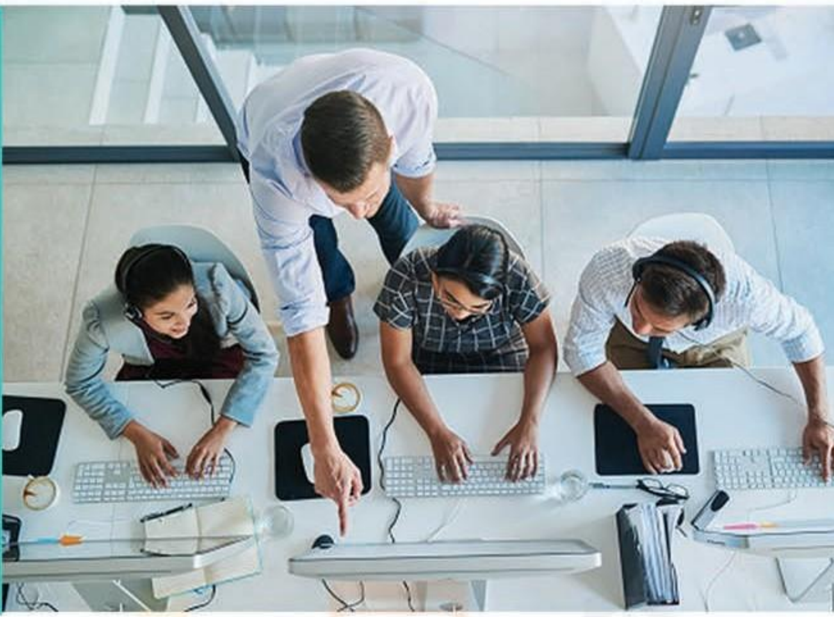
Skill AP
APSSDC

e Skill AP

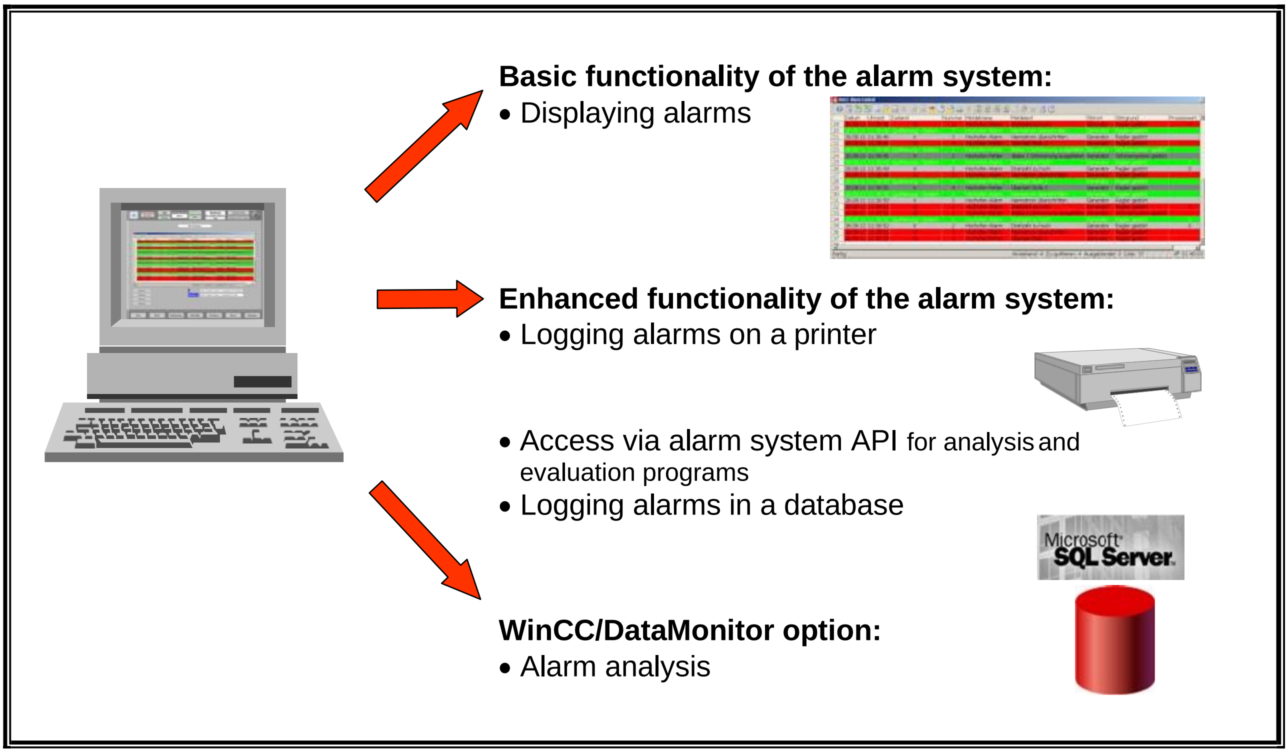
Learn Anytime Anywhere

Andhra Pradesh State Skill Development Corporation



# Alarms

# 7. Alarm system



**Basic functionality of the alarm system:**

- Displaying alarms

**Enhanced functionality of the alarm system:**

- Logging alarms on a printer
- Access via alarm system API for analysis and evaluation programs
- Logging alarms in a database

**WinCC/DataMonitor option:**

- Alarm analysis

In this section participants will learn about and practice using the structure of the alarm system and the alarm logs as well as the configuration of the alarm view and logging.

Participants will get to know the basic functions and potential uses of the alarm system. They will gain an overview of the expanded functionality of the alarm system.

**Training aims:**
Participants will be able to set up alarm classes, configure/display alarms and set up/display alarm logs.

