



Keck Adaptive Optics Note 677

Keck Next Generation Adaptive Optics Alarm Sub System and Service

Jimmy Johnson, Doug Morrison, Erik Johansson
Draft: June 23, 2009

1	Introduction.....	4
2	Related Documents	4
3	Document Conventions.....	4
4	Overview.....	4
4.1	Features	4
4.2	Concepts.....	4
4.2.1	Alarm	5
4.2.2	Condition.....	5
4.2.3	Source	5
4.2.4	Severity	5
4.2.5	Alarm Area.....	5
4.2.6	Alarm States.....	5
4.2.7	Alarm Instance	5
5	Architecture.....	6
5.1	Alarm Configuration.....	7
5.1.1	Alarm Table	8
5.1.2	Alarm Categories	9
5.2	Alarm Service	10
5.2.1	Set Alarm	10
5.2.2	Clear Alarm.....	10
5.2.3	QueryCategories	10
5.2.4	QueryAreas	10
5.2.5	QueryConditionNames	10
5.2.6	QuerySourceConditions.....	10
5.2.7	QueryAttributes.....	10
5.3	API Usage	11
5.3.1	Components	11
5.3.2	Alarm Manager	11
5.3.3	Clients (UI)	11
5.4	Alarm Manager	11
5.4.1	EnableConditionByArea	13
5.4.2	EnableConditionBySource.....	14
5.4.3	DisableConditionByArea.....	14
5.4.4	DisableConditionBySource.....	14
5.4.5	CreateSubscription.....	14
5.4.6	AckCondition.....	14
5.4.7	Refresh	15
5.4.8	CancelRefresh	15
5.4.9	OnAlarm	15
5.5	Alarm Summary Display	15
5.5.1	Alarm Icons.....	17
5.5.2	Alarm fields	17
5.5.3	Repeated alarms	18
5.5.4	View additional features	18
5.5.5	Repeated alarms	20

5.6 Alarm Logging..... 21

5.6.1 Alarm Table 21

1 Introduction

This document describes the alarm subsystem that is part of the Keck Common Services Framework (KCSF) and the associated alarm service.

2 Related Documents

OPC A&E Specification – there is a new UA version but only available to OPC members.

3 Document Conventions

When using code fragments as examples, we will use the non-proportionally spaced Courier New font:

```
class foobar extends foo {  
    int foo1;  
    double foo2;  
}
```

4 Overview

Alarms are reports of component failure (software or hardware) within the system or other abnormal behavior. Alarms occur asynchronously, some alarm conditions may clear themselves and others may require operator intervention.

4.1 Features

- Alarms are defined based on condition rules for device properties.
- Alarms are reported to a central alarm handler that logs all alarms and tracks the state of all alarms.
- Some alarms require explicit acknowledgment from an operator, others may simply disappear when the condition causing the alarm is cleared.
- Alarms may be organized hierarchically, especially for display to an operator.
- It is possible to define actions that can be automatically started on the occurrence of specific alarms.
- Different annunciation methods can be applied to different alarms
- Alarms can be associated with displays, help pages, diagnostic pages or more..
- Applications may request to be notified of alarms.

4.2 Concepts

An alarm can be considered an event expect that is has an associated condition, one which is deemed to be abnormal and requiring special attention. A condition is associated with a Source which is a typically KCSF component.

4.2.1 Alarm

An *alarm* is an abnormal *condition*, which requires special attention outside the control application. Each alarm instance is associated with a *source* (or *owner*) device. For example, the device Ao.Cameras.TT (the *source*) may have an active HighAlarm *condition* or the device AO.MCS.M1. (the *source*) may have an active LowAlarm *condition*. Each alarm instance is uniquely identified by the combination of *source*, and *condition*.

4.2.2 Condition

An alarm *condition* is a named abnormal state for a *source*. Examples of alarm *conditions* are:

LowAlarm, HighAlarm, HighHighAlarm, DeviationAlarm, etc.

4.2.3 Source

All alarms are owned by named items in the domain. Any component can be an alarm source. A source may be the owner of several alarm conditions. The term *owner* is also sometimes used in place of term *source*.

4.2.4 Severity

This specification defines *severity* as an indication of the urgency of the alarm. This value may range from 1 to 1,000, with 1 being the lowest urgency and 1,000 being the highest.

4.2.5 Alarm Area

This specification defines an *Alarm Area* as a collection of alarm groups. A client of the Alarm Manager may request a list of all active and unacknowledged alarms for a specified area.

