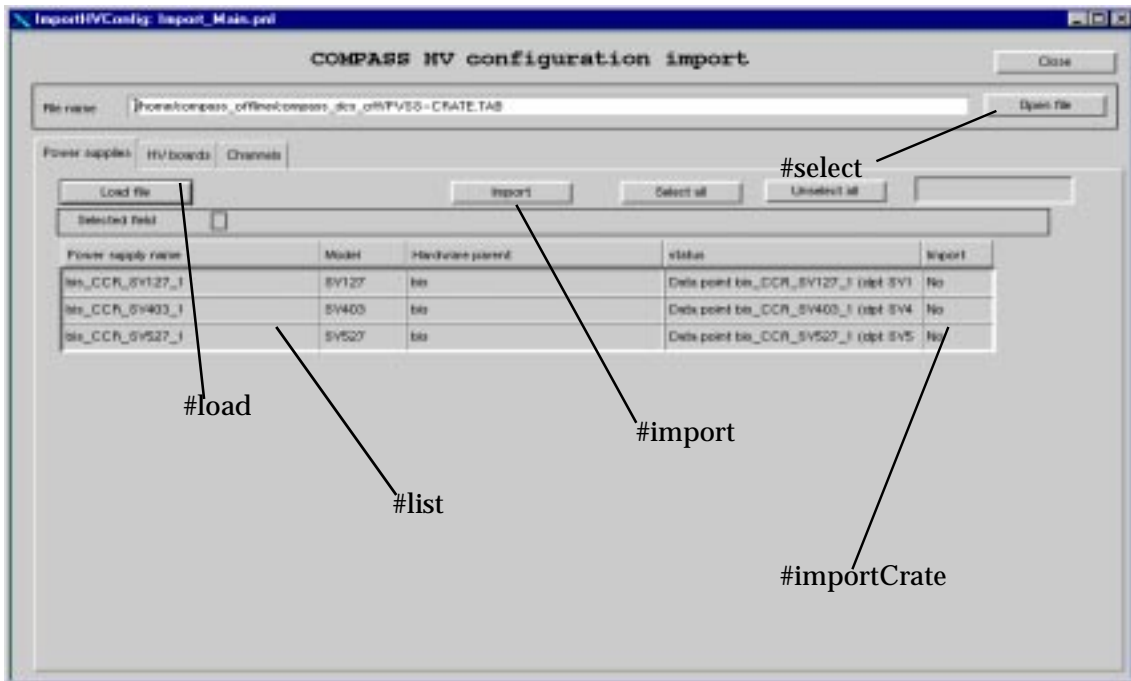


Figure 47: ImportHVConfig panel: CAEN crate.

4.1.39.1.1#select

Select the file containing the list of the crates to load.

4.1.39.1.2#load

Load the file selected in 'file name' text field

4.1.39.1.3#list

List of the crates to import in the PVSS system. The name of the crate, the type of the crate, the hardware parent (element: DETECTOR) and the status are listed

4.1.39.1.4#importCrate

Select which crate to import. By clicking in that box, one can select which crate to import. The value yes with a white background color in the Import column means that the crate will be imported when clicking on the import button. The value no with a grey background color in the Import column means that the crate will not be imported when clicking on the import button. The success is shown by a yes with a green background color, the failure of the import is also shown in this box by a no and an error text in the status column.

4.1.39.1.5#import

Import the selected crates.

4.1.39.2 ImportHVConfig: CAEN board

This panel is used to import the CAEN HV board already configured for the IT-CO front-end software.

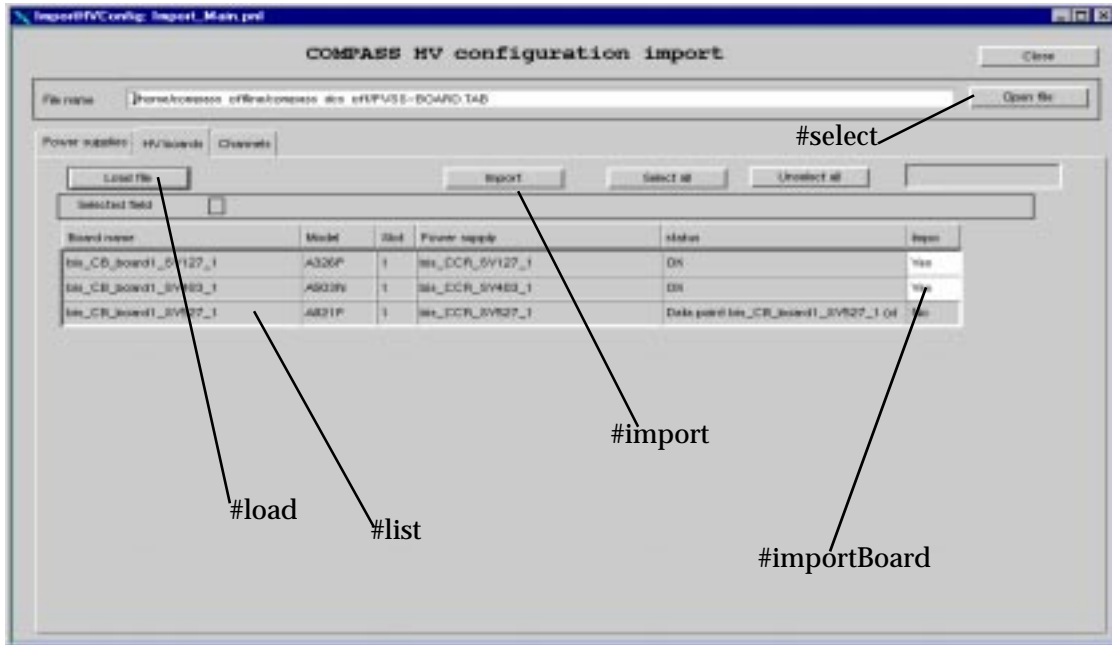


Figure 48: ImportHVConfig panel: CAEN board.

4.1.39.2.1 #select

Select the file containing the list of the boards to load.

4.1.39.2.2 #load

Load the file selected in 'file name' text field

4.1.39.2.3 #list

List of the boards to import in the PVSS system. The name of the board, the type of the board, the crate it belongs to (element: SY127, SY403 and SY527) and the status are listed

4.1.39.2.4 #importBoard

Select which board to import. By clicking in that box, one can select which board to import. The value yes with a white background color in the Import column means that the board will be imported when clicking on the import button. The value no with a grey background color in the Import column means that the board will not be imported when clicking on the import button. The success is shown by a yes with a green background color, the failure of the import is also shown in this box by a no and an error text in the status column.

4.1.39.2.5 #import

Import the selected boards.

4.1.39.3 ImportHVConfig: CAEN channel

This panel is used to import the CAEN HV channel already configured for the IT-CO front-end software.

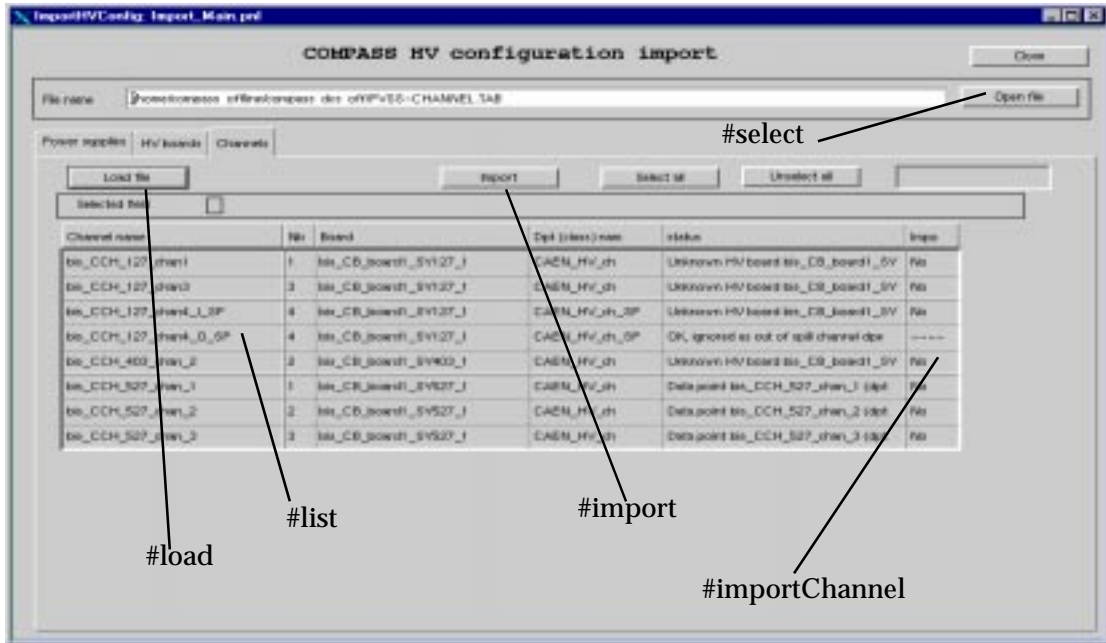


Figure 49: ImportHVConfig panel: CAEN channel.

4.1.39.3.1 #select

Select the file containing the list of the channels to load.

4.1.39.3.2 #load

Load the file selected in 'file name' text field

4.1.39.3.3 #list

List of the channels to import in the PVSS system. The name of the channel, the type of the channel, the board it belongs to (element: CAENHVBoard), the channel position in the board and the status are listed

4.1.39.3.4 #importChannel

Select which channel to import. By clicking in that box, one can select which channel to import. The value yes with a white background color in the Import column means that the channel will be imported when clicking on the import button. The value no with a grey background color in the Import column means that the channel will not be imported when clicking on the import button. The success is shown by a yes with a green background color, the failure⁴³ of the import is also shown in this box by a no and an error text in the status column.

4.1.39.3.5#import

Import the selected channels.

4.1.40 ImportFAN_TRAYConfig

The ImportFAN_TRAYConfig panel (Figure 50) is used to import the IT-CO front-end software configuration of the CAN WIENER FAN_TRAY element already defined in the FileMaker Front-end database (refer to the appropriate documentation to create such a configuration).

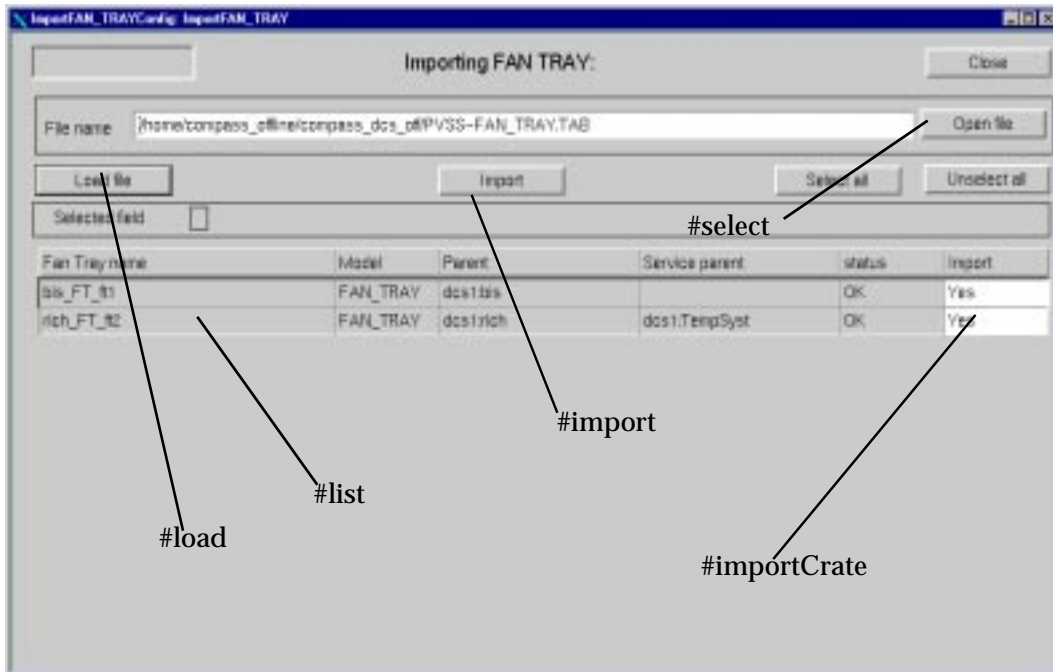


Figure 50: ImportFAN_TRAYConfig panel.

4.1.40.1#select

Select the file containing the list of the WIENER fan-tray to load.

4.1.40.2#load

Load the file selected in 'file name' text field.

4.1.40.3#list

List of the WIENER fan-tray to import in the PVSS system. The name of the WIENER fan-tray, the type of the WIENER fan-tray, the hardware parent (element: DETECTOR), the

⁴³ The channels of type CAEN_HV_ch_SP are shown twice in the list of elements: one with the extension _I_SP and one with the extension _O_SP. The channel with the extension _I_SP is imported in PVSS, and the channel with the extension _O_SP is not. This is because, two instances of HV channels are needed for in spill and out spill readout in the IT-CO front-end software and in PVSS only one is needed.

service parent (element: SERVICE, the computer from which it is read) and the status are listed.

4.1.40.4#importCrate

Select which WIENER fan-tray to import. By clicking in that box, one can select which fan-tray to import. The value yes with a white background color in the Import column means that the fan-tray will be imported when clicking on the import button. The value no with a grey background color in the Import column means that the fan-tray will not be imported when clicking on the import button. The success is shown by a yes with a green background color, the failure of the import is also shown in this box by a no and an error text in the status column.

4.1.40.5#import

Import the selected WIENER fan-tray.

4.1.41 HVConfig

The HVConfig panel (Figure 51) is used to create the CAEN HV configuration or to delete it. Refer to the JCOP framework documentation for more detail. It can be used to create, delete or modify the SY127, SY403 or SY527 elements. The #add button call the addNewPowerSupply panel (Figure 52). A detector has to be selected, this is the hardware location of the power supply. If this new crate has to be used by framework element the name should contain the detector name.

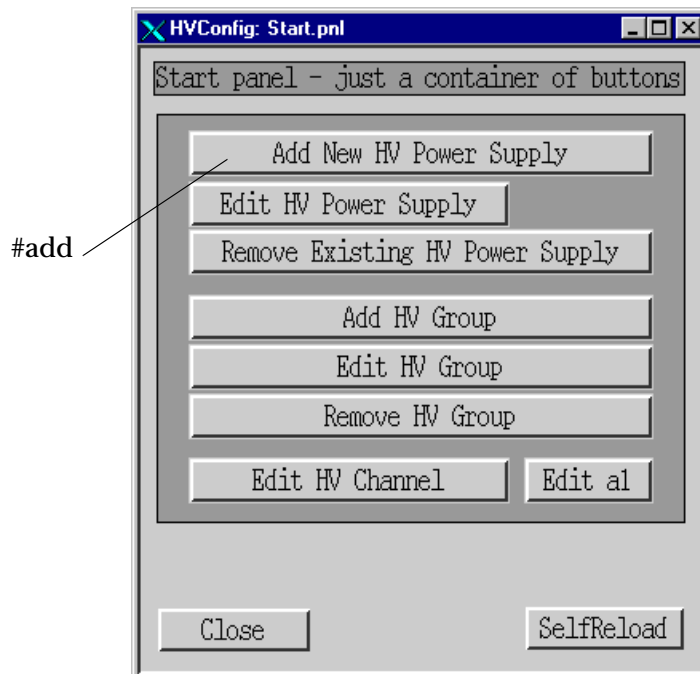


Figure 51: HVConfig panel.



Figure 52: addNewPowerSupply panel.

4.2 Run-time panels

The run-time panels is a set of views which can be used to browse the compass hierarchical tree, to view the data value of the element and to operate them. They call other run-time panels. These run-time panels can also be incorporated in user customized panels like reference panels, Section 4.3, "Reference panels", the appropriate \$-parameter has to be given to the panel. The user can see the value e.g.: voltage level of high-voltage channel but can not modify the configuration of the datapoint like adding or removing alarm, etc. The panel described in Section 4.2.12, "actionPanel" can be called from the run-time panel, it allows the action described in Section 3.3, "Alarms and states".

Only one panel per kind of element can be opened at the same time to avoid to have too many panels on the screen. There is no need to close the panel to open the run-time panel of an element of the same type, this is done by PVSS, the content of the panel is modified to show the new selected element.

Background color convention (Figure 53) for leaf element.

_ETM.	_ETM color is a color between blue and green (dark green) and it is used when the element is disabled.
Cyan.	Cyan (kind of light blue) color is used when the element is enabled and the data value is not valid: the received or sent data is not necessarily equivalent to real data, the quality of the data is uncertain.
Gray.	Gray color is used when the element is enabled and valid, an alarm has been defined for the element and the status of this alarm is masked.
Green.	Green color is used when the element is enabled and valid, an alarm has been defined for the element and the status of alarm is active with no alarm at moment.
White.	White color is used when the element is enabled and valid, and no alarm has been defined on for the element.
_dpdoesnotexist.	Purple color is used if the data point element is not existing.

- blinking Yellow (Yellow/White).** This color is used when the element is enabled and valid, an alarm has been defined for the element, a WARNING alarm occurred and is not acknowledged.
- Yellow.** This color is used when the element is enabled and valid, an alarm has been defined for the element, a WARNING alarm occurred and is acknowledged.
- blinking Orange (Orange/White).** This color is used when the element is enabled and valid, an alarm has been defined for the element, a FAULT alarm occurred and is not acknowledged.
- Orange.** This color is used when the element is enabled and valid, an alarm has been defined for the element, a FAULT alarm occurred and is acknowledged.
- blinking Red (Red/White).** This color is used when the element is enabled and valid, an alarm has been defined for the element, a FATAL alarm occurred and is not acknowledged.
- Red.** This color is used when the element is enabled and valid, an alarm has been defined for the element, a FATAL alarm occurred and is acknowledged.



Figure 53: Background color convention for leaf element.

Background color convention (Figure 53) for node element.

- _ETM.** _ETM color is a color between blue and green (dark green) and it is used when the element is disabled.
- Gray.** Gray color is used when the element is enabled and valid and its state is OK and it is masked.
- Green.** Green color is used when the element is enabled and valid and its state is OK.
- _dpdoesnotexist.** Purple color is used if the data point element is not existing.
- Yellow.** This color is used when the element is enabled and valid and its state is WARNING, this means that a child leaf is in Yellow or

blinking Yellow (Section , "Background color convention (Figure 53) for leaf element.") or a child node is in Yellow.

Orange.

This color is used when the element is enabled and valid and its state is FAULT, this means that a child leaf is in Orange or blinking Orange (Section , "Background color convention (Figure 53) for leaf element.") or a child node is in Orange.

Red.

This color is used when the element is enabled and valid and its state is FATAL, this means that a child leaf is in Red or blinking Red (Section , "Background color convention (Figure 53) for leaf element.") or a child node is in Red.



Figure 54: Background color convention for node element.

4.2.1 browseCompassTree

The browseCompassTree panel (Figure 55) is a utility which allows an operator or an expert to browse the COMPASS hierarchical tree. The gCompass element is the top of the tree. The "masked element list" button opens the panel MaskedList (Section 4.3.48, "MaskedList"), the "disabled element list" button opens the panel DisableList (Section 4.3.49, "DisableList") and the "full element list" button opens the panel ElementList (Section 4.3.50, "ElementList").