## 7.1. Overview

### General information

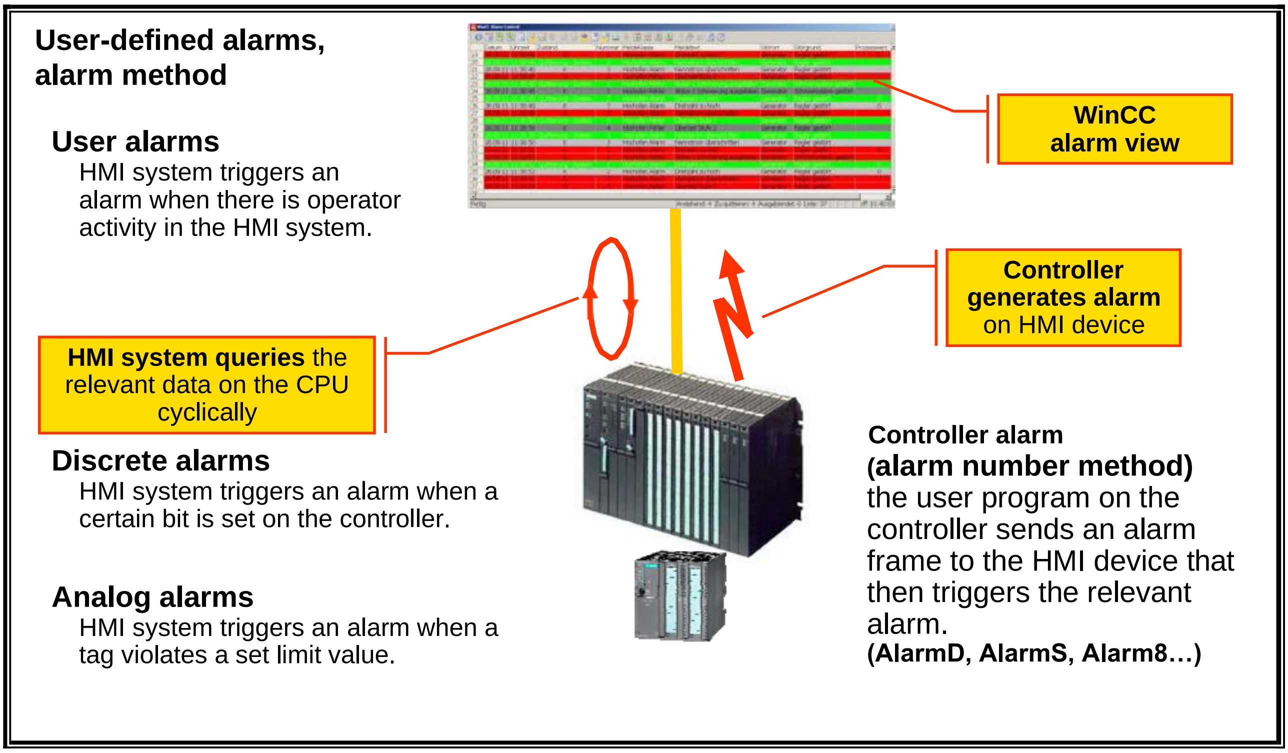
With the alarm system, events from the controller or from the monitoring function in WinCC (operating states, faults, user input etc.) are displayed in the form of alarms, logged if necessary and reported and acknowledgements by the operator are accepted. To allow this, alarms need to be configured in alarm classes. To allow the history to be followed as well, the alarms are stored on the hard disk in a long-term historical alarm list on the local computer. The alarm system of WinCC Professional is based on the directions of DIN 19235.

The alarm system of WinCC Professional provides discrete, analog, user and controller alarms (alarm number method / alarms in the correct chronological order) with:

- Alarm acknowledgements, even to automation systems
- 256 alarm classes
- 17 priorities (0-16)
- Display with row-oriented alarms and graphics
- Freely selectable views of the alarm list and log
- Locking/unlocking and selecting alarms
- Alarm comment, alarm info
- Opening via API (application programming interface) for analysis and evaluation programs
- Triggering actions
- Configurable: 150,000 alarms
- Continuous alarm load without loss (single user/server) 10/sec
- Historical alarm list, in RT no system limit (depends on hard disk)

### 7.1.1. Alarm method

The alarm system processes various alarm methods of the controller and the HMI device.



**User-defined alarms, alarm method**

**User alarms**
HMI system triggers an alarm when there is operator activity in the HMI system.

**HMI system queries** the relevant data on the CPU cyclically

**Discrete alarms**
HMI system triggers an alarm when a certain bit is set on the controller.

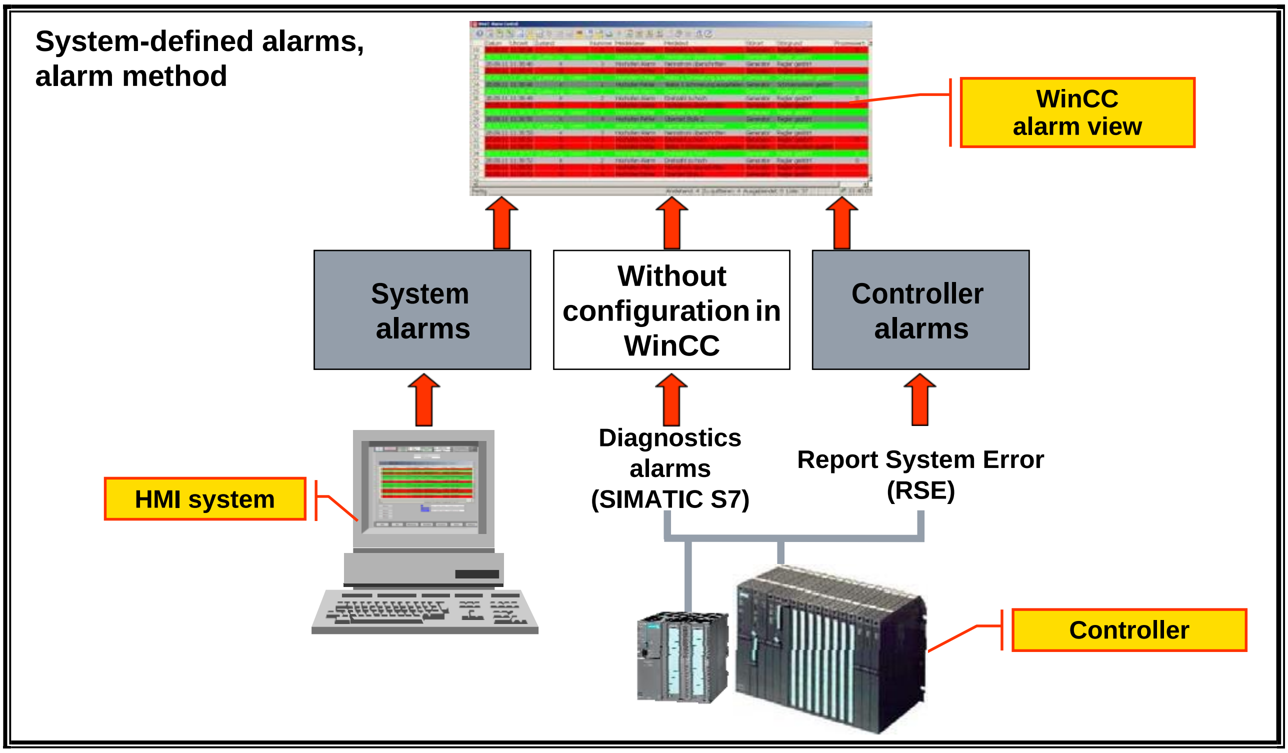
**Analog alarms**
HMI system triggers an alarm when a tag violates a set limit value.