4.2.6 Alarm States

An alarm occurrence can be in one of four states:

- Inactive/Acknowledged (not included in active alarm list)
- Active/Unacknowledged
- Active/Acknowledged
- Inactive/Unacknowledged

4.2.7 Alarm Instance

It is expected that the Alarm manger will maintain the following data (or similar) per alarm instance.

- Source String Identifier for the alarm owner
- Condition String Identifier for the alarm condition
- Group String Identifier for the native alarm group
- Annunciation String Identifier for annunciation method
- TimeStamp Time Date and time for last state change
- Severity Integer Severity level

- Active Boolean True while alarm is Set
- Acked Boolean True is alarm has been Acknowledged
- Enabled Boolean True if the Alarm is Enabled
- HelpText String Alarm Help Text
- AckRequired Boolean Acknowledged Required Flag
- ChangeTime Time Time Alarm last changed state
- Comment String Any user-defined alarm comment
- RepeatCount Integer Number of set/clears since last ack

5 Architecture

The KCSF Alarm System consists of three key subsystems:

- Alarm service
 - The Alarm Service is available to every container. Applications use the service to set and clear alarms. The Alarm Service for each container synchronizes alarm state information with the system-wide Alarm Manager on initialization and sends all alarm state changes to the Alarm Manager during normal run-time operation.
- Alarm Manager
 - The Alarm Manager maintains a list of all active and unacknowledged alarms in the domain and makes this list available for user interface controls and displays. The Alarm Manager also logs all alarm state changes to a historical database.
- Alarm Summary Display
 - The Alarm Summary Display provides the user with a summary view of all active and unacknowledged alarms from the alarm area of interest. The user can view detailed information about each alarm from this display. This display may also be used to acknowledged alarms.

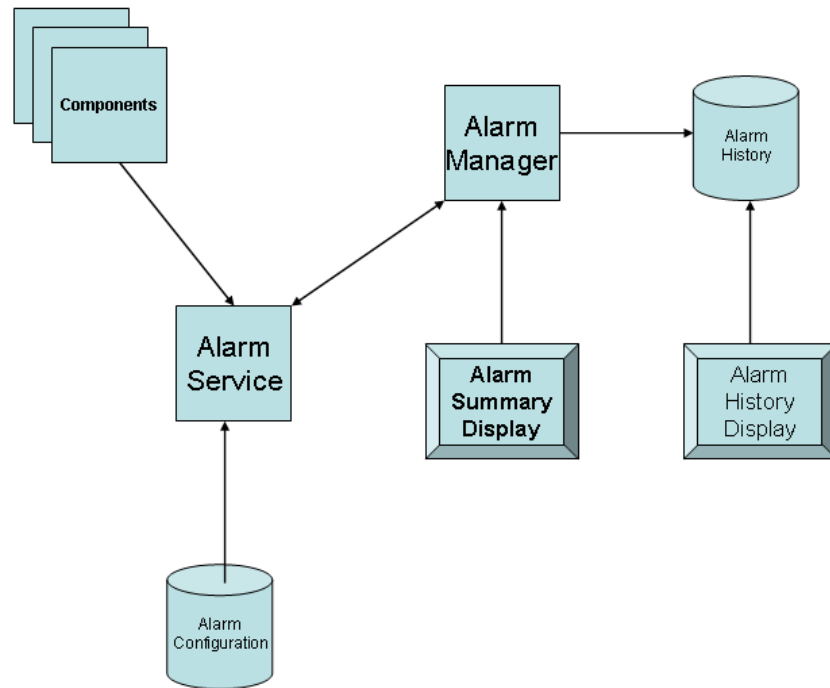
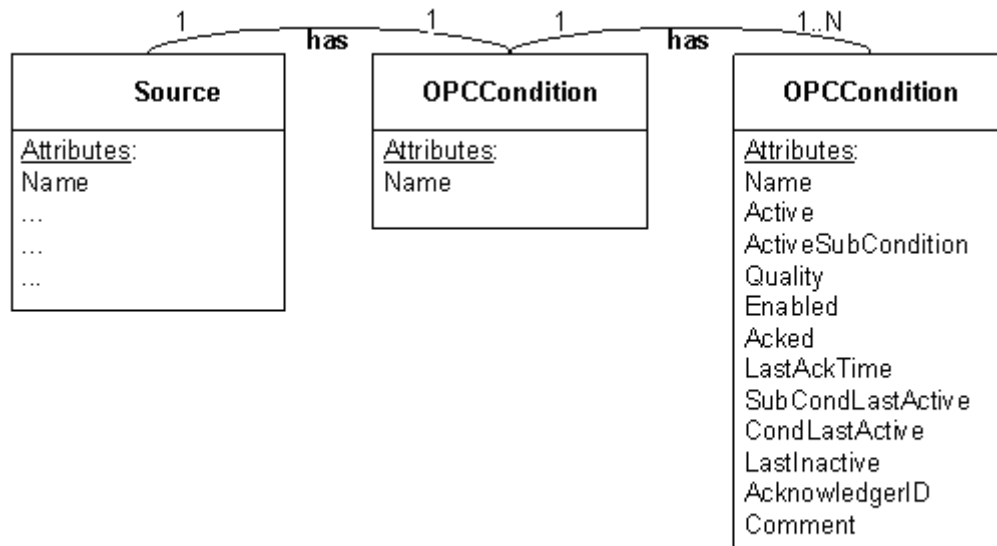