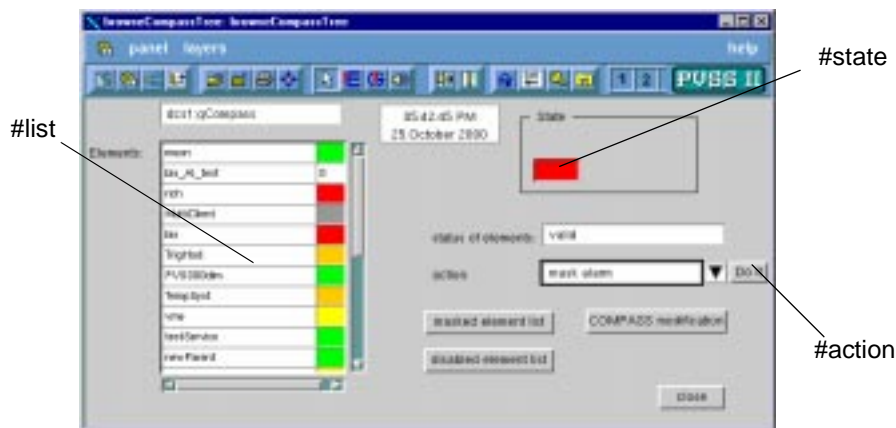


Figure 55: browseCompassTree panel.

4.2.1.1 #list

List of child element of gCompass, calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.1.2 #state

state of gCompass. The background color follows the rule: Background color convention (Figure 53) for node element..

4.2.1.3 #action

Action to be initiated at the gCompass level (Section 3.4, "action on the framework elements").

4.2.2 AI

The \$-parameter of this panel is: \$AI which is the data point name.

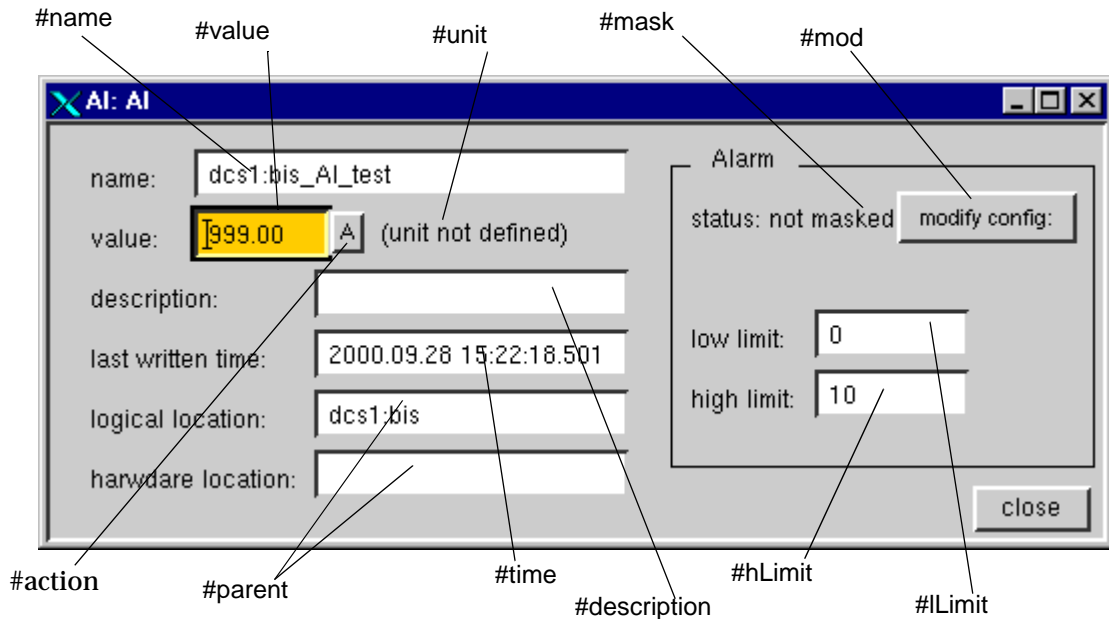


Figure 56: AI panel.

4.2.2.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 56 the background color is orange: the element is in a fault state. If the 'value' dpElement of the AI element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.2.2.2 #mask

Shows the alarm status of 'value' dpElement which can be either "masked" or "not masked". Script for showing the alarm status and alarm limits are implemented behind the working function (dpConnect). Texts and text fields inside alarm frames are disabled if alarm is not defined for the dpElement 'value' (like in Figure 56).

4.2.2.3 #parent

Shows the logical location (parent: HWD, DETECTOR, COMPASS or SUB_SYSTEM) and the hardware location (HWD) of the AI element. A double click on the parent opens the corresponding run-time panel.

4.2.2.4 #description

description of the AI element.

4.2.2.5 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.2.6 #name

Displays the name of datapoint (content of \$AI).

4.2.2.7 #Limit

Text field which shows the lower alarm limit of the 'value' dpElement.

4.2.2.8 #hLimit

Text field which shows the higher alarm limit of the 'value' dpElement.

4.2.2.9 #time

Shows the latest time when PVSS received the new value of 'value' dpElement implemented by a working function (dpConnect).

4.2.2.10#mod

Opens the alarmModification panel (Section 4.2.3, "alarmModification").

4.2.2.11#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$AI is given to this panel, e.g. "\$dp:dcs1:rich_AI_1".

4.2.3 alarmModification

The alarm limit and category of the element can be changed⁴⁴.

⁴⁴ This panel modifies the existing alarm limits and category, the alarms cannot be deleted and cannot be added.

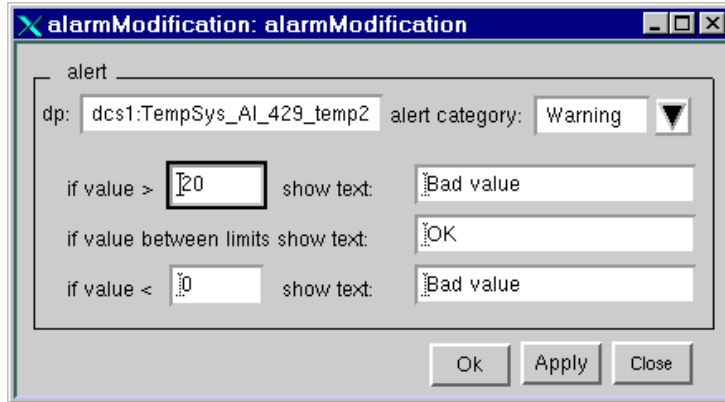


Figure 57: alarmModification panel.

4.2.4 AO

The \$-parameter of this panel is: \$AO which is the data point name.

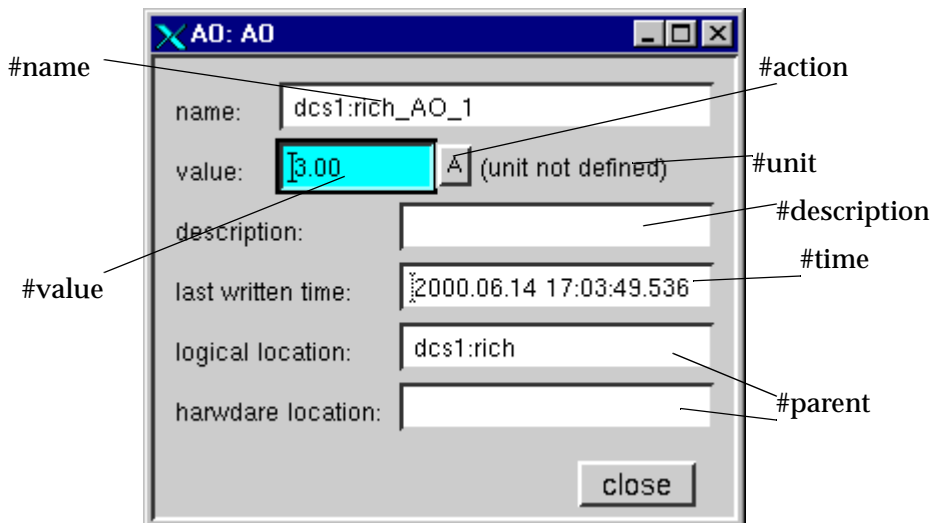


Figure 58: AO panel.

4.2.4.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.2.4.2 #parent

Shows the logical location (parent: HWD, DETECTOR, COMPASS or SUB_SYSTEM) and the hardware location (HWD) of the AO element. A double click on the parent opens the corresponding run-time panel.

4.2.4.3 #description

description of the AO element.

4.2.4.4 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.4.5 #name

Displays the name of the datapoint (content of \$AO).

4.2.4.6 #time

Shows the latest time when PVSS received the new value of 'value' dpElement implemented by a working function (dpConnect).

4.2.4.7 #action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$AI is given to this panel, e.g. "\$dp:dcs1:rich_AO_1".

4.2.5 AO_F

The \$-parameter of this panel is: \$AO_F which is the data point name.

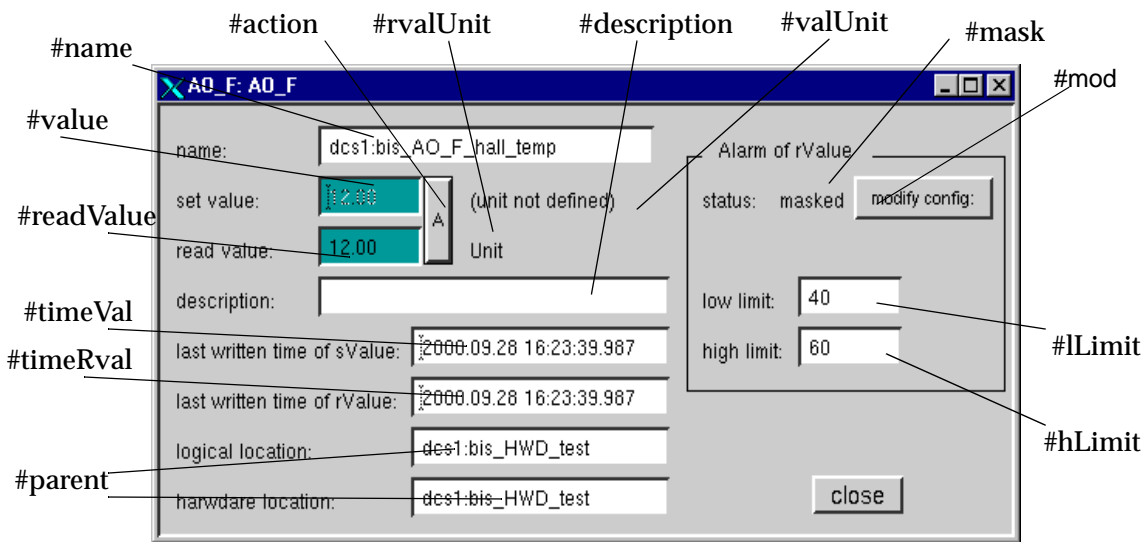


Figure 59: AO_F panel.

4.2.5.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 59 the background color is _ETM: the element is disabled. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.2.5.2 #parent

Shows the logical location (parent: HWD, DETECTOR, COMPASS or SUB_SYSTEM) and the hardware location (HWD) of the AO_F element. A double click on the parent opens the corresponding run-time panel.

4.2.5.3 #description

description of the AO_F element.

4.2.5.4 #valUnit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.5.5 #name

Displays the name of datapoint (content of \$AO_F).

4.2.5.6 #timeVal

Shows the latest time when PVSS received the new value of 'value' dpElement implemented by a working function (dpConnect).

4.2.5.7 #readValue

Shows the value of 'readValue' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 59 the background color is _ETM: the element is disabled. If the 'readValue' dpElement of the AO_F element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'readValue' dpElement by typing new value to text field and pressing 'enter'.

4.2.5.8 #lLimit

Text field which shows the lower alarm limit of the 'readValue' dpElement.

4.2.5.9 #hLimit

Text field which shows the higher alarm limit of the 'readValue' dpElement.

4.2.5.10#timeRval

Shows the latest time when PVSS received the new value of 'readValue' dpElement implemented by a working function (dpConnect).

4.2.5.11#rvalUnit

Shows the unit of 'readValue' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.5.12#mask

Shows the alarm status of 'readValue' dpElement which can be either "masked" or "not masked". Script for showing the alarm status and alarm limits are implemented behind the working function (dpConnect). Texts and text fields inside alarm frames are disabled if alarm is not defined for the dpElement 'readValue' (like in Figure 59).

4.2.5.13#mod

Opens the alarmModification panel (Section 4.2.3, "alarmModification").

4.2.5.14#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$AO_F is given to this panel, e.g. "\$dp: dcs1:rich_AO_F_asd".

4.2.6 DI

The \$-parameter of this panel is: \$DI which is the data point name.

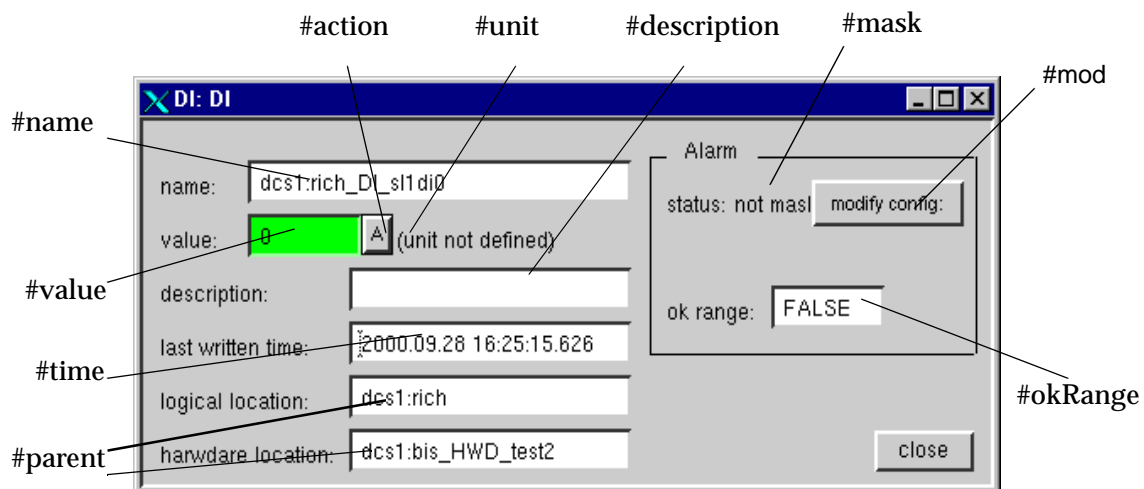


Figure 60: DI panel.

4.2.6.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 60 the background color is green: there is no alarm on the value. If the 'value' dpElement of the DI element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.2.6.2 #parent

Shows the logical location (parent: HWD, DETECTOR, COMPASS or SUB_SYSTEM) and the hardware location of the DI element. A double click on the parent opens the corresponding run-time panel.

4.2.6.3 #description

description of the DI element.

4.2.6.4 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.6.5 #name

Displays the name of datapoint (content of sent \$DI).

4.2.6.6 #okRange

Shows the okRange for defined alarm of the 'value' dpElement, e.g. in Figure 60 alarm will be generated when the value will be equal to TRUE (1).

4.2.6.7 #time

Shows the latest time when PVSS received the new value of 'value' dpElement implemented by a working function (dpConnect).

4.2.6.8 #mask

Shows the alarm status of 'value' dpElement which can be either "masked" or "not masked". Script for showing the alarm status and alarm limits are implemented behind the working function (dpConnect). Texts and text fields inside alarm frames are disabled if alarm is not defined for the dpElement 'value'.

4.2.6.9 #mod

Opens the digAlarmModification panel (Section 4.2.7, "digAlarmModification").

4.2.6.10#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DI is given to this panel, e.g. "\$dp:dcs1:straw_DI_testDi".

4.2.7 digAlarmModification

The alarm state and category of the element can be changed⁴⁵.

⁴⁵ This panel modifies the existing alarm limits and category, the alarms cannot be deleted and cannot be added.

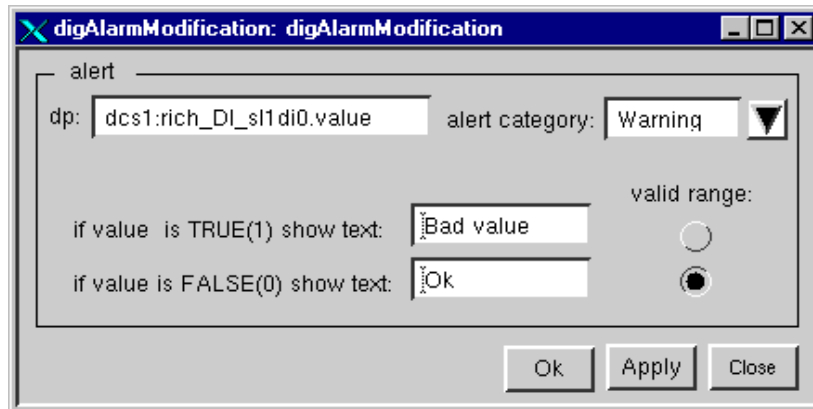


Figure 61: digAlarmModification panel.

4.2.8 DO

The \$-parameter of this panel is: \$DO which is the data point name.

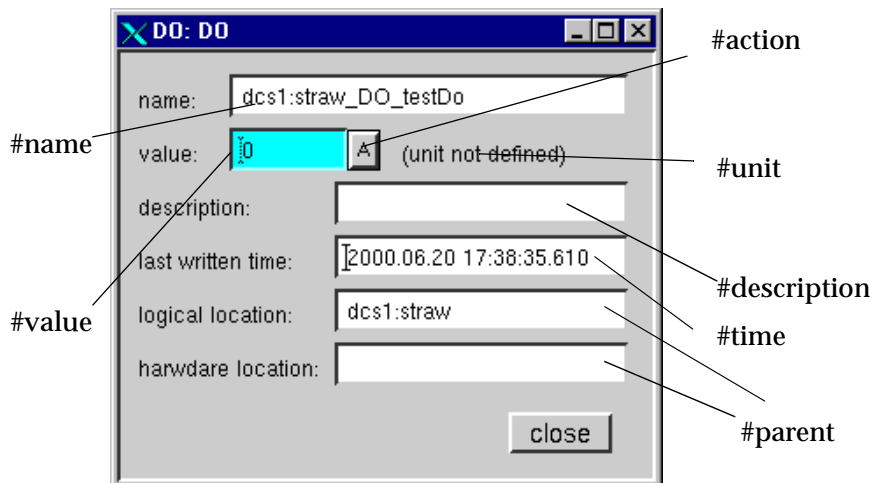


Figure 62: DO panel.

4.2.8.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'. E.g. in Figure 62 the background color is cyan: the element is invalid.

4.2.8.2 #parent

Shows the logical location (parent: HWD, DETECTOR, COMPASS or SUB_SYSTEM) and the hardware location (HWD) of the DO element. A double click on the parent opens the corresponding run-time panel.

4.2.8.3 #description

description of the DO element.

4.2.8.4 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.8.5 #name

Displays the name of datapoint (content of \$DO).

4.2.8.6 #time

Shows the latest time when PVSS received the new value of 'value' dpElement implemented by a working function (dpConnect).

4.2.8.7 #action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DO is given to this panel, e.g. "\$dp:dcs1:straw_DO_testDo".

4.2.9 DO_F

The \$-parameter of this panel is: \$DO_F which is the data point name.