**WinCC alarm view**

**Controller generates alarm** on HMI device

**Controller alarm**
**(alarm number method)**
the user program on the controller sends an alarm frame to the HMI device that then triggers the relevant alarm.
**(AlarmD, AlarmS, Alarm8…)**

The alarm methods can be divided into system-defined alarms and user-defined alarms. User-defined alarms are used to monitor the plant.



**System-defined alarms, alarm method**

**WinCC alarm view**

**System alarms**

**Without configuration in WinCC**

**Controller alarms**

**HMI system**

**Diagnostics alarms (SIMATIC S7)**

**Report System Error (RSE)**

**Controller**

System-defined alarms are used to monitor the HMI device or the controller.

**Configuration instructions for:**

**Discrete or analog alarms**

- WinCC requests the alarm tags (alarm bits or analog values) at 500 ms intervals

+ Simple to configure

+ "Central acknowledgment" via acknowledgment tags

- Must be configured new for each HMI system (time-consuming changes)

+ No CPU program necessary

**"Controller alarms"**

+ Alarm frames sent only with events

+ More accurate time stamp for alarms (approx. length of CPU cycle, e.g. 50 ms)

- Additional user program necessary

- More memory used on the CPU

+ Central configuration can be used for several HMI systems (for WinCC Professional, WinCC Advanced…)

+ Can be changed centrally

+ S7 blocks integrated in the operating system

The alarm method identifies the type of information that triggers an alarm

- Bit change (discrete alarm)
- Value change / limit value violation (analog alarm)
- Operator action (user alarm)
- Programmed event on the controller (controller alarm)

and what generates the alarm

- HMI device
- Controller

This results in advantages ☺ and/or disadvantages ☹.

**Discrete and analog alarm methods**

☺ Simple to configure

☹ Alarms are given the time stamp of the HMI system (time of detection)

☹ No common database for PG and HMI system (alarm texts configured on the HMI device separate from and independent of the controller)

☹ High bus load (HMI system queries the data on the controller constantly every 500 ms)

☹ Alarms can be lost if the event "comes in" and then "goes out" again within the configured update interval

☺ Little additional programming in the controller program required
(discrete alarm method: set/reset bit; analog alarm method: nothing required)

**User alarms**

User alarms indicate operator actions during runtime. If necessary, user alarms can also be called within scripts in runtime for user-defined applications.

☺ Simple to configure

☺ No bus load because the alarms are triggered by operator actions or in a script on the HMI system.

**Controller alarms (alarm number method, alarms in correct chronological order)**

☺ Low bus load (controller signals HMI device when necessary)

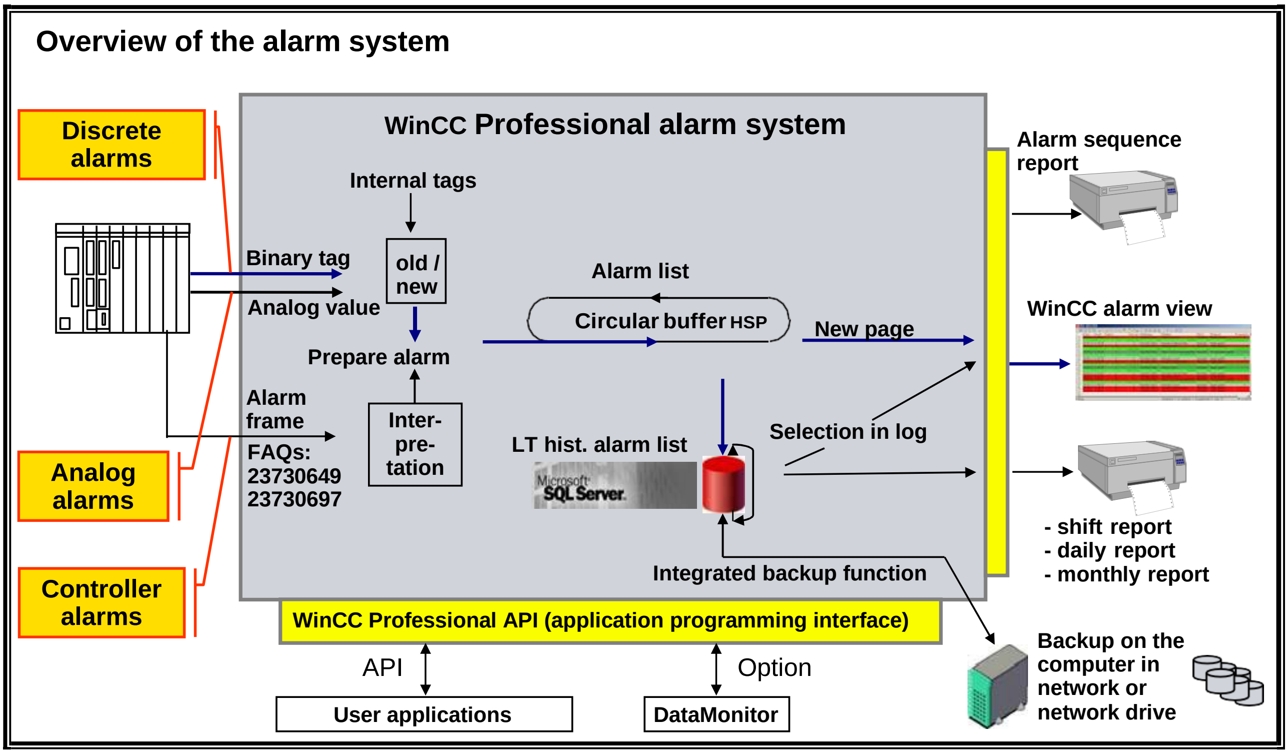
☺ Alarm texts need to be configured separately in the STEP 7 project

