	Paul Scherrer Institut
Projekt / *Project*	**SwissFEL**
Titel / *Title*	**SwissFEL alarmhandler policy**
Dokument Nummer / *Document Identification*	FEL-CK85-001-1
Autor / *Author*	Christoph Kittel
Mitautor(en) / *Co-Author(s)*	Didier Voulot, Nicole Hiller, Andreas Lüdeke
Zusammenfassung / *Summary*	This document defines the basic rules for the configuration of alarms for the *Alarmhandler* in SwissFEL . This is necessary in order to make it a useful tool for commissioning and accelerator operation. It is intended to be a reference document for equipment specialists who need to provide alarm configuration files to the SwissFEL project and operation.
Datum / *Date*	December 7, 2018

Änderungen / *Version history*

Revision	Datum / *Date*	Autor / *Author*	Änderung / *Modification*
1.0	21.04.2017	Christoph Kittel	First version to publish
1.1	13.03.2018	Christoph Kittel	Minor changes
1.2	07.12.2018	Christoph Kittel	ForcePV, Alarmcountfilter, Git, Logs

Contents

1 Introduction

The alarmhandler is a key operational tool to allow a quick and productive response to problems which emerge during operation. Therefore it is not the main task of the alarmhandler to provide general diagnostics of the machine and all of its devices. To allow an efficient operation of the SwissFEL the alarmhandler application has to show alarms according to specific rules. These rules are laid down in this policy document.

2 Basic rules

The basic rules for the definition of alarms in the alarmhandler are:

2.1 An alarm shall only be shown if it requires an action of an operator.

An action of an operator is necessary when there is either a potential or severe problem for the safe operation of the machine.

Therefore an alarm shall **not** be raised if:

- the device is not needed for the current mode of operation or the machine is in shutdown mode (realisation still to be defined),
- the device is unused or spare (see 2.1.1),
- the device is operating as intended (see 2.1.2),
- a failing device has no consequences for the safe operation of the machine (e.g. a Beam Position Monitor, if it is not used in a feedback).

2.1.1 Unused devices should not generate unnecessary alarms.

As long as alarms from unused and spare devices do not represent a potential hazard to operation they shall be disabled. This should ideally be done by integrating the necessary logic directly on the EPICS level. In some cases a ForcePV can be used as well:

```
$FORCEPV forcePVName forceMask <forceValue> <resetValue>
```

Examples:

- A low photon intensity should only raise an alarm during photon delivery, while the user leave the photon shutter open and the machine is not declared as in downtime. This is implemented via CALC records directly in the alarm PV
- A large set of alarms of an unused RF station can be canceled by using a ForcePV:
  CHANNEL S10CB01 S10CB01-RHLA-JOBVSY:SF-VM-LEAK-OK - - - - -
  $FORCEPV S10CB01-RFOP:ALARMS-ENABLE-MSM C - - - - 0 1

2.1.2 Alarms shall reflect only abnormal situations.

An alarm shall not be raised just to show informations on the expected performance of a device.

Example:

> Inhibiting shortly the beam because a diagnostic beam screen is driving to/out of the measurement position is normal. It does not need to be reported, unless it is stuck in an undefined state.

2.2 Alarms should reflect the root cause.

It is good practice to not show secondary alarms which result from another problem. In order to prevent this, one can implement a logic for a kind of disaster mode, where certain PV's can suppress subsequent alarm groups, as long as other alarms are then clearly displaying the root cause instead.

Example:

- RF devices should rather show the first fault, instead of the full alarm cascade.
- If the master timing stops working, the resulting RF alarms can be suppressed.

2.3 Every alarm or alarm group shall have a concise guidance text.

A short guidance text has to explain concisely the reason for the alarm, how to verify/diagnose the problem independently and most importantly give clear instructions for the operators on how to resolve the problem. The text will be displayed on a dialog-panel, which is created by the alarmhandler. It has to be ensured, that the content is valid and explanatory.