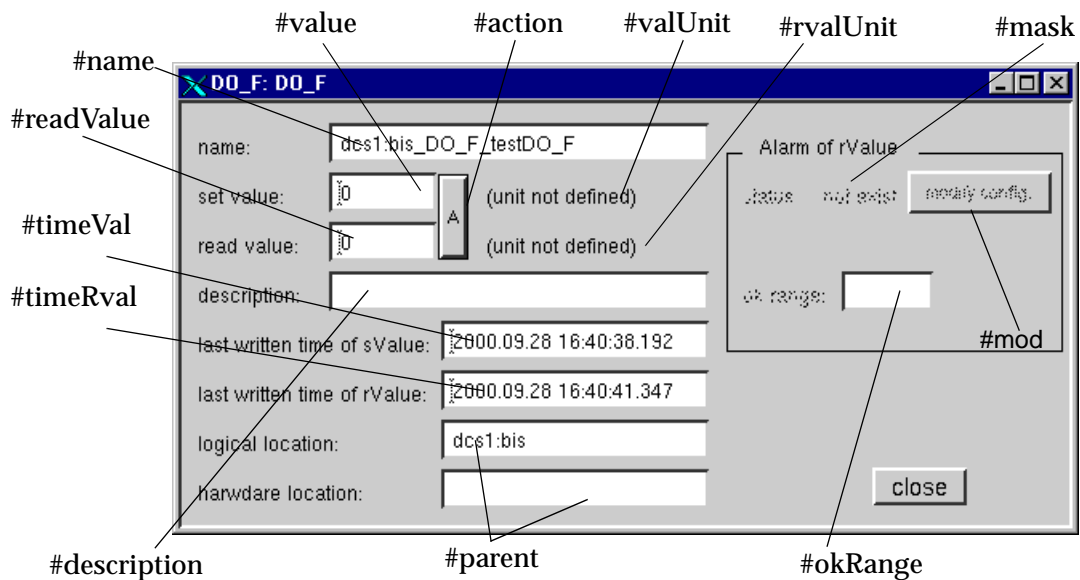


Figure 63: DO_F panel.

4.2.9.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 63 the

background color is white: there is no alarm defined on the dpElement. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.2.9.2 #parent

Shows the logical location (parent: HWD, DETECTOR, COMPASS or SUB_SYSTEM) and the hardware location (HWD) of the DO_F element. A double click on the parent opens the corresponding run-time panel.

4.2.9.3 #description

description of the DO_F element.

4.2.9.4 #valUnit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.9.5 #name

Displays the name of datapoint (content of \$DO_F).

4.2.9.6 #timeVal

Shows the latest time when PVSS received the new value of 'value' dpElement implemented by a working function (dpConnect).

4.2.9.7 #readValue

Shows the value of 'readValue' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 63 the background color is white: the element is enabled and there is no alarm defined on the dpElement. If the 'readValue' dpElement of the DO_F element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'readValue' dpElement by typing new value to text field and pressing 'enter'.

4.2.9.8 #rvalUnit

Shows the unit of 'readValue' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.2.9.9 #okRange

Shows the okRange for defined alarm of the 'readValue' dpElement, e.g. in Figure 63 alarm will be generated when the value will be equal to TRUE (1).

4.2.9.10#mask

Shows the alarm status of 'readValue' dpElement which can be either "masked" or "not masked". Script for showing the alarm status and alarm limits are implemented behind the working function (dpConnect). Texts and text fields inside alarm frames are disabled if alarm is not defined for the dpElement 'readValue'.

4.2.9.11#mod

Opens the alarmModification panel (Section 4.2.3, "alarmModification").

4.2.9.12#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DO_F is given to this panel, e.g. "\$dp:dcs1:rich_DO_F_testDoF".

4.2.9.13#timeRval

Shows the latest time when PVSS received the new value of 'readValue' dpElement implemented by a working function (dpConnect).

4.2.10 HWD

The \$-parameter of this panel is: \$HWD which is the data point name.

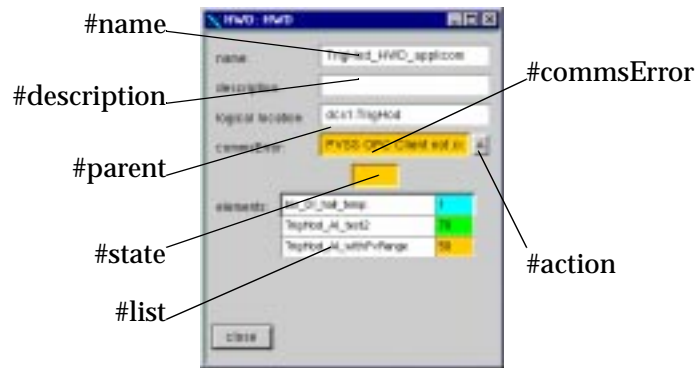


Figure 64: HWD panel.

4.2.10.1#list

Shows the list of child element of the HWD: elements (AI, AO, AO_F, DI, DO, DO_F or HWD) which parent or hwdOrig is this particular HWD, calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.10.2#name

Displays the name of datapoint (content of \$HWD).

4.2.10.3#description

description of the HWD element.

4.2.10.4#parent

Shows the location (parent: DETECTOR or another HWD) of the HWD element. A double click on the parent opens the corresponding run-time panel.

4.2.10.5#state

Shows the state of the HWD element. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.10.6#commsError

Shows the different communication error when a HWD is connected to an OPC Server (Section 3.7, "OPC"): 0=OK, 1=OPC server not connected, 2=PVSS OPC Client not connected. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.10.7#action

Opens the actionPanel: Section 4.2.12, "actionPanel". SHWD is given to this panel, e.g. "\$dp: dcs1:rich_HWD_testHWD".

4.2.11 DETECTOR

The \$-parameter of this panel is: \$DETECTOR which is the data point name.

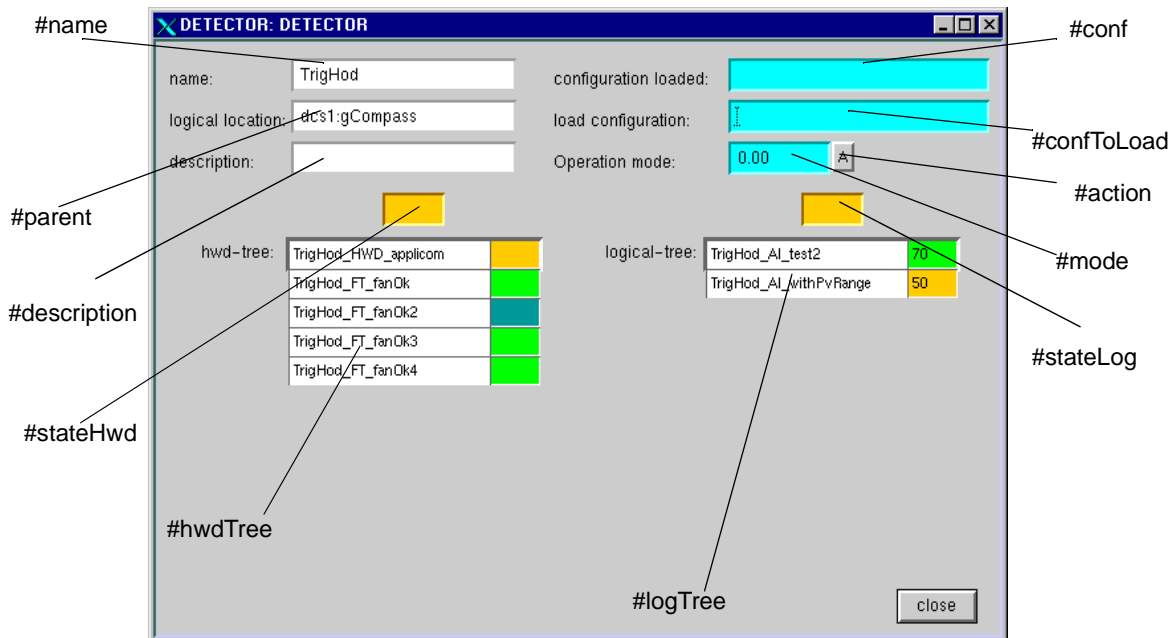


Figure 65: DETECTOR panel.

4.2.11.1#hwdTree

display the list of the elements of the hardware tree of this detector: elements of type HWD, SY127, SY403 and SY527 which parent is this particular detector, calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.11.2#name

Displays the name of datapoint (content of \$DETECTOR).

4.2.11.3#description

description of the DETECTOR element.

4.2.11.4#parent

Display the logical location of the detector. The logical location for all detectors is pre-defined to gCompass. A double click on the parent opens the browseCompassTree panel (Section 4.2.1, "browseCompassTree").

4.2.11.5#mode⁴⁶

Shows the current operational mode of the detector which can be either RUN, MAINTENANCE or SHUTDOWN. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 65 status of 'mode' is invalid.

4.2.11.6#conf⁴⁷

Shows the loaded configuration file which is loaded. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 65 status of 'configurationFile' is invalid.

4.2.11.7#confToLoad⁴⁸

Shows the configuration file to be loaded. The user can change it by typing new configuration file into the text field and pressing 'enter'. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 65 status of 'configurationFile' is invalid.

4.2.11.8#logTree

display the list of the elements of the logical tree of this detector: elements of type AI, AO, AO_F, DI, DO, DO_F, HV_GROUP, HV_SYSTEM, CAEN_HV_ch, CAEN_HV_ch_SP or SUB_SYSTEM which parent is this particular detector, calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.11.9#stateHwd

Shows the state of the 'hwdTree'. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

⁴⁶ not implemented in the release v1.4, the mode can be changed but it has no effect on the detector element itself.

⁴⁷ not implemented in the release v1.4

⁴⁸ not implemented in the release v1.4

4.2.11.10#stateLog

Shows the state of the 'logicalTree'. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.11.11#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DETECTOR is given to this panel, e.g. "\$dp:dcs1:rich".

4.2.12 actionPanel

All defined actions defined in Section 3.4.1, "General purpose actions" can be triggered from that panel. Most of the actions are implemented in the way that pushing the button will only trigger the control script where the actual operation is done. The buttons are disabled when the action is not allowed, the allowed action can be triggered many time. This panel (Section 3.10, "interfacing new element type with the framework") can also be used for element type which are not part of the framework element list (Table 1).

The \$-parameter of this panel is: \$dp which is the data point name

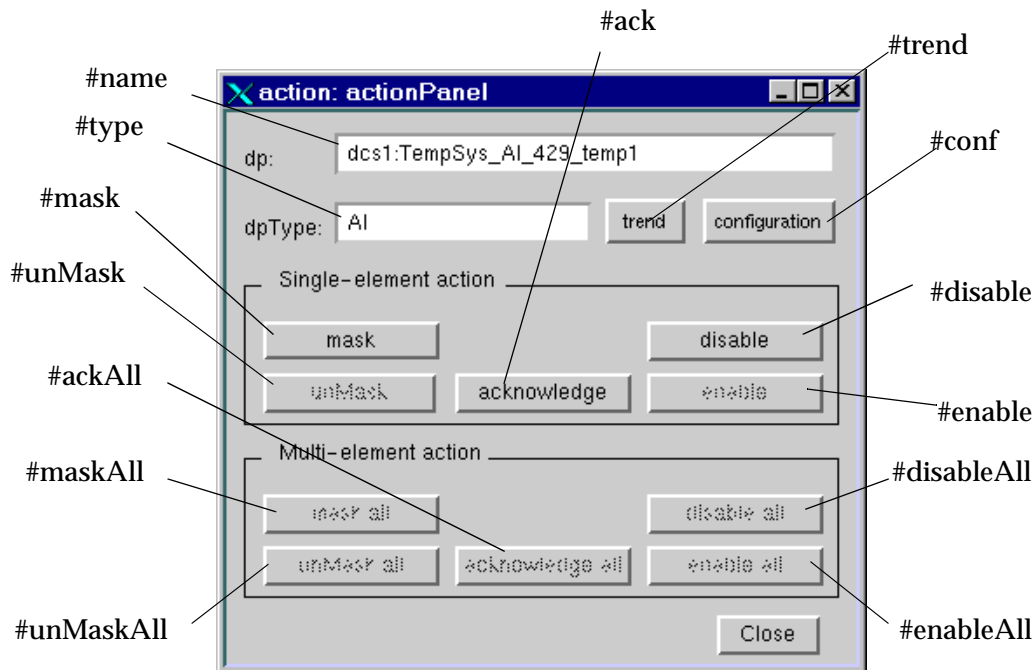


Figure 66: Action panel.

4.2.12.1#unMask

Un-mask the element.

4.2.12.2#mask

Mask the element.

4.2.12.3#maskAll

Mask the element and all its child elements.

4.2.12.4#ackAll

Acknowledge the alarm of the element and the alarms of all the child elements.

4.2.12.5#ack

Acknowledge the alarm of the element.

4.2.12.6#enable

Enable the element.

4.2.12.7#disable

Disable the element.

4.2.12.8#enableAll

Enable the element and all its child elements.

4.2.12.9#disableAll

Disable the element and all its child elements.

4.2.12.10#unMaskAll

Un-mask the element and all its child elements.

4.2.12.11#type

Displays the type of datapoint.

4.2.12.12#name

Displays the name of datapoint (content of \$dp).

4.2.12.13#trend

Opens the variableScaleTrend panel (Section 4.2.14, "variableScaleTrend") for AI type element, the digitalVariableScaleTrend panel (Section 4.2.17, "digitalVariableScaleTrend") for DI type element, the HVTrendPanel panel (Section 4.2.15, "HVTrendPanel") for CAEN_HV_ch type element and the FTTrendPanel (Section 4.2.16, "FTTrendPanel") for FAN_TRAY type element.

4.2.12.14#conf

Opens the modification panel of the element.

4.2.13 HV_ActionPanel

All defined actions defined in Section 3.4.2, "HV actions" can be triggered from that panel (Figure 67 and Figure 68). The "HV download" button calls the HVTable panel ().



Figure 67: HV_ActionPanel when called from a HV_SYSTEM, SY127, SY403 or SY527.



Figure 68: HV_ActionPanel when called from a HV_GROUP.

4.2.14 variableScaleTrend

This panel (Figure 69) trends the dpElement, its pv_range (for the readout.vMon dpElement of the CAEN_HV_ch and CAEN_HV_ch_SP, the pv_range is the settings.v0, and for the readout.iMon dpElement of the CAEN_HV_ch and CAEN_HV_ch_SP, the pv_range is the settings.i0) and its alarm limits if any. The number of days (#days) can be modified, the upper (#up) and lower (#low) limit of the trend (#trend) can be adjusted. The alarm limits are represented if they exist. The #stop/start button stops or starts the trend. If the trend is stopped the scroll bar can be used to see the data in the past. The trend can be printed on a printer (#print). A printer name has to be entered (#printer). The _dpdoesnotexist color (Background color convention (Figure 53) for leaf element.) means that the data is not archived. On the contrary of all the run-time panel, it is possible to have many variableScaleTrend opened. It is also called by the actionPanel (Section 4.2.12, "actionPanel") for AI type element.

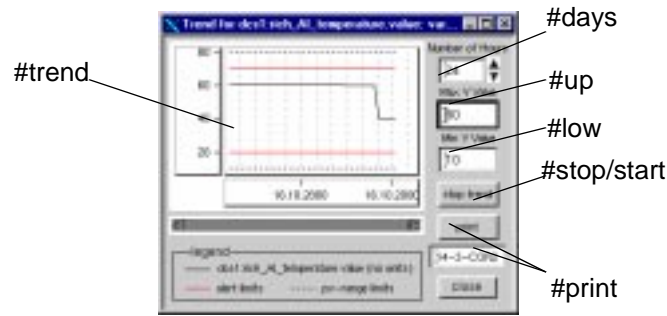


Figure 69: variableScaleTrend panel.

4.2.15 HVTrendPanel

This panel (Figure 70) trends the readout.vMon, readout.iMon and readout.isOn of a CAEN_HV_ch element, it is called by the actionPanel (Section 4.2.12, "actionPanel"). It calls the variableScaleTrend panel and the digitalVariableScaleTrend panel (Section 4.2.14, "variableScaleTrend", Section 4.2.17, "digitalVariableScaleTrend") for readout.vMon, readout.iMon and readout.isOn.

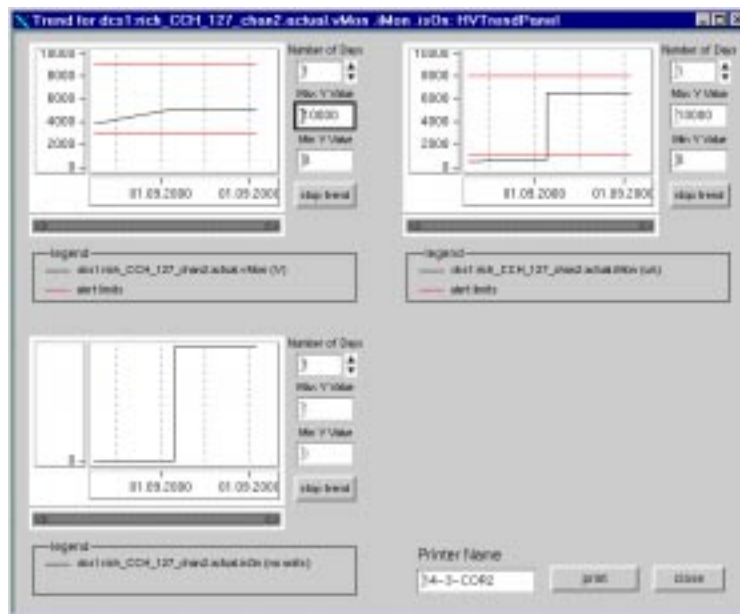


Figure 70: HVTrendPanel panel.

4.2.16 FTTrendPanel

This panel (Figure 71) trends the readout.temperatureAir, readout.temperatureBin, readout.temperaturePS and its respective alarms limits if they exist and the readout.ftOnOff of a FAN_TRAY element, it is called by the actionPanel (Section 4.2.12, "actionPanel"). The readout.temperatureAir, readout.temperatureBin and readout.temperaturePS trend are not shown if the value is less than -900⁴⁹.

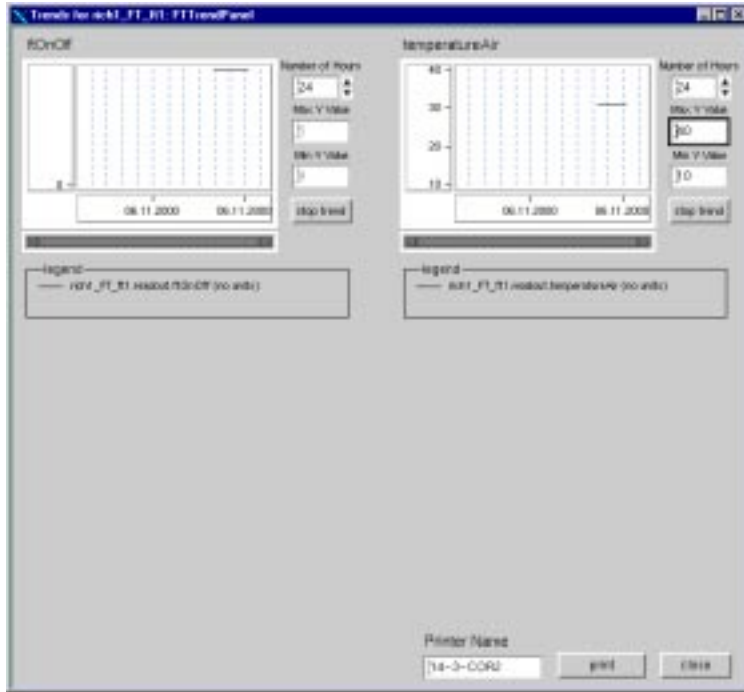


Figure 71: FTrendPanel panel.

4.2.17 digitalVariableScaleTrend

This panel (Figure 72) is used to trend a boolean variable, the dpelement has to be passed as a \$-parameter. It is also called by the actionPanel (Section 4.2.12, "actionPanel") for AI type element.