Figure 1 Alarm System Architecture

Note the archiving of alarm information to the alarm history database could be accomplished through a separate dedicated component. It may also be possible that the system could support the Alarm Summary Displays without the need for the Alarm Manager.

5.1 Alarm Configuration

In the framework alarms are generated by sources which are typically components. A source has an associated alarm category and a category has a number of preset conditions.



It is expected that there will be an Alarm Setup Display used to configure all alarms in the system.

It is expected that there will be at least two alarm tables. A simple table maintaining the list of all sources and another maintaining the list of alarm categories.

5.1.1 Alarm Table

The primary alarm filed is expected to be similar to the following:

Field Name	Description
Name	Source Name
Alarm category	Associated Alarm category
Area	Area may be used to associate an alarm with a particular section of the observatory. The area associated with each alarm can be viewed at the Alarm Summary Display. Alarms can also be sorted and filtered by area at the Alarm Summary Display. A blank section name can be selected if no section is to be associated with this alarm record (default value).
Description	
Alarm help	Instance specific help
Priority	The priority field is used to define the priority or importance of each alarm. Severity may range from 0 to 1,000 with 0 being the lowest severity and 1,000 being the highest. Alarm severity determines the alarm text color on the Alarm Summary Display. Alarms may be sorted or filtered by severity on the Alarm Summary Display. For compatibility reasons, severity values of 0 through 3 are mapped by the system to higher severity values as follows: 0 = 100, 1 = 300, 2 = 600, and 3 = 950.

Ack required	This field determines whether or not operator acknowledgement of this alarm is required. If AckRequired is set to “Yes” then this alarm must be acknowledged by the operator (and cleared by the application) before it will be removed from the Alarm Summary Display. If AckRequired is set to “No” then this alarm is considered an informational alarm, which does not require operator acknowledgement before it is removed from the Alarm Summary Display (this alarm will be removed from the Alarm Summary Display when cleared by the application).
Disabled	The Disabled field is used to disabled individual alarms. Alarms which have the Disabled field set to “Yes” cannot be set by the application, will not appear in the Alarm Summary Display and will not appear in the alarm history log.
Logging disabled	If the LogDisabled field is set to “yes” then changes to the state of this alarm (alarm set, clear or acknowledge) are not logged to the alarm history file. When this field is “No” (which is the default) alarm state changes are logged to the alarm history file.
Associated Display	One associated display can be defined for each alarm record. When an associated display is defined for an alarm then the operator can navigate directly to the associated display from the Alarm Summary Display whenever the alarm is active or unacknowledged.

5.1.2 Alarm Categories

Each alarm record is associated with one Alarm Category. The alarm category determines the possible alarms for a source. For example:

Alarm records with a category of “Process_Inputs” might include the following alarms:

- Bad input device
- High alarm limit
- Low alarm limit

Alarm records with a category of “System_Status” might include the following alarms:

- Disk space is low
- Virtual memory usage is too high
- CPU usage is high
- Fan has failed

Field Name	Description
Name	Category Name
Conditions	Text describing each condition that can exist for this category

5.2 Alarm Service

From a components perspective there are just two APIs:

5.2.1 Set Alarm

This method is used to set an alarm. The service will automatically add the source name.

```
SetAlarm( Category, Condition, AlarmValue);
```

5.2.2 Clear Alarm

This function is used to clear an alarm.

```
ClearAlarm(Category, Condition);
```

In order to offer filtered lists a client made need to retrieve information as to what categories, conditions and areas have been defined. The following methods allow clients to get this information.

5.2.3 QueryCategories

The QueryCategories method gives clients a means of finding out the specific categories of alarms supported by a given server. This method would typically be invoked prior to specifying a filter.

5.2.4 QueryAreas

The QueryAreas method gives clients a means of finding out the specific areas supported by a given server. This method would typically be invoked prior to specifying a filter.