☺ Alarms are assigned controller time stamps (alarms in chronological order)

☺ No alarms are lost

☹ Programming of the triggering alarm events in the controller program required (system function handling)

## 7.1.2. Alarm functionality



Overview of the alarm system

The alarm system of an HMI system handles the following tasks:

**Display**

Configured alarms are displayed on the HMI system depending on events or states occurring in the plant or process.

### Reporting

The alarms are output to a printer. They can be output in the alarm sequence report or within a shift report.

### Logging

The alarms are saved in a file in the Microsoft SQL Server database for further processing and evaluation.

### Alarm log backup

To back up the alarm logs (database files) using the integrated backup function of WinCC Professional, computers in the network or network drives can be used.

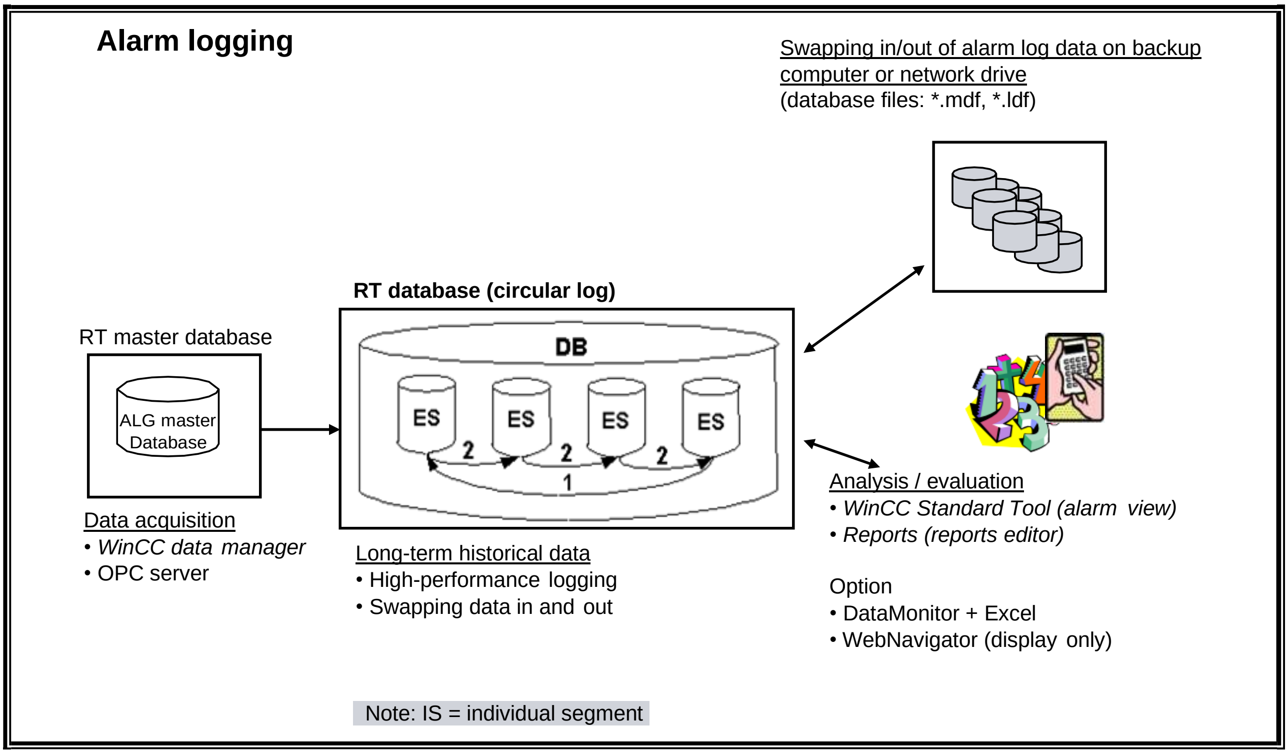
### Analysis

The WinCC/DataMonitor option allows the analysis of logged WinCC alarms.

### WinCC API functions (application programming interface)

Via this interface, user applications can access systems including the alarm system, for example to trigger an alarm in a C script using the "MSRTCreateMsgInstancePlus" function. The functions available there can, however, also be used in C scripts in the WinCC project, such as: "PWRTSilentLogin" for a login with a mouse click. This function will be shown later in the course.