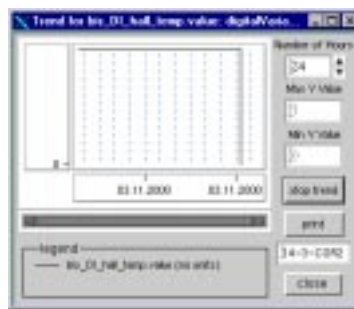


Figure 72: digitalVariableScaleTrend panel.

4.2.18 HVTable panel

The HVTable panel (Figure 73) is used to manipulate HV settings of all the CAEN_HV_ch and CAEN_HV_ch_SP elements which are underneath the node from which is called that panel even if there is a multiple hierarchy of HV systems, crates and HV groups. One can

⁴⁹ The IT-CO front-end software is returning a value less than -900 if the sensor is not existing in the fan-tray.

get the current value of all the CAEN_HV_ch and CAEN_HV_ch_SP elements, change them, load them from a file, save them in a file and download them on the data point.

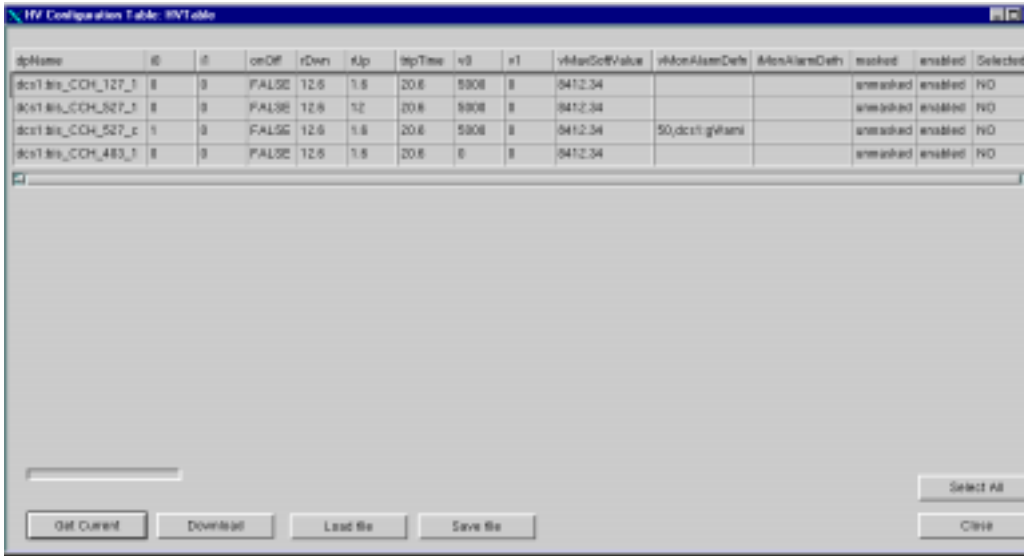


Figure 73: HVTable panel.

The data are downloaded to the data points, saved in a file if the “Selected” field is set to “yes”.

4.2.19 CAENHVBoard

The \$-parameter of this panel is: \$CAENHVBoard which is the data point name.

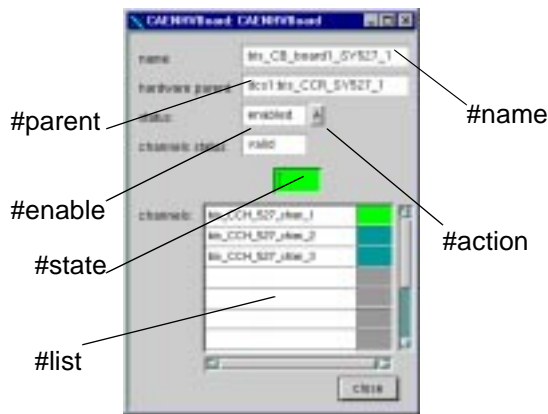


Figure 74: CAENHVBoard panel.

4.2.19.1 #list

Shows all the channels inside board. The order of the channels is equivalent to the order in the HV-board, empty field means that the channel is not yet defined, calls the elementTable

panel (Section 4.3.38, "elementTable"). The HV channels are printed at their physical location within the HV board.

4.2.19.2#enable

Shows the status of board (enable/disable). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.19.3#name

Displays the name of datapoint (content of \$CAENHVBoard).

4.2.19.4#parent

Displays the crate name in which is plugged the board. A double click on the parent opens the corresponding run-time panel.

4.2.19.5#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$CAENHVBoard is given to this panel, e.g. "\$dp: dcs1:rich_CB_board1_127_1".

4.2.19.6#state

Shows the state of the board. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.20 SY127

The \$-parameter of this panel is: \$\$SY127 which is the data point name.

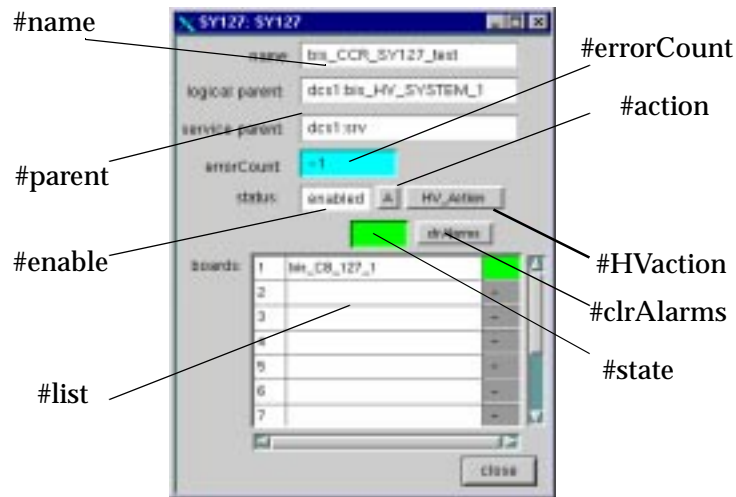


Figure 75: SY127 panel.

4.2.20.1#list

Shows all the boards inside HV crate. The order of the boards is equivalent to the order in the HV-crate, empty field means that the channel is not yet defined, calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.20.2#enable

Shows the status of HV-crate (enable/disable). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.20.3#name

Displays the name of datapoint (content of \$\$SY127).

4.2.20.4#parent

Shows the location (parent: DETECTOR) of the SY127 element and the service parent. A double click on the parent opens the corresponding run-time panel.

4.2.20.5#errorCount

Shows the number of consecutive accessing failure to the crate by the IT-CO front-end software. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.20.6#state

Shows the state the SY127 crate. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.20.7#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$\$SY127 is given to this panel.

4.2.20.8#HVaction

Opens the HV_actionPanel: Section 4.2.13, "HV_ActionPanel". \$\$SY127 is given to this panel.

4.2.20.9#clrAlarms

To clear the alarm in the SY127 crate (refer to the SY127 documentation).

4.2.21 SY403

The \$-parameter of this panel is: \$\$SY403 which is the data point name.

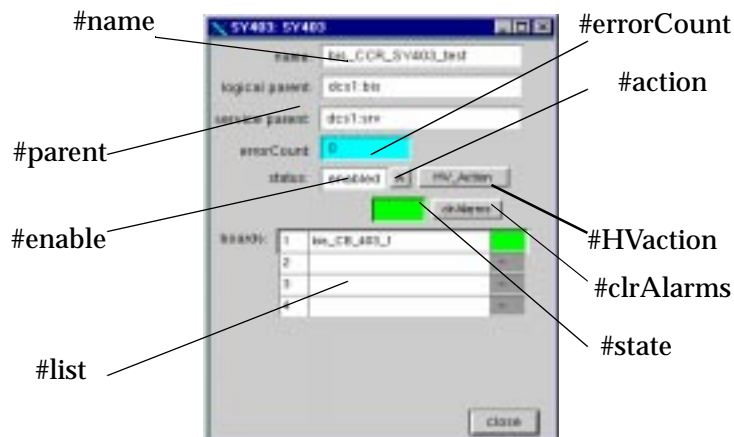


Figure 76: SY403 panel.

4.2.21.1#list

Shows all the boards inside HV crate. The order of the boards is equivalent to the order in the HV-crate, empty field means that the channel is not yet defined, calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.21.2#enable

Shows the status of HV-crate (enable/disable). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.21.3#name

Displays the name of datapoint (content of \$SY403).

4.2.21.4#parent

Shows the location (parent: DETECTOR) of the SY403 element and the service parent. A double click on the parent opens the corresponding run-time panel.

4.2.21.5#errorCount

Shows the number of consecutive accessing failure to the crate by the IT-CO front-end software. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.21.6#state

Shows the state of the SY403 crate. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.21.7#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SY403 is given to this panel.

4.2.21.8#HVaction

Opens the HV_actionPanel: Section 4.2.13, "HV_ActionPanel". \$\$SY403 is given to this panel.

4.2.21.9#clrAlarms

To clear the alarm in the SY403 crate (refer to the SY403 documentation).

4.2.22 SY527

The \$-parameter of this panel is: \$\$SY527 which is the data point name.

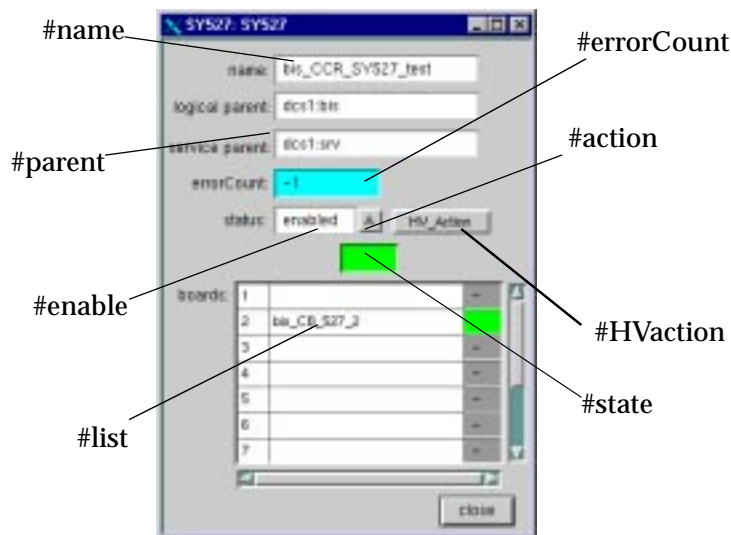


Figure 77: SY527 panel.

4.2.22.1#list

Shows all the boards inside HV crate. The order of the boards is equivalent to the order in the HV-crate, empty field means that the channel is not yet defined, calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.22.2#enable

Shows the status of HV-crate (enable/disable). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.22.3#name

Displays the name of datapoint (content of \$\$SY527).

4.2.22.4#parent

Shows the location (parent: DETECTOR) of the SY527 element and the service parent. A double click on the parent opens the corresponding run-time panel.

4.2.22.5#errorCount

Shows the number of consecutive accessing failure to the crate by the IT-CO front-end software. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.22.6#state

Shows the state of the SY527 crate. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.22.7#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$\$SY527 is given to this panel.

4.2.22.8#HVaction

Opens the HV_actionPanel: Section 4.2.13, "HV_ActionPanel". \$\$SY527 is given to this panel.

4.2.23 PROCS

The \$-parameter of this panel is: \$PROCS which is the data point name. A double click on the parent opens the corresponding run-time panel.

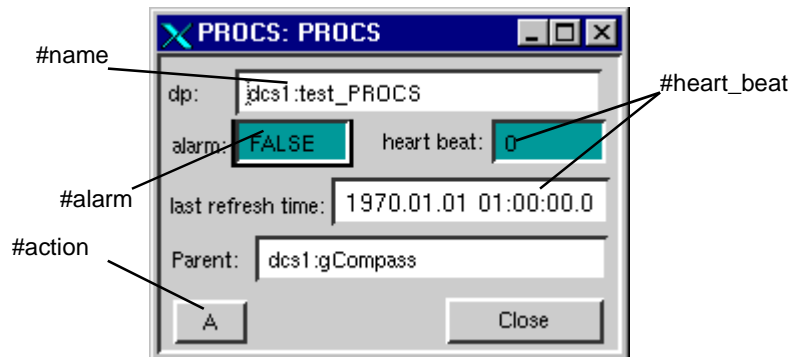


Figure 78: PROCS panel.

4.2.23.1#name

Displays the name of datapoint (content of \$PROCS).

4.2.23.2#alarm

Shows the status of the alarm summary of the PROCS element: a value TRUE means that an alarm for the PROCS element occurred and a value FALSE means that no alarm occurred (Section 3.9, "monitoring the activity of external process"). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.23.3#heart_beat

Shows the activity of the PROCS element and the last time it was refreshed (Section 3.9, "monitoring the activity of external process"). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.23.4#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$PROCS is given to this panel, e.g. "\$dp:dcs1:test_PROCS".

4.2.24 HV_GROUP panel

The \$-parameter of this panel is: \$HV_GROUP which is the data point name.

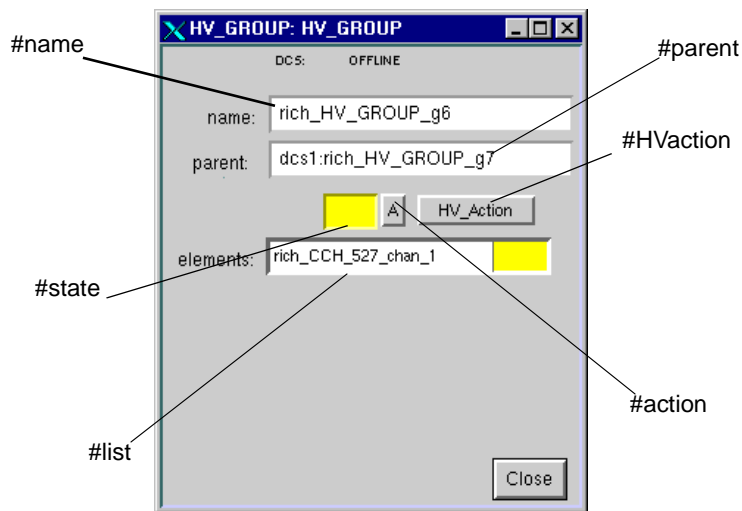


Figure 79: HV_GROUP panel.

4.2.24.1#list

Shows the list of child element of the HV_GROUP: elements (CAEN_HV_ch, CAEN_HV_ch_SP and HV_GROUP), calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.24.2#name

Displays the name of datapoint (content of \$HV_GROUP).

4.2.24.3#parent

Shows the location (parent: DETECTOR, HV_SYSTEM, SUB_SYSTEM or HV_GROUP) of the HV_GROUP element. A double click on the parent opens the corresponding run-time panel.

4.2.24.4#state

Shows the state of the HV_GROUP. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.24.5#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$HV_GROUP is given to this panel.

4.2.24.6#HVaction

Opens the HV_actionPanel: Section 4.2.13, "HV_ActionPanel". \$HV_GROUP is given to this panel.

4.2.25 HV_SYSTEM panel

The \$-parameter of this panel is: \$HV_SYSTEM which is the data point name.

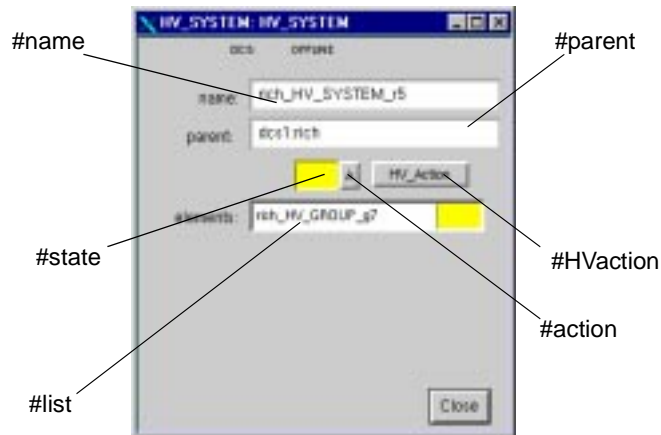


Figure 80: HV_SYSTEM panel.

4.2.25.1#list

Shows the list of child element of the HV_SYSTEM: elements (CAEN_HV_ch, CAEN_HV_ch_SP, HV_SYSTEM and HV_GROUP), calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.25.2#name

Displays the name of datapoint (content of \$HV_SYSTEM).

4.2.25.3#parent

Shows the location (parent: DETECTOR, SUB_SYSTEM or HV_SYSTEM) of the HV_SYSTEM element. A double click on the parent opens the corresponding run-time panel.

4.2.25.4#state

Shows the state of the HV_SYSTEM. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.25.5#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$HV_SYSTEM is given to this panel.

4.2.25.6#HVaction

Opens the HV_actionPanel: Section 4.2.13, "HV_ActionPanel". \$HV_SYSTEM is given to this panel.

4.2.26 SUB_SYSTEM panel

The \$-parameter of this panel is: \$SUB_SYSTEM which is the data point name.

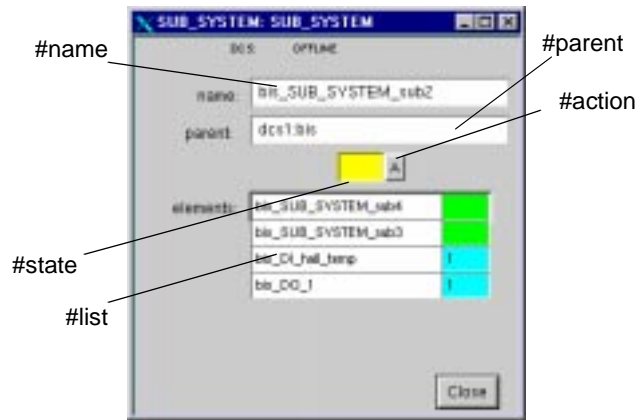


Figure 81: SUB_SYSTEM panel.

4.2.26.1#list

Shows the list of child element of the SUB_SYSTEM: elements (AI, AO, AO_F, DI, DO, DO_F, HV_SYSTEM, HV_GROUP or SUB_SYSTEM), calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.26.2#name

Displays the name of datapoint (content of \$SUB_SYSTEM).

4.2.26.3#parent

Shows the location (parent: DETECTOR or SUB_SYSTEM) of the SUB_SYSTEM element. A double click on the parent opens the corresponding run-time panel.

4.2.26.4#state

Shows the state of the SUB_SYSTEM. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.26.5#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SUB_SYSTEM is given to this panel, e.g. "\$dp:dcs1:bis_SUB_SYSTEM_sub2".

4.2.27 SERVICE panel

The \$-parameter of this panel is: \$SERVICE which is the data point name. A double click on the parent opens the browseCompassTree panel (Section 4.2.1, "browseCompassTree").

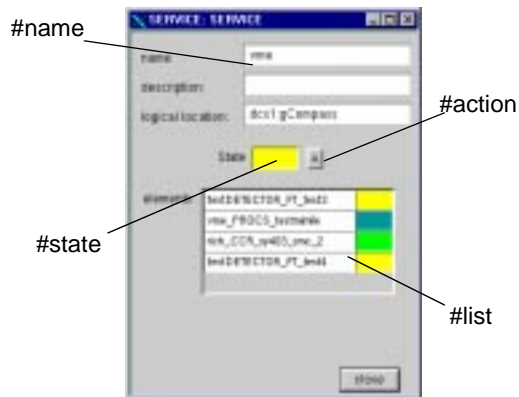


Figure 82: SERVICE panel.

4.2.27.1#list

Shows the list of child element of the SERVICE: elements (HWD, SY127, SY403, SY527, FAN_TRAY and PROCS), calls the elementTable panel (Section 4.3.38, "elementTable").