Examples:

```
$GUIDANCE
<text lines>
$END
```

2.4 Every alarm or alarm group shall give a link to an appropriate panel.

Every Alarm shall offer a link to a panel, which shows more information on the error and the current status of the device. If possible, a functionality to treat the alarm cause of that device should be given in that panel.

Example:

> If a magnet alarm is caused by an interlock, the panel should give more details on the reason of why there is an interlock, e.g. high temperature of a magnet, high temperature of power supply.

2.5 Transient alarms need some form of mitigation.

Transient alarms shall be configured with a dead-time, hysteresis or other measures. This is necessary in order to prevent a constant need for acknowledgements, a potential flooding of the alarm log data base and simply annoyances.

One can make use of the HYST field in EPICS PV's or the alarmcountfilter in the alarmhandler configuration files, which is stated as "$ALARMCOUNTFILTER inputCount inputSeconds". It raises the alarm if either the alarm is activated more often than specified by inputcount (during inputseconds), or by just being active for longer than specified by the inputseconds. Putting zero for inputcount does not seem to work as expected unfortunately.

Examples:

- Slowly changing PV's, like a temperature oscillating around an alarm threshold, should make use of a hysteresis (HYST field in EPICS PV's).
- Fast changing PV's, like BLM's and CPU loads, should make use of a dead time, e.g.:
  $ALARMCOUNTFILTER 50 5

2.6 The hierarchy should be flat and logically structured

It is strongly advised to keep the alarm tree hierarchy flat while keeping everything grouped in a logical and comprehensive way. Five levels of subgroups below the main group should be considered as a limit. It should be also avoided to put more than 50 alarms into a single group. This is important for having a usable interface despite the deficits of the Motif GUI.

Alarm groups with very few alarms/subgroups inside shall be avoided if possible. Instead it can be checked if a mixed group makes sense instead. However, in this case alias names should be considered to provide a visual grouping of the alarms/subgroups.

2.7 Alias for channel names shall still contain PV-names

If an alias is used for a channel name, the PV name shall be always added to the end of that string. This is to ensure that each alarm can be easily related to the corresponding PV, since only the PV name is stored in the alarm log files. The syntax shall be the following: $ALIAS aliasName (PVName).

Example:

```
CHANNEL VACUUM SIN-EVVS-A0010:PLC_GENERAL
$ALIAS Vakuum Sektor 1 SINEG01 (SIN-EVVS-A0010:PLC_GENERAL)
```

2.8 Groups shall have unique names.

In opmod-log files forced mask for groups are stored only together with their respective group-name, without any alias. Therefore alarm groups have to have short, but **unique** names and should optimally reflect their position in the hierarchy. So it would be good to include the abbreviations of their parent-group(s) in their name, to represent at least partly the tree structure and thus allow finding a group in the hierarchy. In order to increase the readability of the hierarchy, one should use the $ALIAS option and give short, descriptive alias group names.

Example:

```
GROUP NULL SF
$ALIAS SwissFEL
    GROUP SF SIN
    $ALIAS Injector (SIN)
```

```
GROUP SIN SIN_LLRF
$ALIAS Low Level RF (LLRF))
    GROUP SIN_LLRF SIN_LLRF_ILK
    $ALIAS Interlocks (ILK)
```

2.9 The alarmhandler language is German.

In particular a mix of English and German in the same line shall be avoided. Although common technical terms can be left untranslated. A check for a proper spelling is very much appreciated. German "Umlaute", as well as tabulators cannot be displayed by the alarmhandler and shall not be used. Empty spaces shall be avoided at the beginning and at the end of a line.

Examples:

- "Girder" does not need to be translated into German.
- Tabulators are wrongly displayed as “%”.