## 7.1.3. Alarm log


Alarm logging
Swapping in/out of alarm log data on backup computer or network drive
(database files: *.mdf, *.ldf)
RT database (circular log)
DB
ES
ES
ES
ES
2
2
2
1
RT master database
ALG master Database
Analysis / evaluation
• WinCC Standard Tool (alarm view)
• Reports (reports editor)
Option
• DataMonitor + Excel
• WebNavigator (display only)
Data acquisition
• WinCC data manager
• OPC server
Long-term historical data
• High-performance logging
• Swapping data in and out
Note: IS = individual segment

### Logging

To log alarms, WinCC uses a circular log of configurable size that can be configured with or without a backup. Log files are always stored in the relevant project locally on the computer. The WinCC alarm log consists of multiple individual segments. You can configure both the size/time range of the alarm log and the size/time range of the individual segments in WinCC.

Example: The alarm log records all alarms that occur within one week (1), each individual segment records the alarms that occur within one day (2). Both conditions are evaluated by the system. If either of the two criteria (time or size) is exceeded, the following happens:
Criterion for alarm log (DB) is exceeded (1)-> the oldest alarms (i.e. the oldest individual segment) are deleted.
Criterion for individual segment exceeded (2)-> a new individual segment (IS) is created.

A new individual segment is also created if alarm data is configured online (when downloading changes online).

WinCC/DataMonitor option

For visualization and evaluation, WinCC/DataMonitor provides a range of tools for viewing (view only) and online analysis, for example using Excel, that can be used via the Internet and that support all the common security mechanisms such as login/password, firewalls, encryption, etc.

OPC

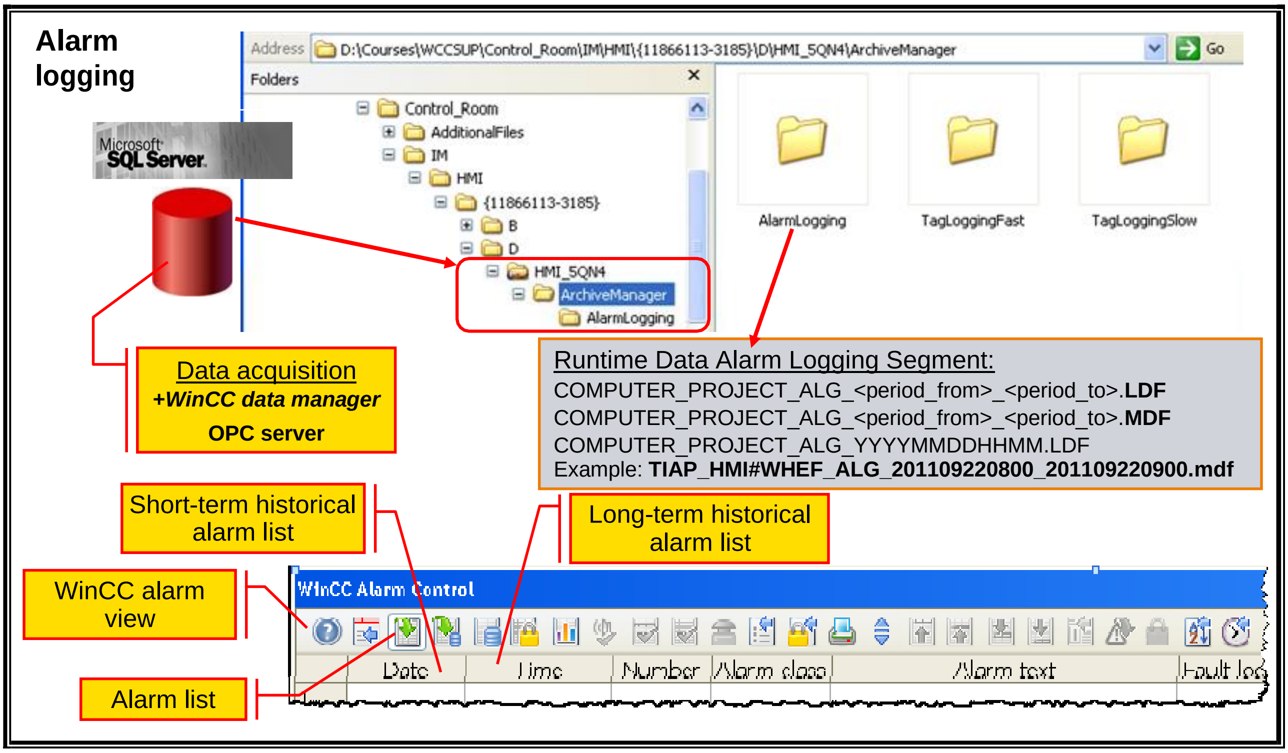
To access WinCC data via OPC, OPC HDA (Historical Data Access) and OPC A&E (Alarms & Events) can be used. The OPC HDA and OPC A&E servers allow access to historical data of the WinCC logging system or for forwarding / acknowledging alarms.

WinCC OLE DB

With WinCC OLE-DB, it is possible to directly access log data saved in the MS SQL Server database by WinCC. the WinCC OLE-DB Provider ships with WinCC Professional.

Note

Without associated values and comments, an alarm requires approximately 172 bytes of memory (RT database), see FAQ 16619980.



Runtime Data Alarm Logging Segment:
COMPUTER_PROJECT_ALG_<period_from>_<period_to>.**LDF**
COMPUTER_PROJECT_ALG_<period_from>_<period_to>.**MDF**
COMPUTER_PROJECT_ALG_YYYYMMDDHHMM.LDF
Example: **TIAP_HMI#WHEF_ALG_201109220800_201109220900.mdf**

**Alarm logging in the runtime database**

The runtime data is distributed over the master database and several Runtime databases. The master database manages the runtime databases and references the individual runtime databases. The master database is created in the project directory. The name of the master database is made up of the project name with "RT" appended to it. The runtime databases contain the logged data for a particular period and are located in a subdirectory "ArchiveManager/Alarm Logging" of the project directory.

The name of the runtime databases is made up as follows:

- computer name
- project name (assigned by the system)
- ID for the type of data:
    - ALG = Alarm Logging -> alarms,
    - TLG_F = Tag Logging Fast – measured value logs, fast detection,
    - TLG_S = Tag Logging Slow – measured value logs, slow detection
    - date/time (year, month, day, hours, minutes.