4.2.27.2#name

Displays the name of datapoint (content of \$SERVICE).

4.2.27.3#state

Shows the state of the SERVICE. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.2.27.4#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SERVICE is given to this panel, e.g. "\$dp:dcs1:PLC2".

4.2.28 FAN_TRAY panel

The \$-parameter of this panel is: \$FAN_TRAY which is the data point name.

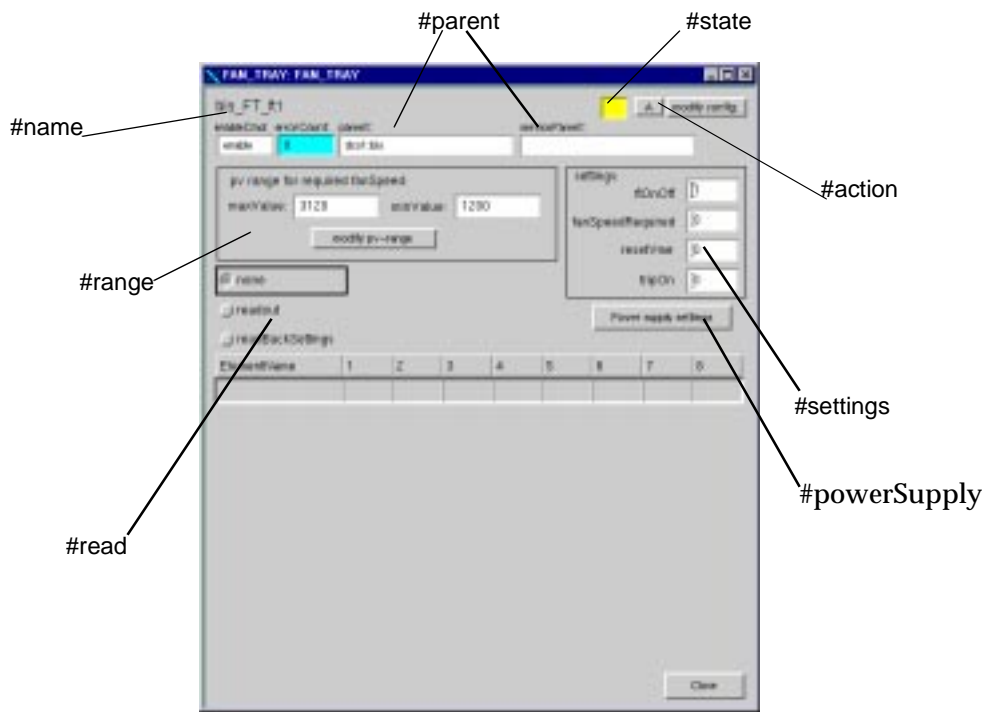


Figure 83: FAN_TRAY panel: no read data.



Figure 84: FAN_TRAY panel: readout data.



Figure 85: FAN_TRAY panel: readback settings data.

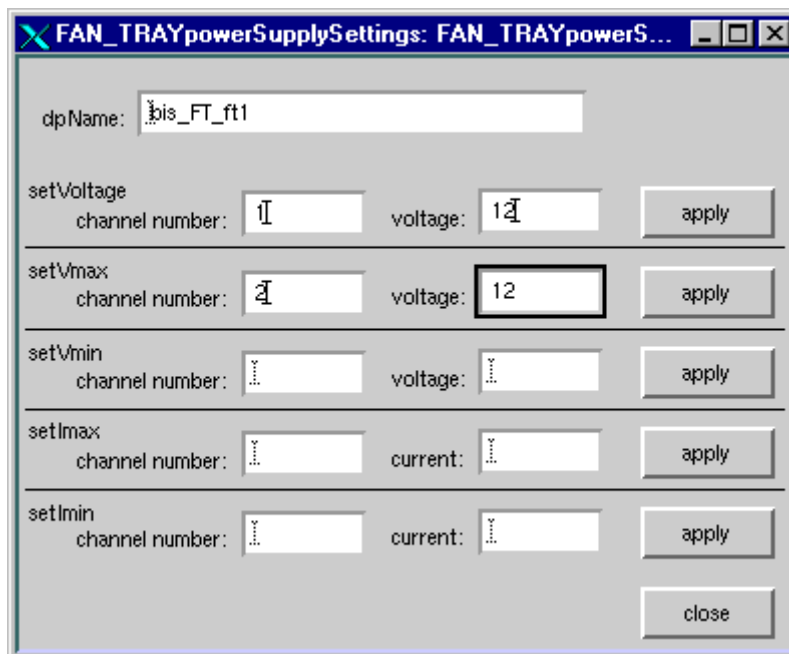


Figure 86: FAN_TRAYpowerSupplySettings panel: setting the power supply value.

4.2.28.1#name

Displays the name of datapoint (content of \$FAN_TRAY).

4.2.28.2#state

Shows the state of the FAN_TRAY. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.28.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SERVICE is given to this panel, e.g. "\$dp:dcs1:PLC2".

4.2.28.4#parent

Shows the parent dp name of the FAN_TRAY: DETECTOR type, SERVICE type.

4.2.28.5#range

Shows the pv_range of the fanSpeedRequired dpElement.

4.2.28.6#settings

Shows the settings of the FAN_TRAY element.

4.2.28.7#read

The "none" case of the radio button displays nothing, the "readout" case of the radio button displays the readout data (Figure 84) of the FAN_TRAY element and the "readBackSettings" case of the radio button displays the readBackSettings (Figure 85) data of the FAN_TRAY. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.2.28.8#powerSupply⁵⁰

Call the panel FAN_TRAYpowerSupplySettings (Figure 86). The channel number has to be between 1 to 8.

4.2.29 Alert panel

The alert panel (Figure 87) is a PVSS panel which presents the list of active alarm⁵¹ in the PVSS system. The masked alarm are not shown. This panel can be called from user customized panels or from the icon bar of the VISION program.

⁵⁰ This is not possible to set the power supply voltage limits in the IT-CO front-end software.

⁵¹ If the alert panel is opened and one defined a new alarm on an element, when this alarm occurs it will not be seen in the alert panel. One has to close and open again the alert panel: this is a PVSS limitation.

sho	prio	time	data-point element/multi-lingual	alert text	direct	value	ack	acknowledgement
F	100	03/07/00 14:48:21	rich_AI_1.value	Bad value	CAME	12345	xxx	03/07/00 14:48:23
F	80	04/07/00 14:18:05	bis_HWD_1.powersError	bad	CAME	1	!!!	
W	60	06/07/00 11:33:48	bis_DO_1.value	bad	CAME	TRUE		
W	60	06/07/00 11:33:51	bis_DO_1.value	bad	WENT	FALSE	!!!	
F	80	07/07/00 15:12:08	bis_DO_F_1.readValue	Bad value	CAME	TRUE		
F	80	07/07/00 15:12:11	bis_DO_F_1.readValue	Bad value	WENT	FALSE	!!!	
W	60	07/07/00 15:20:37	bis_DI_1.value	Bad value	CAME	TRUE	!!!	
F	80	13/07/00 14:11:23	bis_HWD_1	Bad	CAME	TRUE		
F	100	17/07/00 14:23:44	kjhkh	Bad	CAME	TRUE		
F	100	17/07/00 14:23:44	bis_SUB_SYSTEM_sub2	Bad	CAME	TRUE		
F	100	17/07/00 14:23:44	bis_AI_1.value	Bad value	CAME	8	x	17/07/00 20:38:50
W	60	17/07/00 20:38:15	muon_HWD_1	Bad	CAME	TRUE		
W	60	17/07/00 20:38:15	muon_DO_F_1.readValue	Bad value	CAME	TRUE	x	17/07/00 20:38:48
W	60	17/07/00 20:38:15	muon_AO_F_3.readValue	dfg	CAME	-1	!!!	
W	60	17/07/00 20:38:15	muon_AO_F_2.readValue	Bad value	CAME	1	x	17/07/00 20:38:48
W	60	17/07/00 20:38:15	muon_DI_3.value	Bad value	CAME	TRUE	x	17/07/00 20:38:48

Figure 87: alert panel.

The alarms can be acknowledge from that panel, but this acknowledge does not have the same functionality as the one defined in Section 3.4, "action on the framework elements". In the 'ack' column one can have the following value:

- '!!!': the alarm can be acknowledge, this is not the same acknowledge as the one described in Section 3.3, "Alarms and states".
- 'x': the alarm has been acknowledge inside the alert screen panel.
- 'xxx': the alarm has been acknowledge by using the framework acknowledge described in Section 3.4, "action on the framework elements".

Different filter settings can be selected by choosing one of the following choice from the configuration combo box (Section "Alarms"):

- all: all the active alarms and all the active summary alarms.
- default: all the active alarms without the active summary alarms.
- warning: all the active alarms of the warning category.
- fault: all the active alarms of the fault category.
- fatal: all the active alarms of the fatal category.

It is also possible to change the filter settings if desired by selecting in the configuration combo box the '.' choice and by pressing the 'properties...' button (Figure 88). The default filter setting is selected when the alert panel is opened.

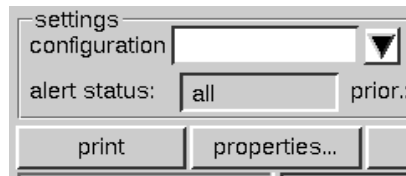


Figure 88: Load new config.

A comment (Figure 89) can be added by clicking in the 'com' column (#com).

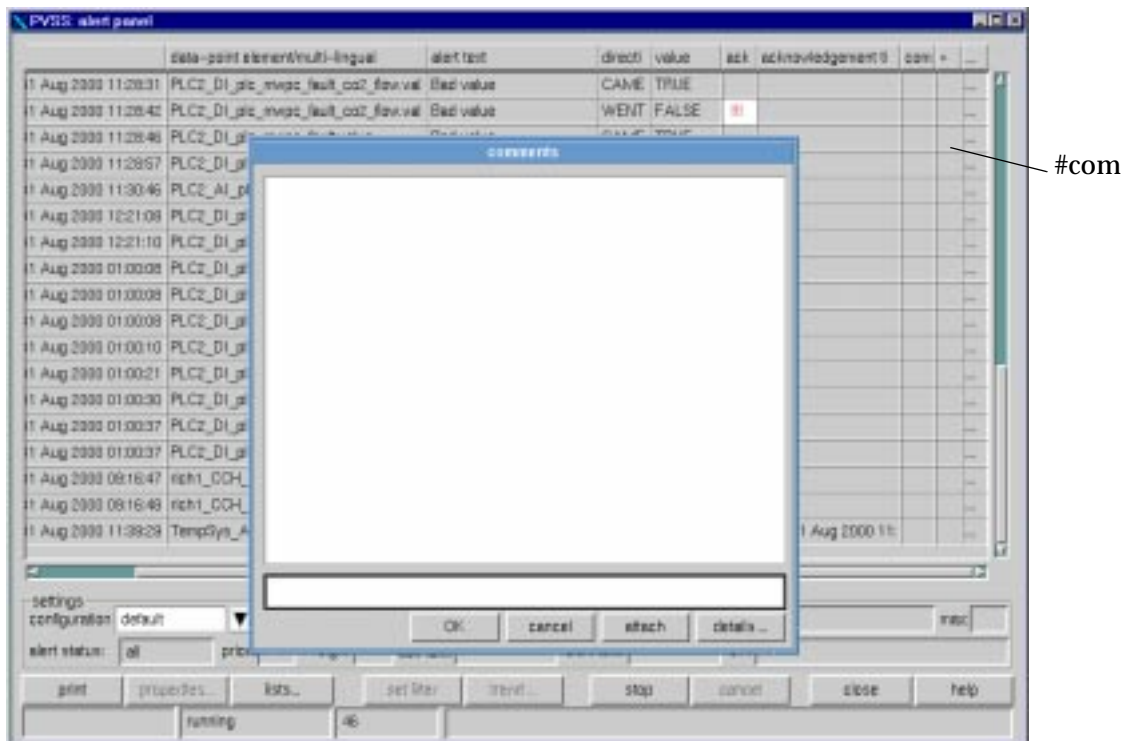


Figure 89: alertPanel comment.

More information (Figure 90) on the alarm of an element can be queried by clicking in the '...' column (#info).

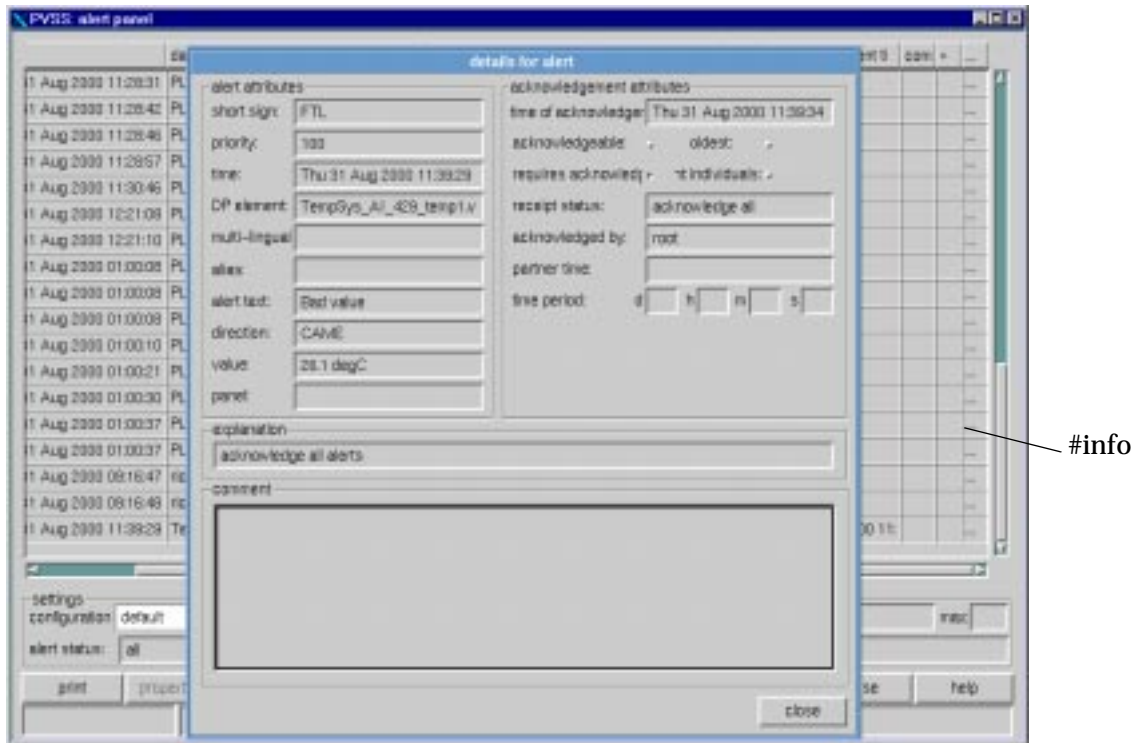


Figure 90: alertPanel info on alarm.

4.2.30 Event panel

The event panel (Figure 91) is a PVSS panel which presents the list of event in the PVSS system. This panel can be called from user customized panels or from the icon bar of the VISION program. By clicking in the column '...' a window 'detail for event' is pop-up. The default filter setting is selected when the event panel is opened. The filter can be modified by selecting in the configuration combo box the '.' choice and by pressing the 'properties...' button.

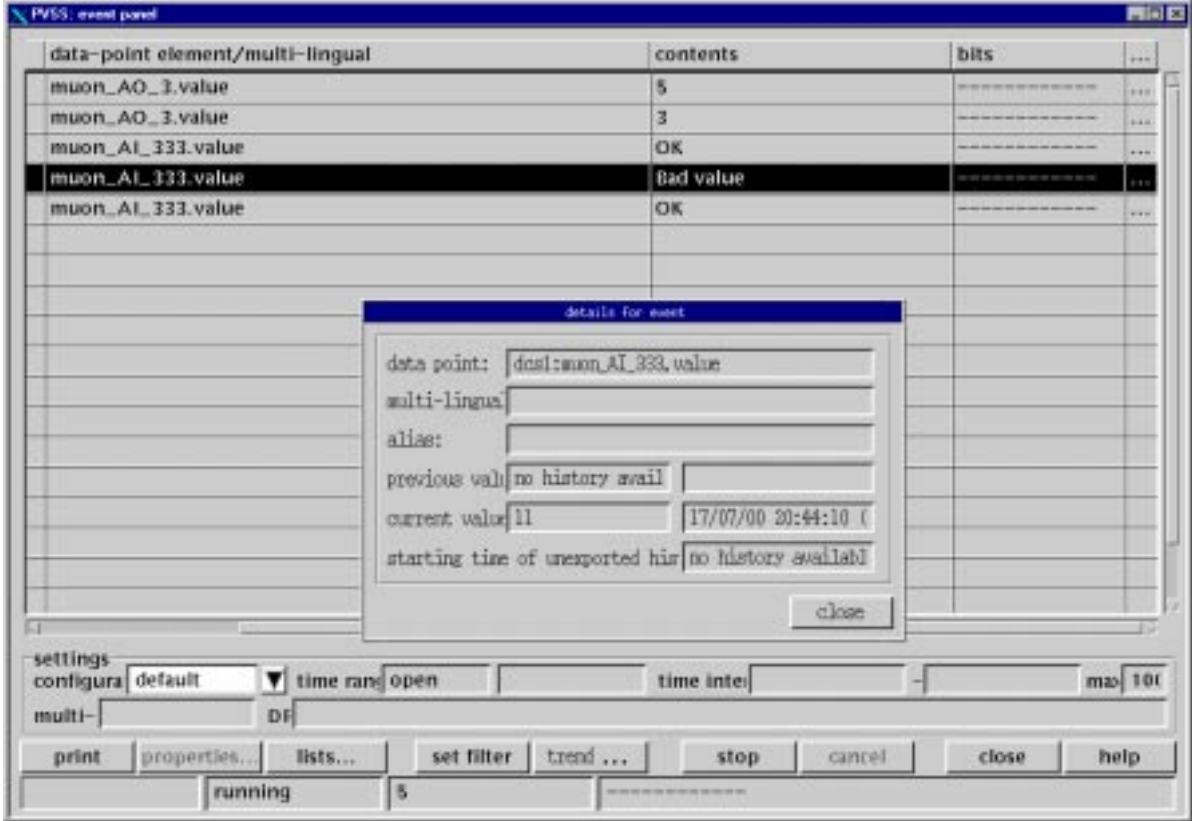


Figure 91: Event panel.

4.3 Reference panels

Reference panels are ready made "components" which can be added into user customized panel by clicking in 'panel reference...' icon (Figure 92) from the PVSS graphical editor: GEDI. These panels follow the color convention defined in Section "Background color convention (Figure 53) for leaf element.". A double click on the reference panel will open the run-time panel of the element.



Figure 92: panel reference icon from the GEDI program.

The user, developing panels, can then pick up a reference panel and copy it in his own panel. He can create a view of his system by mixing reference panels of elements coming from different part of the COMPASS tree. All the run-time panels are designed as reference panels and can be used as such by a user developing customized panels. Panels like: variableScaleTrend (Section 4.2.14, "variableScaleTrend"), HVTrendPanel (Section 4.2.15, "HVTrendPanel"), etc. can be used in any panels.

4.3.1 AISummary

The \$-parameter of this panel is: \$AI which is the data point name.

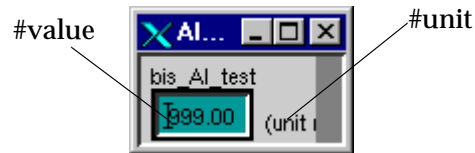


Figure 93: AISummary panel.