5.2.5 QueryConditionNames

The QueryConditionNames method gives clients a means of finding out the specific condition names which the alarm server supports for the specified category. This method would typically be invoked prior to specifying an alarm filter. Condition names are server specific.

5.2.6 QuerySourceConditions

The QuerySourceConditions method gives clients a means of finding out the specific condition names associated with the specified source.

5.2.7 QueryAttributes

This may or may not be needed. Most likely we just define a set of standard attributes and will not need user defined ones. Using the Categories returned by the QueryCategories method, client application can invoke the QueryAttributes method to get information about the vendor-specific attributes the server can provide as part of a notification for an alarm within the specified category. Simple servers may not support *any* vendor-specific attributes for some or even all Categories.

5.3 API Usage

5.3.1 Components

API	Location	Usage
SetAlarm	Service	Calls
ClearAlarm	Service	Calls

5.3.2 Alarm Manager

API	Location	Usage
EnableConditionByArea	Alarm Manager	Implements
EnableConditionBySource	Alarm Manager	Implements
DisableConditionByArea	Alarm Manager	Implements
DisableConditionBySource	Alarm Manager	Implements
CreateSubscription	Alarm Manager	Implements
AckCondition	Alarm Manager	Implements
Refresh	Alarm Manager	Implements
CancelRefresh	Alarm Manager	Implements
OnAlarm	Alarm Manager	Calls

5.3.3 Clients (UI)

API	Location	Usage
QueryCategories	Service	Calls
QueryConditionNames	Service	Calls
QuerySourceConditions	Service	Calls
EnableConditionByArea	Alarm Manager	Calls
EnableConditionBySource	Alarm Manager	Calls
DisableConditionByArea	Alarm Manager	Calls
DisableConditionBySource	Alarm Manager	Calls
CreateSubscription	Alarm Manager	Calls
AckCondition	Alarm Manager	Calls
Refresh	Alarm Manager	Calls
CancelRefresh	Alarm Manager	Calls
OnAlarm	Client	Implements (Callback)

5.4 Alarm Manager

Alarm events can occur as the result of a set alarm, clear alarm or refresh operation. In all cases the data structure that is sent is the same. A proposed data structure modeled after the OPC ONALARMSTRUCT is described below.

Member	Description
szSource	The source of the alarm notification.
ftTime	Time of the occurrence - for conditions, time that the condition transitioned into the new state or sub-condition. For example, if the alarm notification is for acknowledgment of a condition, this would be the time that the condition became acknowledged.
szMessage	Notification message describing the alarm.
dwEventType	OPC_SIMPLE_EVENT, OPC_CONDITION_EVENT, or OPC_TRACKING_EVENT for Simple, Condition-Related, or Tracking events, respectively.
dwEventCategory	Standard and Vendor-specific category codes.
dwSeverity	Alarm severity (0..1000).
	The following items are present only for Condition-Related Events (see dwEventType)
szConditionName	The name of the condition related to this alarm notification.
wChangeMask	Indicates to the client which properties of the condition have changed, to have caused the server to send the alarm notification. It may have one or more of the following values: OPC_CHANGE_ACTIVE_STATE OPC_CHANGE_ACK_STATE OPC_CHANGE_ENABLE_STATE OPC_CHANGE_QUALITY OPC_CHANGE_SEVERITY OPC_CHANGE_SUBCONDITION OPC_CHANGE_MESSAGE OPC_CHANGE_ATTRIBUTE If the alarm notification is the result of a Refresh, these bits are to be ignored. For a “new alarm”, OPC_CHANGE_ACTIVE_STATE is the only bit which will always be set. Other values are server specific. (A “new alarm” is any alarm resulting from the related condition leaving the Inactive and Acknowledged state.)
wNewState	A WORD bit mask of three bits specifying the new state of the condition: OPC_CONDITION_ACTIVE, OPC_CONDITION_ENABLED, OPC_CONDITION_ACKED.
wQuality	Quality associated with the condition state.

bAckRequired	This flag indicates that the related condition requires acknowledgment of this alarm. The determination of those alarms which require acknowledgment is server specific. For example, transition into a LimitAlarm condition would likely require an acknowledgment, while the alarm notification of the resulting acknowledgment would likely not require an acknowledgment.
ftActiveTime	Time that the condition became active (for single-state conditions), or the time of the transition into the current sub-condition (for multi-state conditions). This time is used by the client when acknowledging the condition
dwCookie	Server defined cookie associated with the alarm notification. This value is used by the client when acknowledging the condition. This value is opaque to the client.
	The following is used only for Tracking Events and for Condition-Related Events which are acknowledgment notifications (see dwEventType).
szActorID	For tracking events, this is the actor ID for the event notification. For condition-related events, this is the AcknowledgerID when OPC_CONDITION_ACKED is set in wNewState. If the AcknowledgerID is a NULL string, the event was automatically acknowledged by the server. For other events, the value is a pointer to a NULL string.