Example of the name of a log segment:
TIAP_HMI#WHEF_ALG_201109220800_201109220900.mdf.

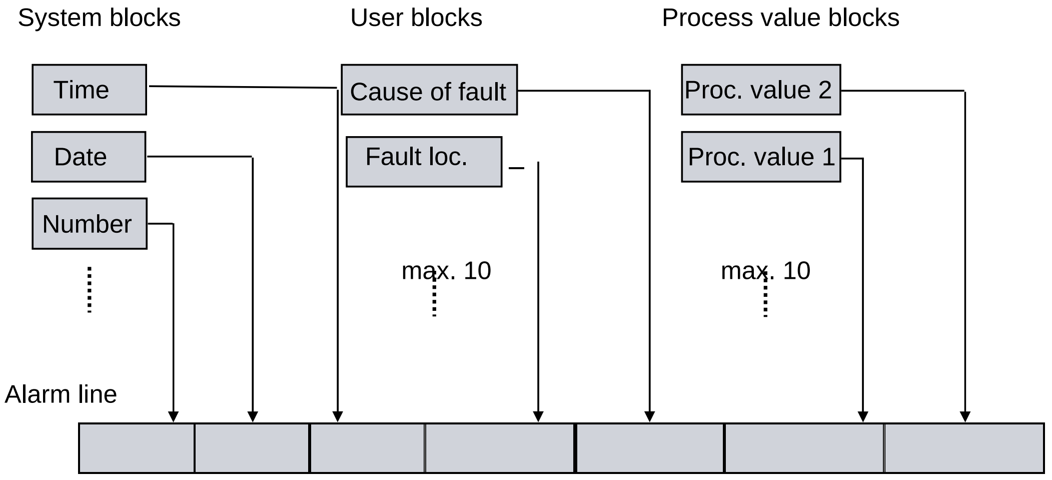
The file name of this log segment indicates that this segment contains the alarms for a specific period and that no more data will be written to this log. If the file only contains the time for the segment change, this file will continue to be filled with data (until the next segment change).

**Note on the buttons:**

The "Alarm list" button shows the currently pending alarms in the WinCC alarm view (WinCC Alarm Control) in runtime mode. The "Historical alarm list (short-term)" and "Historical alarm list (long-term)" buttons display the same log data. In the long-term historical alarm list, the online comments can also be stored.

### 7.1.4. Components and properties of alarms

**Alarm list structure:**



**General**

Alarms are shown in the form of tables with the WinCC Professional alarm view (WinCC Alarm Control). An alarm consists of configured alarm text blocks. The table layout is set during the configuration of the alarm list by selecting the system, user text and parameter blocks.

System blocks

The system blocks output system information, for example date, time, alarm number, alarm class and status

### User text blocks

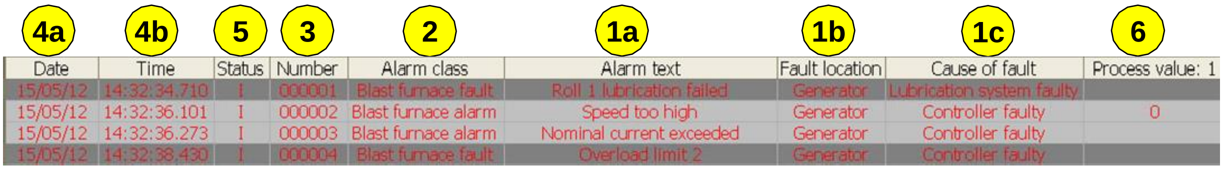
User text blocks contain the alarm text with the description of the cause of a fault and additional texts with information, for example the location to help localize the fault. Up to 10 user text blocks can be used per alarm.

### Parameter blocks

Parameter blocks are used to link the alarms to process values, for example, current fill levels, temperatures or speeds. Up to 10 parameter blocks can be used per alarm.

### Basic alarm list components (blocks), display in Runtime mode:

**1** - Alarm text, max. 10 user text blocks
**2** - Alarm class for alarm properties
**3** - Alarm number
**4** - Time stamp
**5** - Status
**6** - Process value, max. 10 parameter blocks



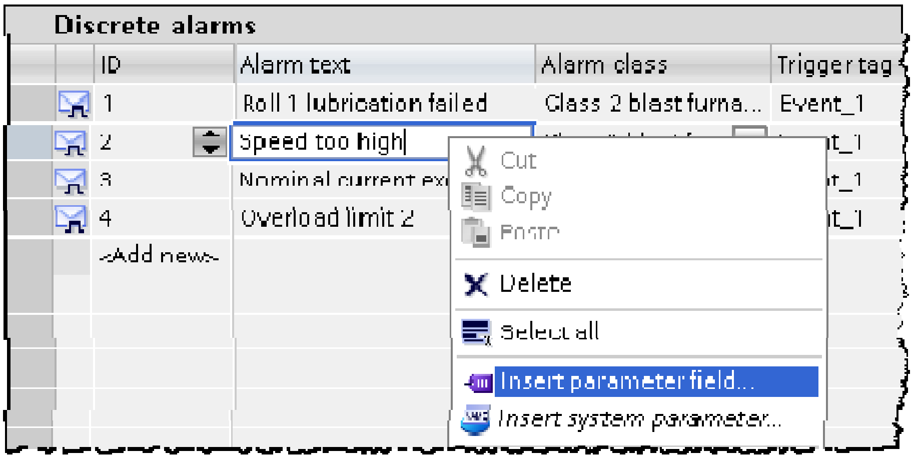
| Date | Time | Status | Number | Alarm class | Alarm text | Fault location | Cause of fault | Process value: 1 |
|---|---|---|---|---|---|---|---|---|
| 15/05/12 | 14:32:34.710 | I | 000001 | Blast furnace fault | Roll 1 lubrication failed | Generator | Lubrication system faulty | |
| 15/05/12 | 14:32:36.101 | I | 000002 | Blast furnace alarm | Speed too high | Generator | Controller faulty | 0 |
| 15/05/12 | 14:32:36.273 | I | 000003 | Blast furnace alarm | Nominal current exceeded | Generator | Controller faulty | |
| 15/05/12 | 14:32:38.430 | I | 000004 | Blast furnace fault | Overload limit 2 | Generator | Controller faulty | |

### Alarm properties

- Acknowledgment
- Color scheme
- Alarm status (I - Incoming, O - Outgoing, IO - without acknowledgment incoming/outgoing, A - acknowledged)
- Archiving

### Alarm text

The alarm text, for example, contains the description of the cause of a fault. Process values can also be embedded in a user text block using the "Insert parameter field" function.