4.3.1.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 93 the background color is _ETM: the element is disabled. If the 'value' dpElement of the AI element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.1.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.2 AISummaryAct

The \$-parameter of this panel is: \$AI which is the data point name.

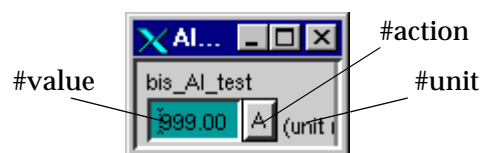


Figure 94: AISummaryAct panel.

4.3.2.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 94 the background color is _ETM: the element is disabled. If the 'value' dpElement of the AI element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.2.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.2.3 #action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$AI is given to this panel, e.g. "\$dp:dcs1:rich_AI_testAi".

4.3.3 AOSummary

The \$-parameter of this panel is: \$AO which is the data point name.

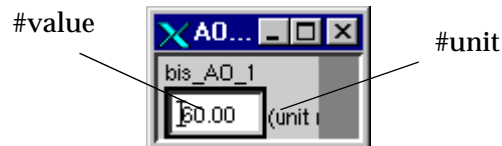


Figure 95: AOSummary panel.

4.3.3.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.3.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.4 AOSummaryAct

The \$-parameter of this panel is: \$AO which is the data point name.

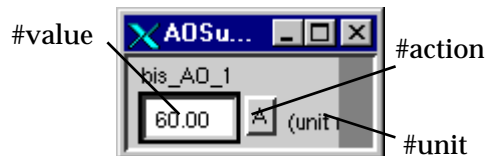


Figure 96: AOSummaryAct panel.

4.3.4.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.4.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.4.3 action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$AO is given to this panel, e.g. "\$dp:dcs1:rich_AO_1".

4.3.5 AO_FValSummary

The \$-parameter of this panel is: \$AO_F which is the data point name.

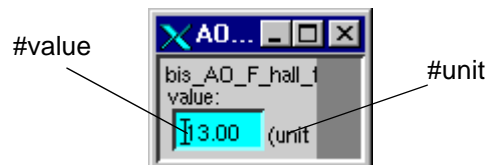


Figure 97: AO_FValSummary panel.

4.3.5.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 97 the background color is cyan: the value dpElement is invalid. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.5.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.6 AO_FRValSummary

The \$-parameter of this panel is: \$AO_F which is the data point name.

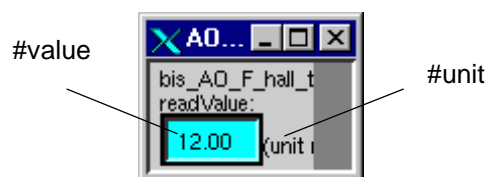


Figure 98: AO_FRValSummary panel.

4.3.6.1 #value

Shows the value of 'readValue' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 98 the background color is cyan: the value dpElement is invalid. If the 'readValue' dpElement of the AO_F element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'readValue' dpElement by typing new value to text field and pressing 'enter'.

4.3.6.2 #unit

Shows the unit of 'readValue' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.7 AO_FSummaryAct

The \$-parameter of this panel is: \$AO_F which is the data point name.

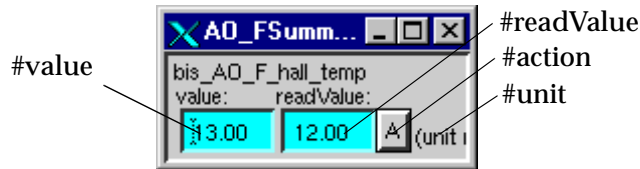


Figure 99: AO_FSummaryAct.

4.3.7.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 99 the background color is cyan: the value dpElement is invalid. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.7.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.7.3 #readValue

Shows the value of 'readValue' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 99 the background color is cyan: the value dpElement is invalid. If the 'readValue' dpElement of the AO_F element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'readValue' dpElement by typing new value to text field and pressing 'enter'.

4.3.7.4 #action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$AO_F is given to this panel.

4.3.8 DISummary

The \$-parameter of this panel is: \$DI which is the data point name.

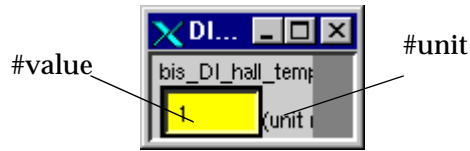


Figure 100: DISummaryPanel.

4.3.8.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 100 the background color is yellow: the element has an alarm in the Warning category. If the 'value' dpElement of the DI element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.8.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.9 DISummaryAct

The \$-parameter of this panel is: \$DI which is the data point name.



Figure 101: DISummaryAct panel.

4.3.9.1 #value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 101 the background color is yellow: the element has an alarm in the Warning category. If the 'value' dpElement of the DI element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.9.2 #unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.9.3 #action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DI is given to this panel.

4.3.10 DOSummary

The \$-parameter of this panel is: \$DO which is the data point name.

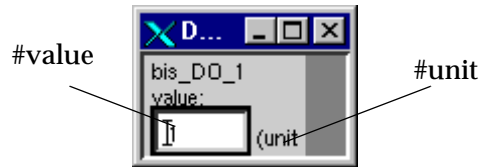


Figure 102: DOSummary.

4.3.10.1#value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.10.2#unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.11 DOSummaryAct

The \$-parameter of this panel is: \$DO which is the data point name.

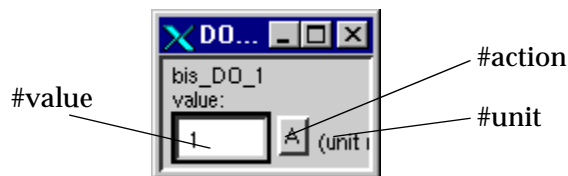


Figure 103: DOSummaryAct.

4.3.11.1#value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.11.2#unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.11.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DO is given to this panel.

4.3.12 DO_FValSummary

The \$-parameter of this panel is: \$DO_F which is the data point name.

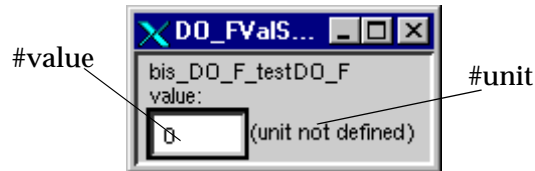


Figure 104: DO_FValSummary panel.

4.3.12.1#value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. E.g. in Figure 104 the background color is white: the value is has no alarm defined. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.12.2#unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.13 DO_FRValSummary

The \$-parameter of this panel is: \$DO_F which is the data point name.

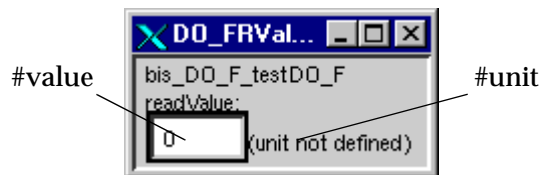


Figure 105: DO_FRValSummary panel.

4.3.13.1#value

Shows the value of 'readValue' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. If the 'readValue' dpElement of the DO_F element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'readValue' dpElement by typing new value to text field and pressing 'enter'.

4.3.13.2#unit

Shows the unit of 'readValue' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.14 DO_FSummaryAct

The \$-parameter of this panel is: \$DO_F which is the data point name.

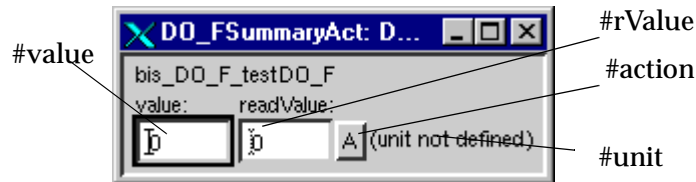


Figure 106: DO_FSummaryAct panel.

4.3.14.1#value

Shows the value of 'value' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The user can set the new value for 'value' dpElement by typing new value to text field and pressing 'enter'.

4.3.14.2#unit

Shows the unit of 'value' dpElement. If user hasn't defined a unit "(unit not defined)" is displayed.

4.3.14.3#rValue

Shows the value of 'readValue' dpElement. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. If the 'readValue' dpElement of the DO_F element does not have a peripheral address (Section "Peripheral address"), it is possible to set the new value for 'readValue' dpElement by typing new value to text field and pressing 'enter'.

4.3.14.4#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DO_F is given to this panel.

4.3.15 HWDSummary

The \$-parameter of this panel is: \$HWD which is the data point name.

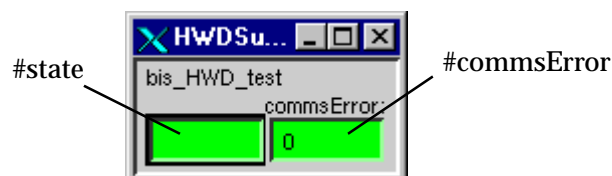


Figure 107: HWDSummary panel.

4.3.15.1#state

Shows the state of the HWD. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.15.2#commsError

Shows the different communication error when a HWD is connected to an OPC Server (Section 3.7, "OPC"): 0=OK, 1=OPC server not connected, 2=PVSS OPC Client not connected. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.16 HWDSummaryAct

The \$-parameter of this panel is: \$HWD which is the data point name.

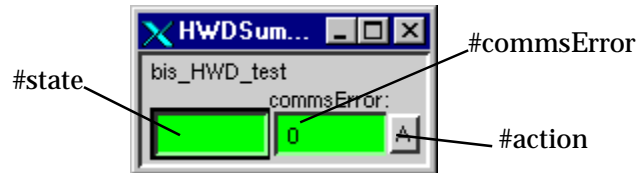


Figure 108: HWDSummaryAct panel.

4.3.16.1#state

Shows the state of the HWD. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.16.2#commsError

Shows the different communication error when a HWD is connected to an OPC Server (Section 3.7, "OPC"): 0=OK, 1=OPC server not connected, 2=PVSS OPC Client not connected. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.16.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$HWD is given to this panel.

4.3.17 DETECTORSummary

The \$-parameter of this panel is: \$DETECTOR which is the data point name.



Figure 109: DETECTORSummary panel.

4.3.17.1#name

Name of the DETECTOR element.

4.3.17.2#state

Shows the state of the DETECTOR. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.18 DETECTORSummaryAct

The \$-parameter of this panel is: \$DETECTOR which is the data point name.



Figure 110: DETECTORSummaryAct panel.

4.3.18.1#name

Name of the DETECTOR element.

4.3.18.2#state

Shows the state of the DETECTOR. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.18.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$DETECTOR is given to this panel.

4.3.19 DETECTORBannerSummary

The \$-parameter of this panel is: \$DETECTOR which is the data point name.

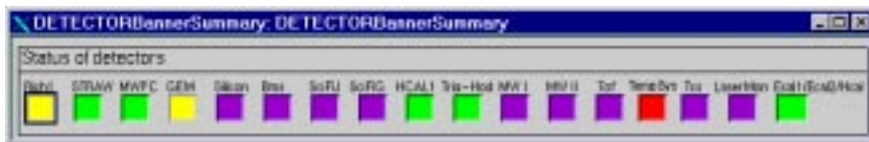


Figure 111: DETECTORBannerSummary panel.

This is a banner which shows all the detectors summary alarm status. The background color follows the rule: Background color convention (Figure 53) for node element. described above. It can be used in any user customized panels to show the status of the detectors of the COMPASS experiment, this can be useful to locate from where are the problems.

4.3.20 CAEN_HV_ch

The \$-parameter of this panel is: \$CAEN_HV_ch which is the data point name.

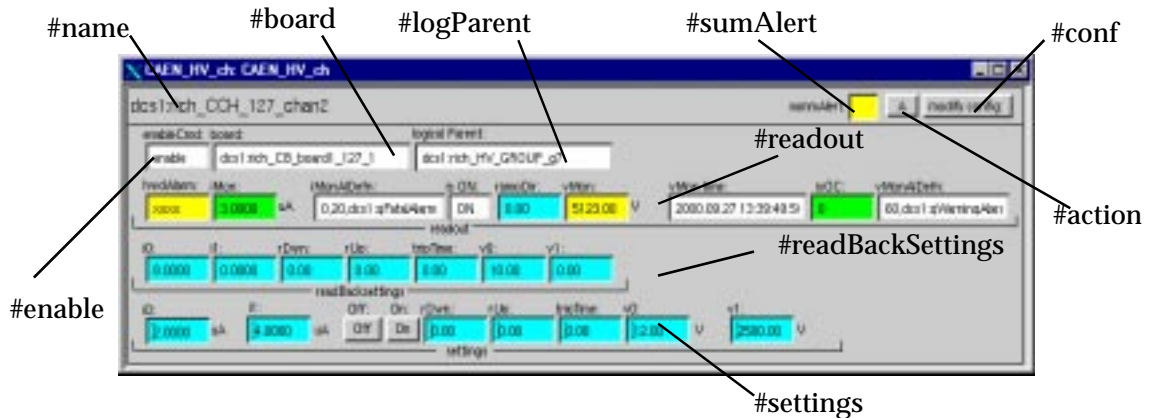


Figure 112: CAEN_HV_ch panel.

4.3.20.1#name

Name of the HV channel.

4.3.20.2#enable

Shows the status of HV-channel (enable/disable). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.20.3#board

Shows the board the CAEN_HV_ch element belongs to. A double click opens the CAENHVBoard panel (Section 4.2.19, "CAENHVBoard").

4.3.20.4#logParent

Shows the logical parent of the CAEN_HV_ch: a HV_SYSTEM or HV_GROUP element. A double click opens the corresponding run-time panel.

4.3.20.5#settings

Text field which shows the value of the data to set to the CAEN_HV_ch. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. New settings for the HV channel can be entered by setting a new value in the appropriate text field and pressing 'enter' or by pushing the button 'On' or 'Off' to switch on/off the HV channel.

4.3.20.6#readBackSettings

Text field which shows the value of the settings read back from the CAEN_HV_ch. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.20.7#readout

Text field which shows the value of the readout data of the CAEN_HV_ch. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The hardwareAlarm element can have the value xxxx: unknown, ovv: over-voltage, none: ok, trip: channel has tripped or uvv: under-voltage.

4.3.20.8#sumAlert

Shows the status and short sign of the alarm summary of the CAEN_HV_ch. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.20.9#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$CAEN_HV_ch is given to this panel.

4.3.20.10#conf

Opens the CAEN_HV_chAlertConfig panel (Section 4.1.24, "CAEN_HV_chAlertConfig").

4.3.21 CAEN_HV_ch_SP

The \$-parameter of this panel is: \$CAEN_HV_ch_SP which is the data point name.

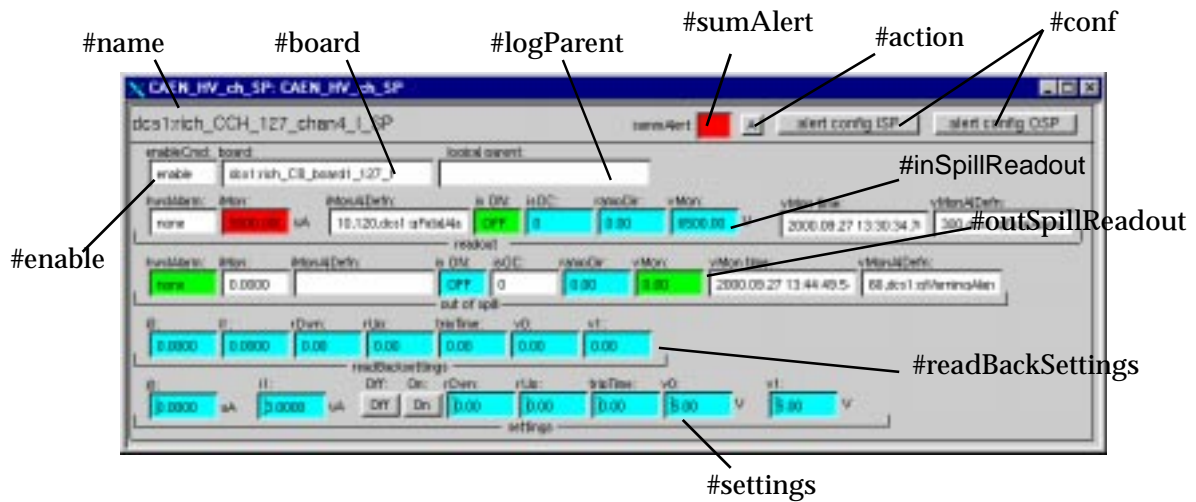


Figure 113: CAEN_HV_ch_SP panel.

4.3.21.1#name

Name of the HV channel.

4.3.21.2#enable

Shows the status of HV-channel (enable/disable). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.21.3#board

Shows the board the CAEN_HV_ch_SP element belongs to. A double click opens the CAENHVBoard panel (Section 4.2.19, "CAENHVBoard").

4.3.21.4#logParent

Shows the logical parent of the CAEN_HV_ch_SP: a HV_SYSTEM or HV_GROUP element. A double click opens the corresponding run-time panel.

4.3.21.5#settings

Text field which shows the value of the data to set to the CAEN_HV_ch_SP. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. New settings for the HV channel can be entered by setting a new value in the appropriate text field and pressing 'enter' or by pushing the button 'On' or 'Off' to switch on/off the HV channel.

4.3.21.6#readBackSettings

Text field which shows the value of the setting read back from the CAEN_HV_ch_SP. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.21.7#inSpillReadout

Text field which shows the value of the readout data of the CAEN_HV_ch_SP read in the spill. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The hardwareAlarm element can have the value xxx: unknown, ovv: over-voltage none: ok, trip: channel has tripped or uvv: under-voltage.

4.3.21.8#outSpillReadout

Text field which shows the value of the readout data of the CAEN_HV_ch_SP read out of the spill. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The hardwareAlarm element can have the value xxx: unknown, ovv: over-voltage none: ok or uvv: under-voltage.

4.3.21.9#sumAlert

Shows the status and short sign of the alarm summary of the CAEN_HV_ch_SP. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.21.10#action

Opens the actionPanel: Section 4.2.12, "actionPanel". SCAEN_HV_ch_SP is given to this panel.

4.3.21.11#conf

Opens the CAEN_HV_chAlertConfig panel (Section 4.1.24, "CAEN_HV_chAlertConfig") or the CAEN_HV_ch_SPAlertConfig panel (Section 4.1.27, "CAEN_HV_ch_SPAlertConfig").

4.3.22 channel

The \$-parameter of this panel is: \$CAEN_HV_ch which is the data point name.

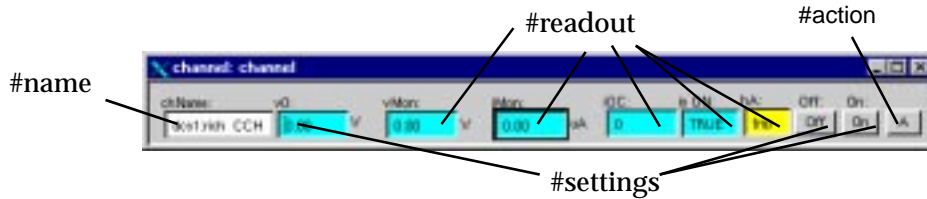


Figure 114: channel panel.

4.3.22.1#name

Name of the HV channel.

4.3.22.2#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$CAEN_HV_ch is given to this panel.

4.3.22.3#readout

Text field which shows the value of the readout data of the CAEN_HV_ch. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The hardwareAlarm element can have the value xxx: unknown, ovv: over-voltage none: ok, trip: channel has tripped or uvv: under-voltage.