The Alarm Manager maintains the alarm information it receives from the framework via the SetAlarm and ClearAlarm APIs. It will propagate these alarms to the client based on the client provided filter (optional implementation) and based on whether or not an area or category is enabled/disabled.

The Alarm Manager responses to alarm acknowledgements and will remove the alarms as they are cleared and acknowledged.

5.4.1 EnableConditionByArea

Places all conditions for all sources within the specified process areas into the enabled state. Therefore, the server will now generate condition-related alarms for these conditions.

The effect of this method is global within the scope of the alarm manager. Therefore, if the manager is supporting multiple clients, the conditions are enabled for all clients, and they will begin receiving the associated condition-related alarms.

* Will need to decide if this needs to propagate to each alarm service or if it is just applied to the Alarm Manager.

5.4.2 EnableConditionBySource

Places all conditions for the specified sources into the enabled state. Therefore, the alarm manager will now generate condition-related alarms for these conditions.

The effect of this method is global within the scope of the alarm manager. Therefore, if the alarm manager is supporting multiple clients, the conditions are enabled for all clients, and they will begin receiving the associated condition-related alarms.

* Will need to decide if this needs to propagate to each alarm service or if it is just applied to the Alarm Manager.

5.4.3 DisableConditionByArea

Places all conditions for all sources within the specified process areas into the disabled state. Therefore, the alarm manager will now cease generating condition-related alarms for these conditions.

The effect of this method is global within the scope of the alarm manager. Therefore, if the alarm manager is supporting multiple clients, the conditions are disabled for all clients, and they will stop receiving the associated condition-related alarms.

* Will need to decide if this needs to propagate to each alarm service or if it is just applied to the Alarm Manager.

5.4.4 DisableConditionBySource

Places all conditions for the specified sources into the disabled state. Therefore, the alarm manager will no longer generate condition-related alarms for these conditions.

The effect of this method is global within the scope of the alarm manager. Therefore, if the alarm manager is supporting multiple clients, the conditions are disabled for all clients, and they will stop receiving the associated condition-related alarms.

* Will need to decide if this needs to propagate to each alarm service or if it is just applied to the Alarm Manager.

5.4.5 CreateSubscription

This allows a client (for example, UI or logging service) to connect to the Alarm Manager and to start receiving alarms. Alarms are sent to the client(s) via an OnAlarm callback.

5.4.6 AckCondition

The client uses the AckCondition method to acknowledge one or more conditions in the Alarm Manager. The client receives alarms notifications from conditions via the OnAlarm callback. This AckCondition method specifically acknowledges the condition becoming active. One or more conditions belong to a specific source – the source of the

alarm notification. For each condition-related alarm notification, the corresponding Source, Condition Name, Active Time and Cookie is received by the client as part of the OnAlarm callback parameters.

5.4.7 Refresh

Force a refresh for all active conditions and inactive, unacknowledged conditions whose alarms notifications match the filter of the subscription.

Clients will often need to get the current condition information from the alarm manager, particularly at client startup, for things such as a current alarm summary. The Alarm Manager supports this requirement by resending the most recent alarm notifications which satisfy the filter in the alarm subscription and which are related to active and/or unacknowledged conditions. The client can then derive the current condition status from the “refreshed” alarm notifications.

5.4.8 CancelRefresh

Cancels a refresh in progress for the alarm subscription.

If a refresh is in progress, the alarm manager will send one final callback with the last refresh flag set and the number of alarms equal to zero.