Discrete alarms

| ID | Alarm text | Alarm class | Trigger tag |
|---|---|---|---|
| 1 | Roll 1 lubrication failed | Class 2 blast furna... | Event_1 |
| 2 | Speed too high | | t_1 |
| 3 | Nominal current ex | | t_1 |
| 4 | Overload limit 2 | | t_1 |
| <Add new> | | | |

This makes it possible, for example to display the current speed list value in the alarm text. As an alternative, tag values can also be displayed using parameter blocks.

### Alarm class

The alarm class defines whether or not the alarm needs to be acknowledged. The appearance of the alarm when displayed in runtime mode can also be modified. The alarm class also determines whether the corresponding alarm is logged.

⇒ Max. 256 alarm classes can be configured

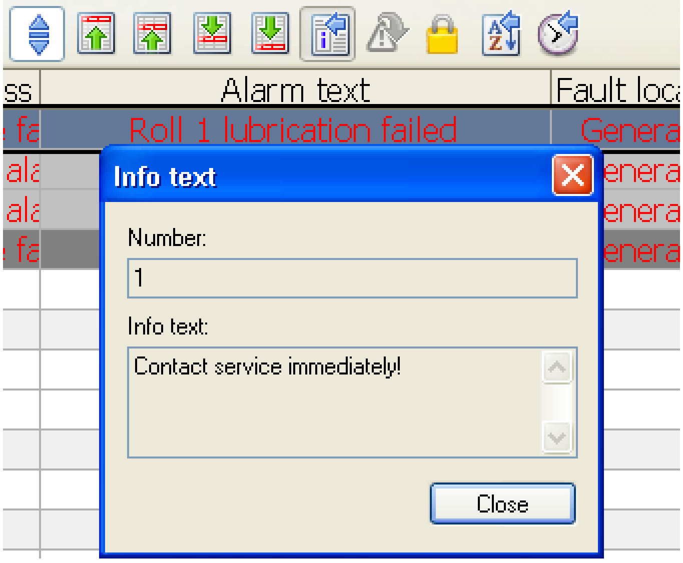
### Alarm number

The alarm number is used to internally manage an alarm. Each alarm number is unique within the following types of alarms:

⇒Discrete alarms, analog alarms, HMI system alarms, < alarms from the controller within a CPU.

### Information text

The info text can contain additional information on an alarm. This information needs to be entered when configuring the individual alarm. The info text is displayed in a separate window on the HMI device when the operator clicks the "Infotext dialog" button.



| ss | Alarm text | Fault loca |
|---|---|---|
| fa | Roll 1 lubrication failed | Genera |
| ala | | enera |
| ala | | enera |
| fa | | enera |

Info text

Number: 1

Info text: Contact service immediately!

Using the "Comment dialog" button, comments can also be created for individual alarms in the long-term historical alarm list.

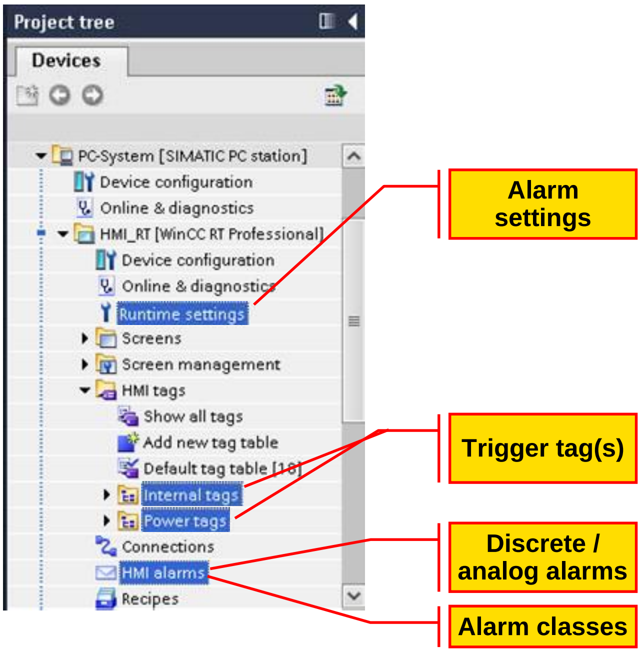
### Printing an alarm sequence report

If the alarm sequence report has been created and is enabled in the runtime settings in "Start sequence of WinCC Runtime", each status change of an alarm in runtime is output continuously on a printer. No operator action is required for this in Runtime.

☞ If a line printer is used here (e.g. dot matrix printer), every change to an alarm status of an alarm selected for reporting is printed out immediately.
When using page printers (normally all printer drivers for Windows printers), the printout is not made until an alarm page is full and the printer can print a complete page.