4.3.22.4#settings

Text field which shows the value of data to set to the CAEN_HV_ch. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. New settings for the v0 can be entered by setting a new value in the text field and pressing 'enter' or by pushing the button 'On' or 'Off' to switch on/off the HV channel.

4.3.23 channelSP

The \$-parameter of this panel is: \$CAEN_HV_chSP which is the data point name.

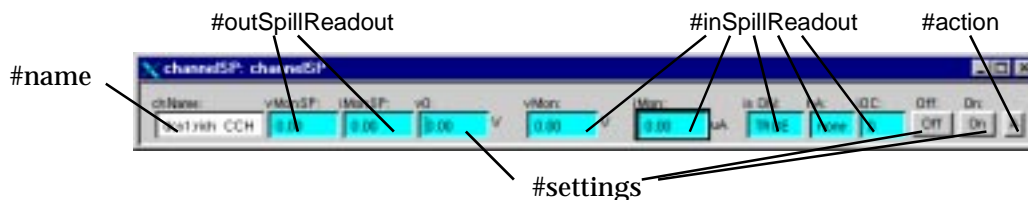


Figure 115: channelSP panel.

4.3.23.1#name

Name of the HV channel.

4.3.23.2#inSpillReadout

Text field which shows the value of the readout data of the CAEN_HV_ch_SP read in the spill. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. The hardwareAlarm element can have the value xxx: unknown, ovv: over-voltage none: ok, trip: channel has tripped or uvv: under-voltage.

4.3.23.3#outSpillReadout

Text field which shows the value of the readout data of the CAEN_HV_ch_SP read out of the spill. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.23.4#settings

Text field which shows the value of data to set to the CAEN_HV_ch_SP. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above. New settings for the v0 can be entered by setting a new value in the text field and pressing 'enter' or by pushing the button 'On' or 'Off' to switch on/off the HV channel.

4.3.23.5#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$CAEN_HV_ch_SP is given to this panel.

4.3.24 chAlarmSummary

The \$-parameter of this panel is: \$CAEN_HV_ch which is the data point name.

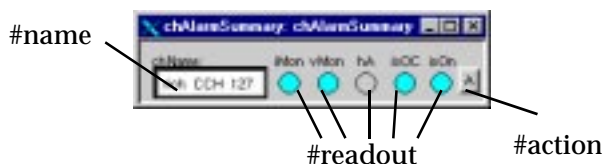


Figure 116: chAlarmSummary panel.

4.3.24.1#name

Name of the HV channel.

4.3.24.2#readout

Text field which shows the status of value of the readout data of the CAEN_HV_ch. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.24.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$CAEN_HV_ch is given to this panel.

4.3.25 chSPAlarmSummary

The \$-parameter of this panel is: \$CAEN_HV_ch_SP which is the data point name.

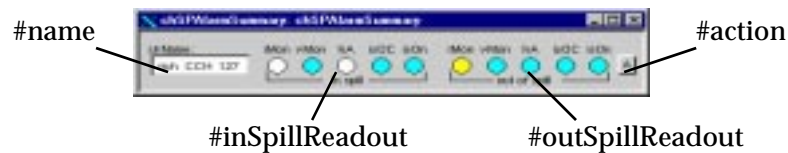


Figure 117: chSPAlarmSummary panel.

4.3.25.1#name

Name of the HV channel.

4.3.25.2#inSpillReadout

Shows the status of the value of the readout data of the CAEN_HV_ch_SP read in the spill. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.25.3#outSpillReadout

Shows the status of the value of the readout data of the CAEN_HV_ch_SP read out of the spill. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.25.4#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$CAEN_HV_ch_SP is given to this panel.

4.3.26 SY127Summary

The \$-parameter of this panel is: \$SY127 which is the data point name.

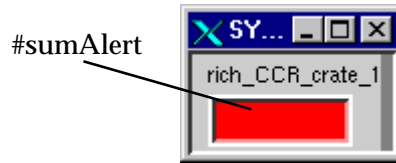


Figure 118: SY127Summary panel.

4.3.26.1#state

Shows the state of the SY127. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.27 SY127SummaryAct

The \$-parameter of this panel is: \$SY127 which is the data point name.

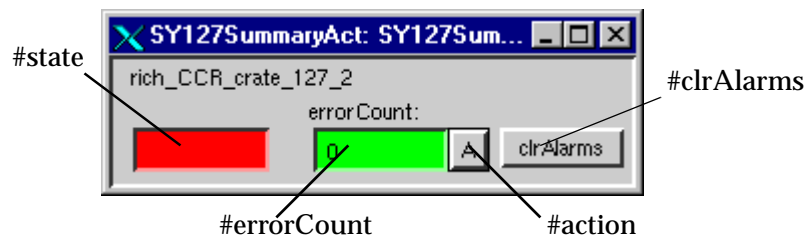


Figure 119: SY127SummaryAct panel.

4.3.27.1#state

Shows the state of the SY127. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.27.2#errorCount

Shows the number of consecutive accessing failure to the crate by the IT-CO front-end software. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.27.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SY127 is given to this panel.

4.3.27.4#clrAlarms

To clear the alarm in the SY127 crate (refer to the SY127 documentation).

4.3.28 SY403Summary

The \$-parameter of this panel is: \$SY403 which is the data point name.

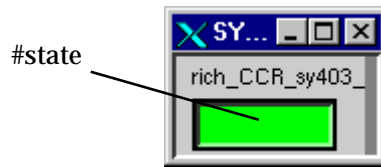


Figure 120: SY403Summary panel.

4.3.28.1#state

Shows the state of the SY403. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.29 SY403SummaryAct

The \$-parameter of this panel is: \$SY403 which is the data point name.

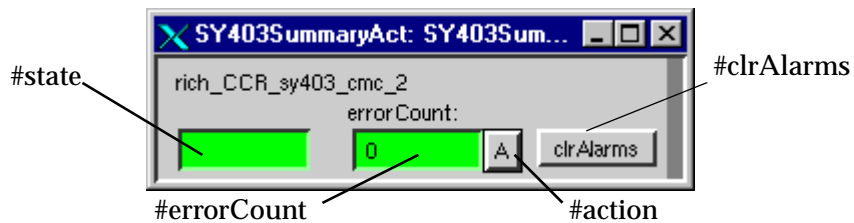


Figure 121: SY403SummaryAct panel.

4.3.29.1#state

Shows the state of the SY403. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.29.2#errorCount

Shows the number of consecutive accessing failure to the crate by the IT-CO front-end software. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.29.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SY403 is given to this panel.

4.3.29.4#clrAlarms

To clear the alarm in the SY403 crate (refer to the SY403 documentation).

4.3.30 SY527Summary

The \$-parameter of this panel is: \$SY527 which is the data point name.

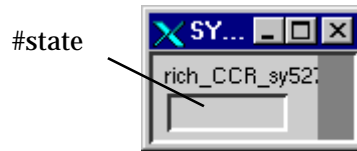


Figure 122: SY527Summary panel.

4.3.30.1#state

Shows the state of the SY527. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.31 SY527SummaryAct

The \$-parameter of this panel is: \$SY527 which is the data point name.

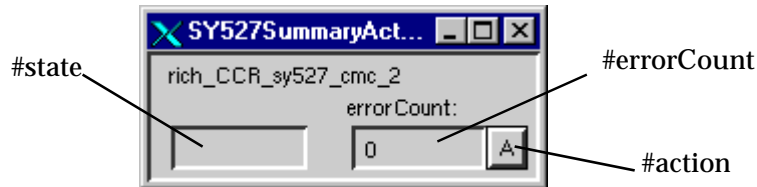


Figure 123: SY527SummaryAct panel.

4.3.31.1#state

Shows the state of the SY527. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.31.2#errorCount

Shows the number of consecutive accessing failure to the crate by the IT-CO front-end software. The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

4.3.31.3#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SY527 is given to this panel.

4.3.32 HV_GROUPSSummary

The \$-parameter of this panel is: \$HV_GROUP which is the data point name.



Figure 124: HV_GROUPSSummary panel.

4.3.32.1#state

Shows the state of the HV_GROUP. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.33 HV_GROUPSSummaryAct

The \$-parameter of this panel is: \$HV_GROUP which is the data point name.



Figure 125: HV_GROUPSSummaryAct panel.

4.3.33.1#state

Shows the state of the HV_GROUP. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.33.2#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$HV_GROUP is given to this panel.

4.3.34 HV_SYSTEMSummary

The \$-parameter of this panel is: \$HV_SYSTEM which is the data point name.

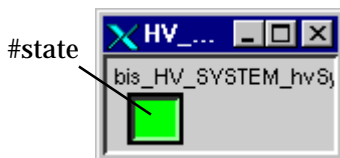


Figure 126: HV_SYSTEMSummary panel.

4.3.34.1#state

Shows the state of the HV_SYSTEM. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.35 HV_SYSTEMSummaryAct

The \$-parameter of this panel is: \$HV_SYSTEM which is the data point name.

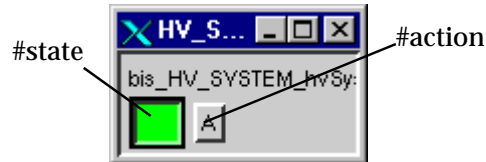


Figure 127: HV_SYSTEMSummaryAct panel.

4.3.35.1#state

Shows the state of the HV_SYSTEM. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.35.2#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$HV_SYSTEM is given to this panel.

4.3.36 SUB_SYSTEMSummary

The \$-parameter of this panel is: \$SUB_SYSTEM which is the data point name.

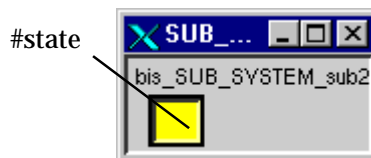


Figure 128: SUB_SYSTEMSummary panel.

4.3.36.1#state

Shows the state of the SUB_SYSTEM. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.37 SUB_SYSTEMSummaryAct

The \$-parameter of this panel is: \$SUB_SYSTEM which is the data point name.

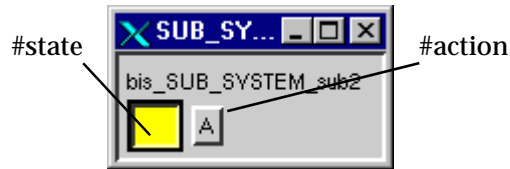


Figure 129: SUB_SYSTEMSummaryAct panel.

4.3.37.1#state

Shows the state of the SUB_SYSTEM. The background color follows the rule: Background color convention (Figure 53) for node element. described above.

4.3.37.2#action

Opens the actionPanel: Section 4.2.12, "actionPanel". \$SUB_SYSTEM is given to this panel.

4.3.38 elementTable

This panels shows the content of a list of elements. In the left column the dp name is printed, in the right column the status from the fwk_status dp element is shown. The background color follows the rule: Background color convention (Figure 53) for node element. and Background color convention (Figure 53) for leaf element. described above, a dark grey background color means that the element has no fwk_status. For AI, AO, AO_F, DI, DO and DO_F the value dpElement is also printed in the right column. The \$-parameter is the name of a list of element. The run-time of the known element types is opened when double clicking on the element name.

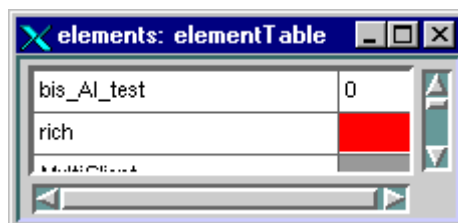


Figure 130: elementTable panel.

4.3.39 FAN_TRAYSummy

The \$-parameter of this panel is: \$FAN_TRAY which is the data point name.

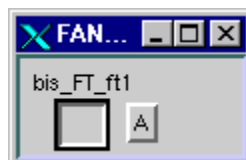


Figure 131: FAN_TRAYSummy panel.

4.3.40 FAN_TRAYSSummaryAct

The \$-parameter of this panel is: \$FAN_TRAY which is the data point name.

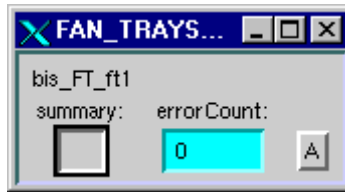


Figure 132: FAN_TRAYSSummaryAct panel.

4.3.41 FT

The \$-parameter of this panel is: \$FAN_TRAY which is the data point name. The readout data of the FAN_TRAY are shown. The switch ON/OFF setting and the required fan speed setting can be changed.



Figure 133: FT panel.

4.3.42 FTAlarmSummary

The \$-parameter of this panel is: \$FAN_TRAY which is the data point name.

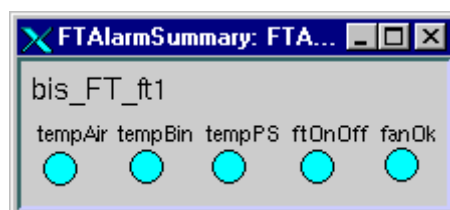


Figure 134: FTAlarmSummary panel.

4.3.43 trend

The \$-parameter of this panel is: \$dp which is the data point name.

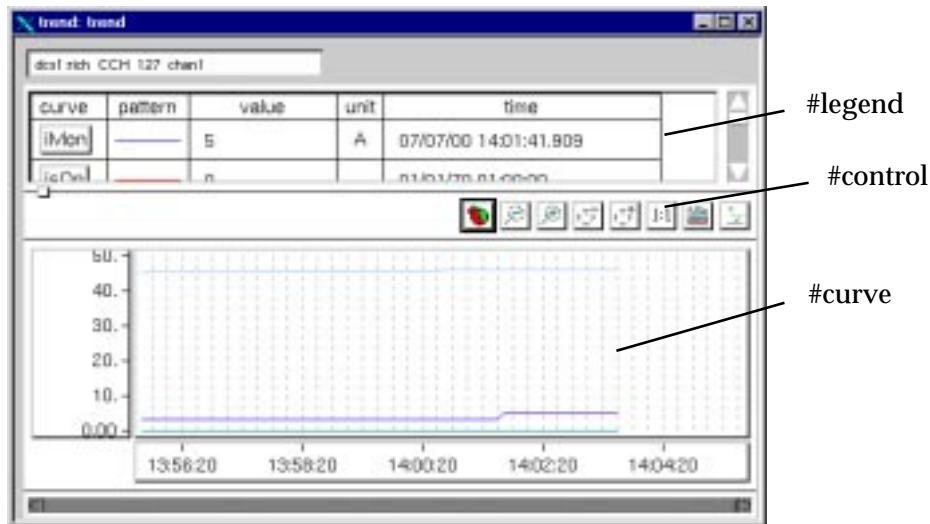


Figure 135: trend panel.

The trend panel can be use to view the run-time and historical value (if they were archived) of the AI, AO, AO_F, DI, DO, DO_F, CAEN_HV_ch and CAEN_HV_ch_SP elements.

4.3.43.1#legend

Legend of the trend, curve can be remove or added by clicking on the square box of the dpElement.

4.3.43.2#curve

Run-time trend and historical trend (if data was archived), the trend data dpElement are:

- AI: value.
- AO: value.
- AO_F: value and readValue.
- DI: value.
- DO: value.
- DO_F: value.
- CAEN_HV_ch: iMon, vMon, isOn, isOvc, i0, i1, v0, v1.
- CAEN_HV_ch_SP: in spill: iMon, vMon, isOn, isOvc; out spill: iMon, vMon; readBackSettings: i0, i1, v0, v1.

4.3.43.3#control

The time scale, y scale can be modified, etc. refer the PVSS documentation for more detail.

4.3.44 variableScaleMultiTrend

This panel (Figure 136) is used to trend up to 8 curves. The dp list has to be given to the panel (Figure 137).

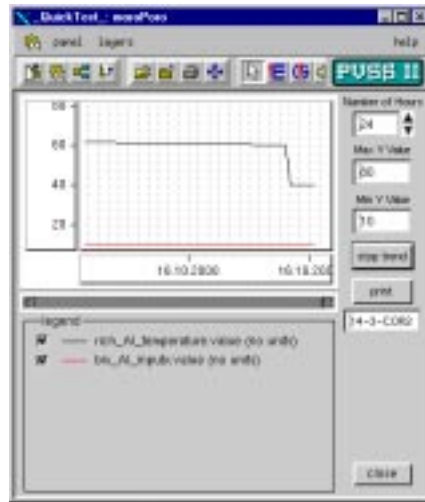


Figure 136: variableScaleMultiTrend panel.



Figure 137: variableScaleMultiTrend configuration parameter list.

4.3.45 PROCSSummary

The \$-parameter of this panel is: \$PROCS which is the data point name. The #alarm shows the status of the alarm summary of the PROCS element and the #heart_beat show the activity of the PROCS element (Section 3.9, "monitoring the activity of external process"). A double click opens the PROCS panel (Section 4.2.23, "PROCS"). The background color follows the rule: Background color convention (Figure 53) for leaf element. described above.

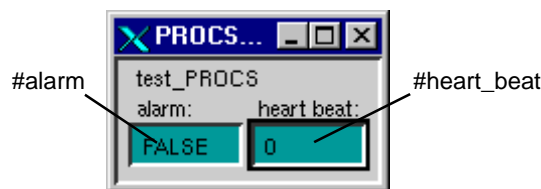


Figure 138: PROCSSummary panel.

4.3.46 SERVICESummary

The \$-parameter of this panel is: \$SERVICE which is the data point name. The #state shows the state of the SERVICE. The background color follows the rule: Background color convention (Figure 53) for node element. described above.



Figure 139: SERVICESummary panel.

4.3.47 SERVICESummaryAct

The \$-parameter of this panel is: \$SERVICE which is the data point name. The #state shows the state of the SERVICE. The background color follows the rule: Background color convention (Figure 53) for node element. described above. The #action opens the actionPanel: Section 4.2.12, "actionPanel". \$SUB_SYSTEM is given to this panel.



Figure 140: SERVICESummaryAct panel.

4.3.48 MaskedList

This panel list all the elements which are masked and enabled. The panel is not refreshed automatically when one elements is masked or un-masked, the "refresh" button refresh the list. A double click on one element of the list opens the corresponding run-time panel.

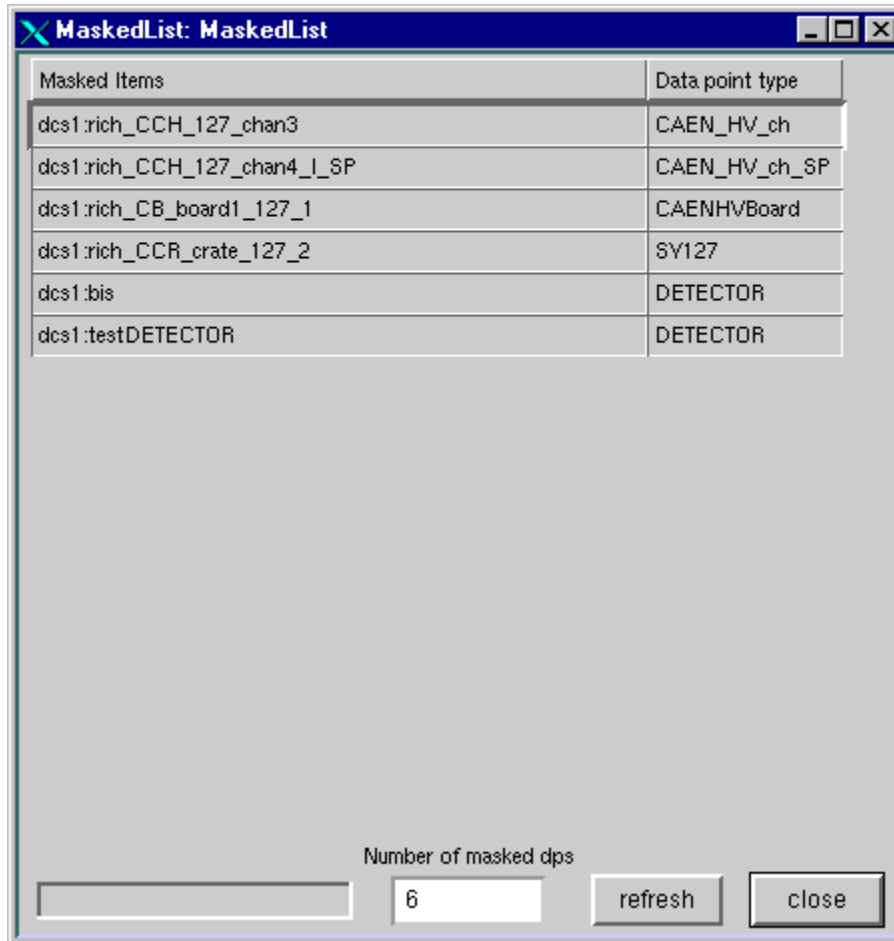


Figure 141: MaskedList panel.