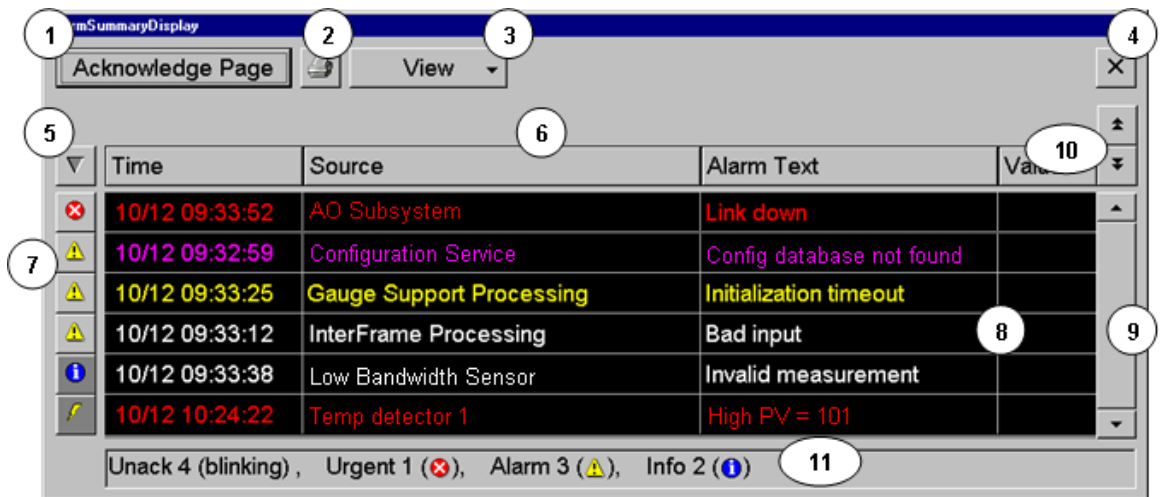
5.4.9 OnAlarm

The Alarm Manager invokes the OnAlarm method to notify the client of alarms which satisfy the filter criteria for the particular alarm subscription.

Note that callbacks can occur for two reasons: alarm notification or refresh. A client can determine the ‘cause’ of a particular callback by examining the bRefresh parameter in the OnAlarm callback.

5.5 Alarm Summary Display

The following shows what the alarm summary display may look like and gives an indication of the functionality that could be made available. The snapshot is reduced in height for brevity. There can be many instances of the Alarm Summary Display running.



As shown it has the following fields:

1. **Acknowledgement button** to acknowledge all unacknowledged alarms on the visible page. After acknowledgement the page is updated and the alarm lines will disappear if the alarm condition was cleared or if the alarm is still active the alarm button (7) will stop blinking and change color from light gray to dark gray. Alarms can be also be individually acknowledged.
2. **Print button** to print all filtered alarm lines in the current sorting order with the fields shown in the columns title buttons (6) and the fields selected for printing.
3. **View selection button** to select which features are shown in the summary display. More details below.
4. **Exit button** to hide the Alarm Summary Display.
5. **Standard sorting order.** By clicking on this button the alarm lines are sorted into the following order:
 - a. Unacknowledged events that require acknowledgement (new events)
 - b. Unacknowledged active alarms that require acknowledgement (new alarms)
 - c. Unacknowledged passive alarms that require acknowledgement (waiting ack)
 - d. Acknowledged active alarms (old alarms)
 - e. Acknowledged events (old events)
 - f. Active alarms that do not require acknowledgement (info alarms)
 - g. Events that do not require acknowledgement (info events)

The new events and alarms are sorted by the severity (from highest to lowest). All other lines are sorted by time (from newest to oldest). After the sorting first page of alarms are shown.
6. **Column title buttons.** By clicking one of these buttons the alarm lines are sorted by the selected column. On the first click the lines are sorted in ascending order and on the next click the lines are sorted in descending order. Alarms that have the same key value are sorted by time (from newest to oldest)

7. **Alarm line buttons.** By clicking one of these buttons the Alarm Acknowledgement dialog window is invoked for the selected alarm line. The Alarm Acknowledgement dialog window is explained in more details in following chapters. The Alarm line button indicates the alarm status in the following way:
 - a. Blinking icon = alarm is unacknowledged
 - b. Light gray button = alarm is unacknowledged
 - c. Dark gray button = alarm is acknowledged
 - d. Colored icon = alarm is active
 - e. Gray icon = alarm has returned to normal state
8. **Alarm list.** By clicking a field in the alarm list selects the alarm line for potential detail pane view and sets the field as a candidate for filtering.
9. **Alarm list scroller.** The handle of the scroller indicates the portion of alarm lines visible in the alarm summary display. The little arrows at the end of the scroller allows the content to be moved up or down by a single line. Dragging the handle or clicking on a target position can move the content by more than a single line.
10. **Page up and page down buttons.** By clicking these buttons the content of alarm list (8) is scrolled up or down by a visible page. The number of alarms on a page depends on the Alarm Summary Display height. If shift key is held down then the first or last page is shown instead of the previous or next page.
11. **Alarm summary counts.** This text indicates the current counts and explains the meaning of the active icon symbols:
 - a. Number of alarm lines waiting acknowledgement (unacknowledged alarms)
 - b. Number of events
 - c. Number of urgent alarms
 - d. Number of other alarms
 - e. Number of information messages

5.5.1 Alarm Icons

The following symbols are used for different alarm types.