2.10 How to receive automatically Email and SMS alarm notifications

Alarms can be configured in a way, that they execute a command upon a change of the alarm severity. A command to send an Email or a SMS is provided. The service can be used to stay informed about devices. Valid severity change values are: UP_INVALID, UP_MAJOR, UP_MINOR, UP_ANY, DOWN_MAJOR, DOWN_MINOR, DOWN_NO_ALARM, DOWN_ANY, UP_ALARM. Email is recommended for the standard cases, due to the costs of SMS. Furthermore SMS are truncated to 160 characters.

> However, be warned about the potential dangers of this service! Caution has to be exercised, due to the costs of 0.25 CHF per SMS and potential flooding/spamming issues for SMS and Email. Additional safeguards or safer alternatives (e.g. browsing the alarm logs and ELOG) should be considered.

Example:

- Sending an Email:
  $SEVRCOMMAND UP_ALARM message -s "alarm subject" -m “alarm message” -a emailAddress
- Sending a SMS:
  $SEVRCOMMAND UP_ALARM message -m "alarm message" -p Phonenumber

3 Severity levels

The SEVR alarm field in an EPICS database record specifies the severity of an alarm state. Currently the Alarmhandler in SwissFEL defines the following alarm severities.

3.1 Normal

Default state for a functioning device.

3.2 Minor

A minor alarm represents a **true potential problem** for the operation of the machine or the machine safety. It needs to be investigated and treated by the operators. By default it raises a sound and needs acknowledgement!

3.3 Major

A major alarm represents a **severe problem** for the operation of the machine or machine safety. It needs to be diagnosed and treated immediately by the operators. By default it raises a sound and needs acknowledgement!

3.4 Invalid

An alarm with the status invalid means for example that there are time-outs, old values or just invalid values seen by the PV.

3.5 Error

The alarm severity Error does not exist in EPICS, but it will be called as such in the Alarmhandler if there is no connection to a PV. This can be either through a spelling error in the definition or due to a real connection problem, e.g. by the IOC or the network.

4 Alarm mask settings

Associated with each Alarm Channel are two five bit masks (default and current). The current mask can be changed by force commands or by force process variables. The default mask is defined in the alarm configuration file. A reset command forces all associated masks to return to the default values.

The default mask shall be always set to (- - - - -), except for some special cases. Instead the current mask shall be used for normal cases. The meaning of each bit in the mask value is briefly shown below.

Following below is a brief description of the individual alarm flags.

1. Cancel Alarm

   " C" means cancel. It will not show the alarm, nor will the alarm be logged.

2. Enable/Disable Alarm

   " D" means alarm disabled, but is still logged in the data base

3. NoAck

   " A" means alarm acknowledgement is not required.

4. NoAck Transient Alarms

   " T" means acknowledgement of transient alarms is not required.

5. No Log Alarms

   " L" means no alarm logging .

5 Git-Project

The configuration files for the alarmhandler can be found in the git project “sf_alh_config” (git@git.psi.ch:alarmhandler_config/sf_alh_config.git). Configuration changes should be done by using the “app_config” tool, provided by controls, from the swissfel network. Pushing via app_config creates automatically merge requests which need to be approved by the operations team. Once approved, the changes will be delivered automatically to the operational consoles.

In general it is advisable to generate the configuration files for bigger sets of alarms in a scripted way, in order to be able to quickly adapt to changes without too much manual commitment.

Example:

- Pull the project folder from the git repository:
  ```
  $ app_config alarmhandler pull
  ```
- Push your locally committed changes to the git repository and generate a merge request:
  ```
  $ app_config alarmhandler push
  ```

6 Further information

Further explanations can be looked up in the Alarm Handler User’s Guide version 1.2.35, which can be found here:
`http://www.aps.anl.gov/epics/EpicsDocumentation/ExtensionsManuals/AlarmHandler/alhUserGuide-1.2.35/ALHUserGuide.html`

Past alarms can be looked up in the log file. An easy way to do this is browsing it via the web interface: `http://gfa-operation.intranet.psi.ch/alh/index.php`