4.3.49 DisableList

This panel list all the elements which are disabled (not enabled). The panel is not refreshed automatically when one elements is enabled or disabled, the "refresh" button refresh the list. A double click on one element of the list opens the corresponding run-time panel.

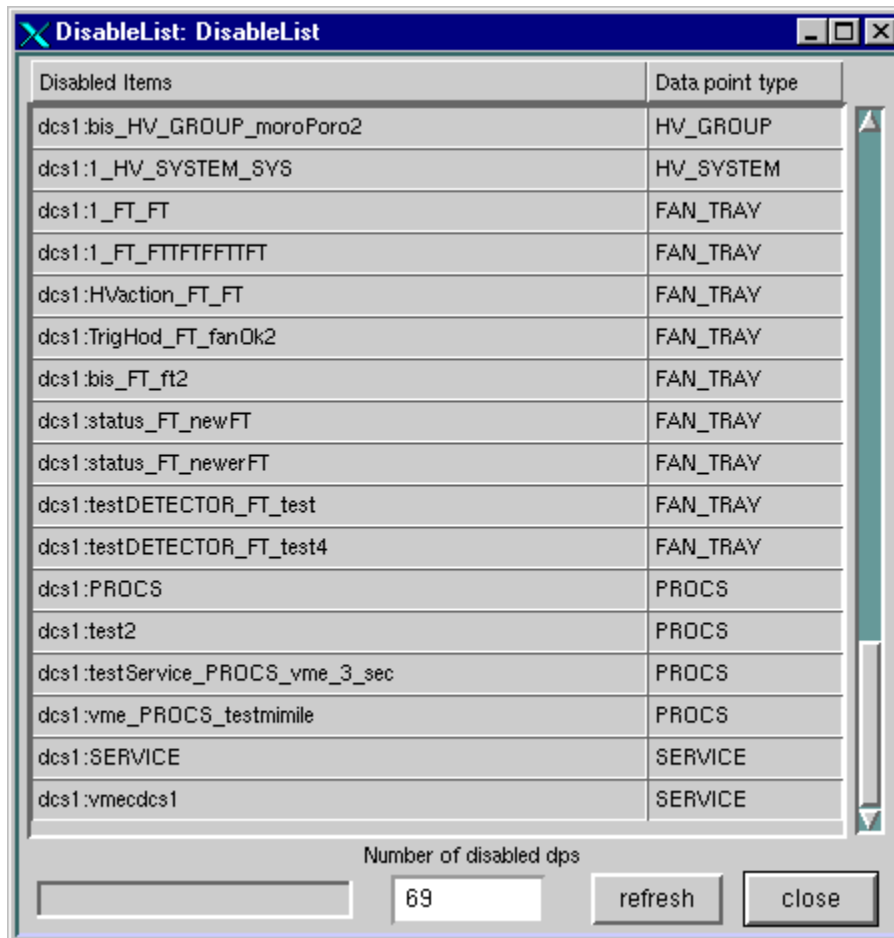


Figure 142: DisableList panel.

4.3.50 ElementList

This panel list all the elements of the system. The panel is not refreshed automatically when one elements is added or removed, the “refresh” button refresh the list. A double click on one element of the list opens the corresponding run-time panel.

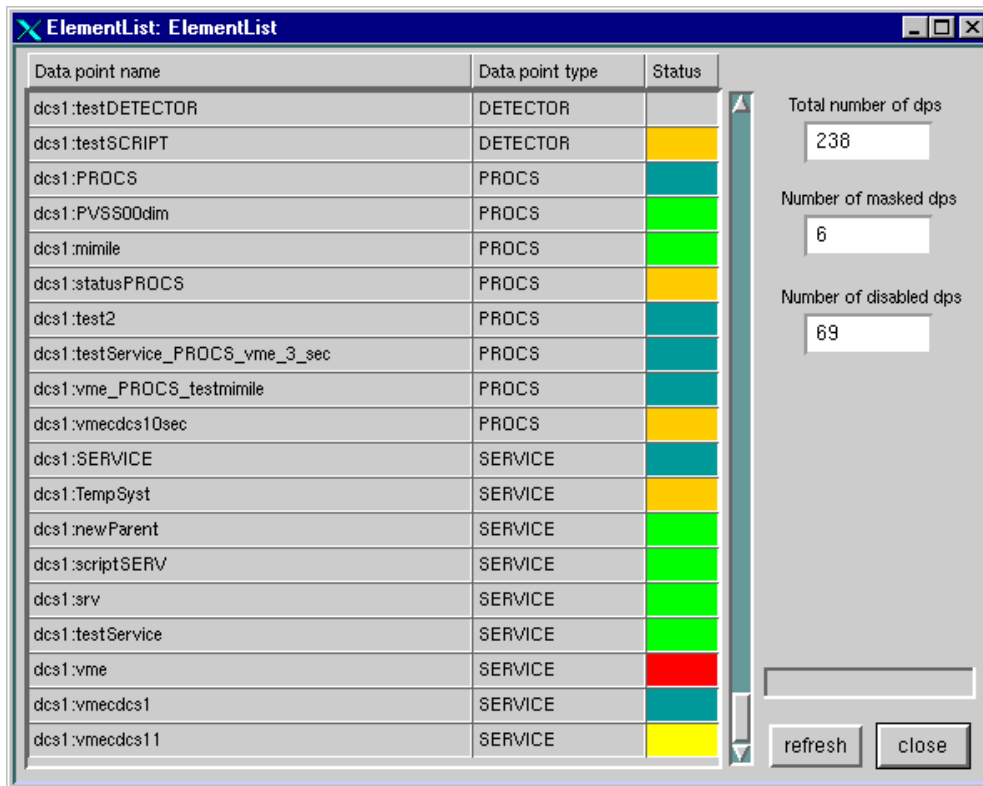


Figure 143: ElementList panel.

4.3.51 AlertRow

The AlertRow panel (Figure 144) is a PVSS panel which can be incorporated in a user customized panel. It shows the list of active alarm⁵² in the PVSS system in a row format. The masked alarm are not shown. The 'THAN' button opens the alert panel (Figure 87). The #pos is the alert number of the current shown in the row list over the quantity of alert. The #nav button is used for navigation upward or backward in the list, the alarm is shown in the #list. The #list shows the current alert selected with the #nav buttons.



Figure 144: AlertRow panel.

⁵² If the panel containing the AlertRow is opened and one defined a new alarm on an element, when this alarm occurs it will not be seen in the AlertRow. One has to close and open again the panel containing the AlertRow: this is a PVSS limitation.

5 Framework Libs

The libraries are used by the run-time, reference and modification panels to query the different config used by the compass framework. They are located in the PROJECT_PATH/common/scripts/libs folder or in PROJECT_PATH/common-hv/scripts/libs.

All the functions of these libraries have a parameter variable: dyn_string &errorText⁵³. The errorText variable shows the errors which occur within the function. An empty errorText means that there was no error. The compDisplayError function (Section 5.9, "fwk_libGUI.ctl") can be called to show the errorText in the errorDisplay panel (Section 4.1.38, "errorDisplay").

5.1 HVC_user.ctl

These library functions are called by the HV configuration panels used to import the HV configuration of the IT-CO front-end: Section 4.1.39, "ImportHVConfig" and the HV config panel: Section 4.1.41, "HVConfig". The available functions are:

- HVC_userRoutineAfterHVChannelsCreated: called after the HV channel creation, it set the peripheral address to DIM, create the alert summary of the channel data point, adds this alert summary in the alert summary of the crate the channel belongs to, and set the units of all the voltage and current dpElements.
- HVC_userRoutineAfterHVBoardIsCreated: called after the HV board creation.
- HVC_userRoutineAfterHVPowerSupplyIsCreated: called after the HV crate (SY127, SY403 or SY527) creation, it set the peripheral address to DIM, creates the alert summary of the crate data point and adds the crate in the hwdTree of the detector the create belongs to.
- HVC_userRoutineBeforeHVChannelsDeleted: called before the HV channel deletion, it removes the dp from the summary alert of the crate the channel belongs to.
- HVC_userRoutineBeforeHVBoardIsDeleted: called before the HV board deletion.
- HVC_userRoutineBeforeHVPowerSupplyIsDeleted: called before the HV crate (SY127, SY403 or SY527) deletion, it removes the dp from the summary alert of the hwdTree of the detector the crate belongs to.

⁵³ This feature was developed by Christian Schuegger, TU Munchen.

5.2 fwk_DIMlib.ctl

This library is used to configure the DIMCLIENT interface from the PARA module and from the compass framework. The following functions are available:

- fwk_dim_getDpType: get the type of the dp.
- fwk_saveClientService: creates the address config for the dpElement connected to a DIM Service.
- fwk_saveClientCommand: creates the address config for the dpElement connected to a DIM Command.
- fwk_isClientService: query if the dpElement is connected to a DIM Service.
- fwk_isClientComplexService: query if the dpElement is connected to a complex DIM Service⁵⁴.
- fwk_isClientCommand: query if the dpElement is connected to a DIM Command.
- fwk_isGetClientServiceDefaults: get the DIM Service default.
- fwk_isGetClientCommandDefaults: get the DIM Command default.

5.3 fwk_complibElementCommon.ctl⁵⁵

This library contains a set of library function of common use to get the characteristics of the elements. Refer to the html documentation for more information.

5.4 fwk_complibListConfig.ctl

The tree hierarchy of the COMPASS DCS is build as a set of list. This library contains functions handling list:

- compAddDpInList: add a dp in a list of PVSS dp ids.
- compAddDpInFixedLengthList: add a dp in a fixed length list of PVSS dp ids.
- compRemoveDpFromList: remove a dp from a list of PVSS dp ids.

The functions from this library have to be used with care, they may not work with user defined dp type because they set the “methodName:_original.._value” dp element of the selected dp.

5.5 fwk_libAddressConfig.ctl

This library contains a set of functions to query, delete and set the peripheral address config for OPC and DIM SERVER. Refer to the documentation of the file to have more information on the parameter passed to the functions. The following functions are available:

- compPeriphAddressOPC: set the peripheral address config of a dpElement, e.g. AI.value to an OPC Item. OPC Group, the type of the OPC Item and driver number

⁵⁴ This feature is not supported by the PARA module, however the PVSS00dim manager supports it. Currently the compass framework does not support it as it is, this has to be configured manually in the startup file of the manager.

⁵⁵ This library was developed by Christian Schuegger, TU Munchen.

are automatically set according the OPC Server selected, e.g. the dpElement value of an AI element has a data type of input, etc.

- compDeletePeriphAddress: delete an existing peripheral address
- compPerpiphAddressDIMSERVER: set the peripheral address config of a dpElement to a data interfaced via DIM.
- compGetPeriphAddress: get the peripheral address config of a dpElement.
- compIsPeriphAddressHadChanged: check if the periph address config had changed.

The peripheral address config for a DIMCLIENT is not supported by this library, the functions of the fwk_libDIM.ctl file (Section 5.2, "fwk_DIMlib.ctl") have to be used instead.

5.6 fwk_libAlertConfig.ctl

This library contains the functions to query, delete and set the alert config⁵⁶ of the dpElement. Refer to the documentation of the file to have more information on the parameter passed to the functions. The following functions have to be used only by the panels running in whatever system which has a name different than 'PRODUCTION':

- compAlertConfigDigital: set a digital alert config of a dpElement. Internally in PVSS the alert config will be deleted and created.
- compDeleteAlertConfigDigital: delete a digital alert config of a dpElement.
- compAlertConfigAnalog: set an analog alert config of a dpElement, this function calls other functions depending on the number of threshold given to it, in the release v1.4, only two thresholds alert for an analogue value is supported. Internally in PVSS the alert config will be deleted and created.
- compDeleteAlertConfigAnalog: delete an analog alert config of a dpElement.
- compCreateAlertConfigSum: create a summary alert of a dpElement. Internally in PVSS the alert config will be deleted and created.
- compDeleteAlertConfigSum: delete a summary alert of a dpElement.

The following functions have to be used by the panels for all the PVSS systems:

- compAddDpAlertSum: add a dp which has an alert config defined in a summary alert list. The alert is just modified (there is no deletion and no creation).
- compDeleteDpAlertSum: remove a dp which has an alert config defined from a summary alert list. The alert is just modified (there is no deletion and no creation).
- compModAlertConfigSum: modify a summary alert config of a dpElement.
- compModAlertConfigDigital: modify a digital alert config of a dpElement.
- compModAlertConfigAnalog: modify an analog alert config of a dpElement. This function calls other functions depending on the number of threshold given to it, in the release v1.4, only two thresholds alert for an analogue value is supported

⁵⁶ a boolean dpElement can only have an digital alert config, a number dpElement can have an analog alert config. A struct and string dpElement can only have an alert summary config.

5.7 **fwk_libArchiveConfig.ctl**

This library contains the functions to query, delete and set the archive config of the dpElement. Refer to the documentation of the file to have more information on the parameter passed to the functions. The following functions are available:

- compDeleteArchiveConfig: delete the archive config of a dpElement.
- compArchiveConfig: set the archive config of a dpElement.
- compGetArchiveConfig: query the archive config of a dpElement.

5.8 **fwk_libCompassDriver.ctl**

This library is used for the OPC configuration of the elements of the framework. The following functions are available:

- compRefreshList: trigger the dcsDrvMain.ctl (Section 6.2, "dcsDrvMain.ctl"), to refresh its internal list of elements connected to an OPC Server.
- compGetDrvId: get the PVSS OPC Client driver manager number connected to the OPC Server.

5.9 **fwk_libGUI.ctl**

This library is used by all the modification panels when applying the setting to the elements. The available function are:

- a set of functions called by the modification panels (Section 4.1, "Modification panels").
- raiseError: create the error text used in all the functions of the libraries.
- guiDebugMessage: print on the stdout and stderr the error of the graphical interface text message.
- compDisplayError: to display in the errorDisplay panel (Section 4.1.38, "errorDisplay"), the error set by any functions of the libraries.

5.10 **fwk_libHWDConfig.ctl**

HWD elements can be connected to an OPC Server (Section 6.2, "dcsDrvMain.ctl" and Section 4.1.10, "HWDmodification"). This library is used for the OPC configuration of the HWD elements of the framework which are connected to an OPC Server. The following functions are available:

- compGetHWDConfig: get the front-end config of a HWD element.
- compDeleteHWDConfig: delete the front-end config of a HWD element.
- compSetHWDConfig: set the front-end config of a HWD element.
- compGetOPCID: get the internal PVSS id of the compass OPC element which is handling the OPC Server connection.

5.11 **fwk_libMsgConfig.ctl**

This library is used by the AI and AO_F panels for the conversion configurations, this configuration is only supported for the elements which are connected to an OPC Server. The following functions are available:

- compCreateMsgConv: create a conversion config of a dpElement.
- compDeleteMsgConv: delete a conversion config of a dpElement.

5.12 **fwk_libPvRange.ctl**

This library handles the pv_range configuration, this configuration is only supported for CAEN_HV_ch CAEN_HV_ch_SP, AI and AO_F elements. The available functions are:

- compCreatePvRange: create the pv_range config of a dpElement.
- compGetPvRange: query the pv_range config of a dpElement.
- compDeletePvRange: delete the pv_range config of a dpElement.

5.13 **fwk_libRuntime.ctl**

This library is used by the run-time and reference panels (Section 4.2, "Run-time panels", Section 4.3, "Reference panels") to handle: Background color convention (Figure 53) for leaf element. and Background color convention (Figure 53) for node element.

- compConnectColourToStatus: does a dpConnect to the dp element and its attributes for the color setting of the element.
- a set of working functions called by the dpConnect.

5.14 **fwk_libSmoothingConfig.ctl**

This library handles the smoothing configuration, this configuration is only supported for AI and AO_F elements which are connected to an OPC Server. The available functions are:

- compSaveSmoothingConfig: create the smooth config of a dpElement.
- compDeleteSmoothingConfig: delete the smooth config of a dpElement.
- compGetSmoothingConfig: query the smooth config of a dpElement.

5.15 **fwk_libUserAccess.ctl**

This library implements all the user access checking. The available function is:

- compActionAllowed⁵⁷: check if the user is allowed to execute a particular action.

5.16 **fwk_libFanTray.ctl**

This library implements function to create, modify and delete a FAN_TRAY element.

⁵⁷ in the release v1.4, the user access is not implemented, this function always returns TRUE.

5.17 **as.ctl**

This file is a library inherited from PVSS, this is the alert screen library which handles the behavior of the PVSS AlertScreen panel. It was modified to load automatically a pre-defined configuration when the AlertScreen panel is opened.

5.18 **fwk_libSyncMan.ctl**

This library handles the synchronization with the script: dcsMain-HVMain.ctl (Section 6.1, "dcsMain-HVMain.ctl"), dcsMainStatus.ctl (Section 6.4, "dcsMainStatus.ctl") and dcsHV-FTAlarm.ctl (Section 6.3, "dcsHV-FTAlarm.ctl") when the elements are added or deleted and when the alarms on the elements are added or removed. The available functions are:

- compSyncAllMan: to synchronize all the scripts.
- compSyncAllMan: to synchronize one of the scripts.

The library functions wait 1 second before checking that the synchronization is executed, an error message is return back if the synchronization was not successfully done.

6 **Framework Scripts**

Three scripts are executed in the PRODUCTION PVSS system. They are located in the common/scripts folder. These scripts can also be started in the OFFLINE system. Some function of the Section 5, "Framework Libs" are used inside the scripts.

6.1 **dcsMain-HVMain.ctl**

This script is handling all the actions described in Section 3.4, "action on the framework elements": Table 2, Table 3, Table 4, Table 5 and Table 6. The PVSS manager number of this script is 1 and is executed by the PVSS01ctrl executable file.

6.2 **dcsDrvMain.ctl**

This script implements the behavior of the OPC connection described in Section 3.7, "OPC". The PVSS manager number of this script is 3 and is executed by the PVSS03ctrl executable file.

6.3 **dcsHV-FTAlarm.ctl**

This script implements the behavior of the CAEN HV channels described in Section •, "setTrip: set the trip time value to all the CAEN HV channels." and the behavior of the CAN WIENER FAN_TRAY described in Section 3.6, "FAN TRAY behavior". The PVSS manager number of this script is 2 and is executed by the PVSS02ctrl executable file.

6.4 dcsMainStatus.ctf

This script is handling the state calculation of the elements (Section 3.3, "Alarms and states"). The PVSS manager number of this script is 4 and is executed by the PVSS02ctrl executable file.