Event	Event which requires the operator attention	
Urgent	An alarm which requires immediate actions	
Alarm	An alarm condition which requires acknowledgement	
Information	An alarm condition which does not require acknowledgement	

5.5.2 Alarm fields

An engineer selects the alarm fields shown in the Alarm Summary Display. The selection includes the relative positions and allocated widths. In this example case the following fields are shown:

- **Time** in selected time format. The field contains selected subfields and separators for year in two or four digits, month name or number, day of the month or year, weekday, hours, minutes, seconds, and AM/PM indicator. The subfields can be in any order. The maximum accuracy of the second's subfield is in milliseconds.

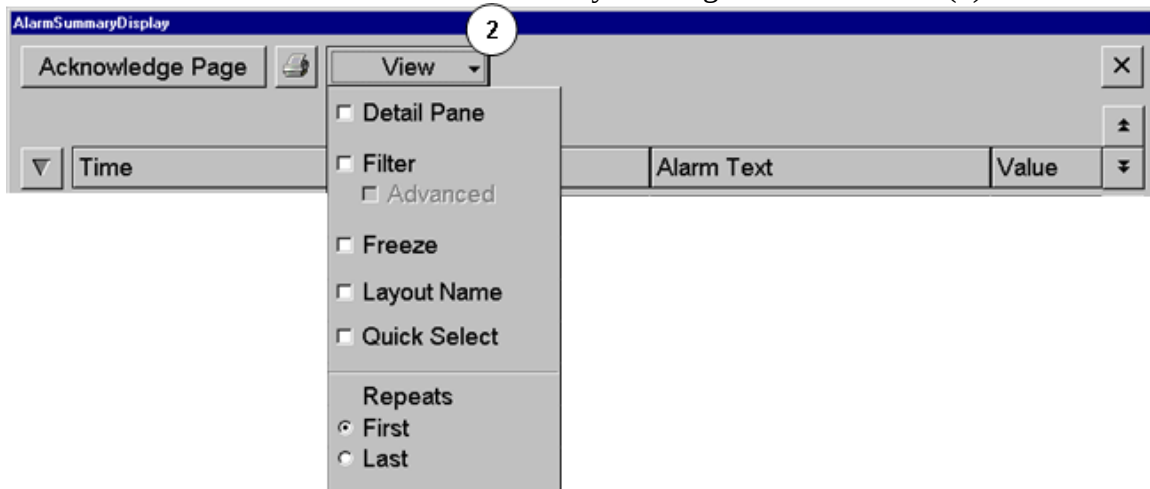
- **Source** is the owner of an alarm. In a typical case the source is identified with an instrumentation tag. It can be also a process device, scanner, sensor, camera, actuator, or a subsystem.
- **Alarm text** is a definition of the condition that caused the alarm or a message describing the event. This is translated to local language and it can contain the values of the other fields as embedded elements.
- **Value** is an example of optional fields. It indicates the variable value used in the condition calculation at the time of first alarm occurrence.

5.5.3 Repeated alarms

If an acknowledgement is required and the alarm condition goes to on and off several times, then the time for the first instance and last instance are shown together with the count during that period.

5.5.4 View additional features

The user can see a list of additional features by clicking the View Button (2).



The additional features are selected by clicking the feature names or the associated check boxes and radio buttons.

5.5.4.1 Detail pane

If the detail pane is activated then space is allocated for a detailed text box. A line is selected for the detail pane by clicking on any field on the line. The detailed text box is updated to show all known fields for the selected alarm line – independently from which fields are selected to the list view.

5.5.4.2 Filter

If the filter is activated then a Filter Button and Filter Text field will be shown.

The image shows a software window titled "AlarmSummaryDisplay". At the top, there is a toolbar with "Acknowledge Page" (with a printer icon), a "View" dropdown menu, and a close button (X). Below the toolbar, there is a "Filter" button and a yellow text input field. Below the input field, there is a table with four columns: "Time", "Source", "Alarm Text", and "Value". Each column has a small downward arrow icon next to it. The "Time" column has a small upward arrow icon next to it. The "Value" column has a small downward arrow icon next to it.

Filtering can occur on Area, Group (if we implement it), Source, and Severity or on a custom filter such as a text string in the alarm message. Once the selection is accepted, only the alarm lines satisfying the criteria are shown

The image shows a dialog box titled "Alarm line filter selection". It has two yellow text input fields. The first field is labeled "Alarm Text" and contains a tilde (~) character. The second field is labeled "Sensor Disabled" and contains a tilde (~) character. To the right of the second field is a "Find" button. Below the input fields, there are five buttons: "Area", "Group", "Source", "Severity", and "Other". To the right of these buttons is a "Clear" button. At the bottom right, there is a "Cancel" button.

The number of unacknowledged alarms and urgent alarms in the Alarm Summary Counts (11) are calculated from the unfiltered alarm list. If the Quick Select feature was activated, then the number of filtered alarms is shown on right of the Quick Select buttons.

An option will be to allow filters to be saved and reused

The image shows a software window titled "AlarmSummaryDisplay". At the top, there is a toolbar with "Acknowledge Page" (with a printer icon), a "View" dropdown menu, and a close button (X). Below the toolbar, there is a "Filter" button and a yellow text input field containing "Source ~ trend". Below the input field, there is a table with four columns: "Time", "Source", "Alarm Text", and "Value". Each column has a small downward arrow icon next to it. The "Time" column has a small upward arrow icon next to it. The "Value" column has a small downward arrow icon next to it. Below the table, there is a row of buttons: "Store Filter", "1", "2", "3", "4", "5", "6", "7", "8", "9", "Clear Filter", and "Filtered 95". At the bottom, there is a status bar with the text "Unack 22 (blinking), Urgent 1 (X), Alarm 6 (Y), Info 500 (I)".

5.5.4.3 Freeze Button

If the freeze checkbox is selected then the Freeze button (G) is shown. In a alarm rush situation the operator can freeze the alarm summary display update for 30 seconds to evaluate the situation. During the freeze, the alarm summary display should look different from when it is operating normally. In the mockup it is shown with a blue border and yellow background. A blue bar in the freeze timer indicates the progression of freeze time. The timer allows the freeze to be repeated (G) or interrupted (H).

AlarmSummaryDisplay

Acknowledge Page View [STOP] [STOP]

Time	Source	Alarm Text	Value
Oct 13 18:12:41	GSP InterFrame Processing	Bad Now Value	
Oct 13 18:12:41	GSP InterFrame Processing	Bad Standardization	
Oct 14 08:12:04	GSP InterFrame Processing	Sensor Disabled	
Oct 13 17:50:09	GSP InterFrame Processing	Bad Background	
Oct 14 08:04:46	Gauge Support Processing	Can't declare event	

Store Filter 1 2 3 4 5 6 7 8 9 Clear Filter Total 522

Unack 22 (blinking), Urgent 1 (X), Alarm 6 (A), Info 500 (i)

5.5.5 Repeated alarms

An alarm instance can occur multiple times in the alarm system. This happens when the alarm requires acknowledgement and is not acknowledged before it occurs again. The user can select how to filter these types of repeated alarms. The choices are:

- First: shows only the first (oldest) occurrence - indicated with “|..” in time field.
- Last: shows only the last (newest) occurrence - indicated with “..|” in time field.

AlarmSummaryDisplay

Acknowledge Page View [Detail Pane] [Filter] [Advanced] [Freeze] [Layout Name] [Quick Select] [Repeats] [First] [Last]

Time	Source	Alarm Text	Severity	Value
10/13 18:12:41 ..	GSP	Bad Now Value	999	
10/13 18:12:41 ..	GSP	Bad Standardization	900	
10/14 08:12:04	GSP	Sensor Disabled	600	
10/13 17:50:09 ..	GSP	Bad Background	600	
10/14 08:04:46	Gau	Can't declare event	300	

Unack 22 (blinking), U (A), Info 500 (i)

5.6 Alarm Logging

All alarms can be logged to a database. It is expected that there will be an Alarm Logging Display that can be used to view all recent alarm state changes and events. This display may be used to query the alarm history files in many ways. The alarms may be logged by the Alarm Manager or there may be a separate system wide component that is responsible for monitoring and logging alarms.

5.6.1 Alarm Table

The following shows a possible schema for the alarm table:

- **TimeStamp:** UT time when the alarm state change or event occurred.
- **Category:** Alarm category for this alarm. Alarm categories are defined by the alarm system configuration and are meant to provide logical alarm groupings.
- **Source:** Owner of the alarm or event.
- **Alarm Text:** Description of the alarm or event.
- **Action:** A classification of the alarm state change or event. Typical actions are Set Alarm, Clear Alarm, or Acknowledge Alarm.
- **Value:** This field may be used to save the value of some key variable associated with the alarm or event. Any value shown represents the value at the time of the alarm or event occurrence.
- **Operator:** May be used to record the operator performing the event or action.
- **Severity:** The severity (or priority) of the alarm or event. This may range from 1 to 1,000. Low values indicate a low urgency and high value represent a higher urgency.