## 7.2. Configuring discrete alarms

### 7.2.1. Steps in configuring an alarm



Alarm settings

Trigger tag(s)

Discrete / analog alarms

Alarm classes

1. **Configure alarm blocks**
   → system blocks
   → user blocks
   → process value blocks

1. Set **alarm classes**
   or configure your own alarm classes

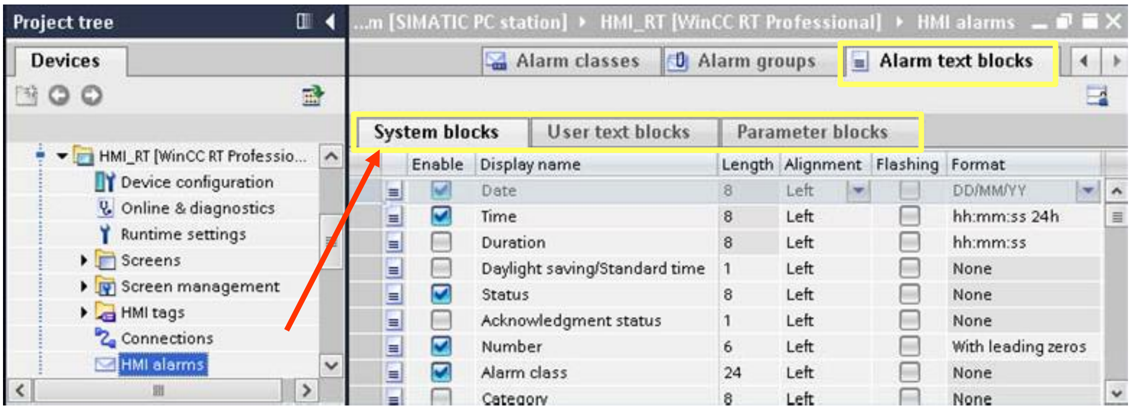
2. Specify **alarm settings**
   → display suppression
   → backtransfer of alarms

3. Configure **trigger tag(s)**
   → bit string for discrete alarms
   → threshold tags for Analog alarms

4. Configure **discrete alarms**
   Configure **analog alarms**

### 7.2.2. Message blocks

**1. HMI alarms editor / alarm blocks**



| Enable | Display name | Length | Alignment | Flashing | Format |
|---|---|---|---|---|---|
| ☑ | Date | 8 | Left | □ | DD/MM/YY |
| ☑ | Time | 8 | Left | □ | hh:mm:ss 24h |
| □ | Duration | 8 | Left | □ | hh:mm:ss |
| □ | Daylight saving/Standard time | 1 | Left | □ | None |
| ☑ | Status | 8 | Left | □ | None |
| □ | Acknowledgment status | 1 | Left | □ | None |
| ☑ | Number | 6 | Left | □ | With leading zeros |
| ☑ | Alarm class | 24 | Left | □ | None |
| □ | Category | 8 | Left | □ | None |

- **Alarm blocks**
  **→ system blocks**
  **→ user text blocks**
  **→ parameter blocks**

## Message blocks

The display of the alarms is made up of alarm blocks. Each alarm text block corresponds to one column in the table display of the alarm view.

There are three groups of alarm text blocks:

### System blocks

System blocks are system data, for example date, time, alarm number and status.

### User text blocks

User text blocks contain the alarm text with the description of the cause of a fault and additional texts with information, for example the location to help localize the fault

### Parameter blocks

Parameter blocks are used to link the alarms to process values, for example, current fill levels, temperatures or speeds. Up to 10 parameter blocks can be configured per alarm.

## Configure Message Blocks

The alarm blocks are configured and enabled on the "Alarm blocks" tab in the HMI alarms editor. To display alarm blocks in an alarm view, the required alarm blocks are selected from the enabled alarm blocks in the "Screens" editor when setting the parameters for the alarm view.

### Properties of alarm blocks:

### Text length

The text of a user text block can be up to 255 characters long. With controller alarms: the maximum text length is 32 characters

### Flashing

An alarm that requires acknowledgement or individual alarm blocks can be displayed flashing in runtime mode when necessary. This is possible if the following requirements are met:

The "Flashing ON" property must be activated in the alarm class of the alarm. This property is configured on the "Alarm classes /... blocks" tab.

In the Inspector window of the alarm text blocks with the "Flashing ON" property, the "Flashing" option is shown as being activated under "General" in the "Alarm block" area.

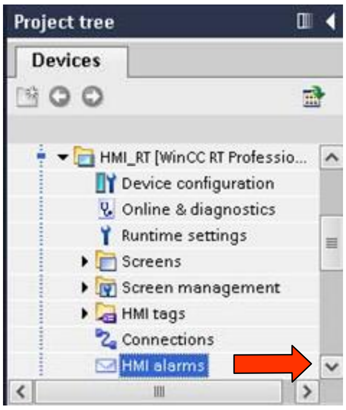
**Alignment**

Using the "Alarm text blocks / ...blocks" tab, you specify whether or not the alarm block is displayed to the right, left or centered in a column of the alarm view.

**Format**

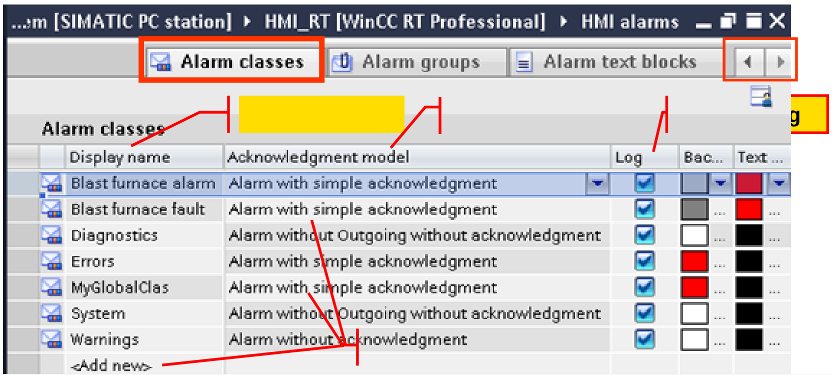
For certain alarm text blocks, it is possible, when necessary, to select different display formats (for example date or time format, leading zeros).

## 7.2.3. Message classes



**Predefined alarm classes**

- **Diagnostics alarms**
  → error / diagnostics alarms of the controller
  → are generated by the connected controller
- **Error** (freely available)
  → for discrete or analog alarms requiring acknowledgment
- **System** (freely available)
  → are generated by the HMI system
- **Warnings** (can be used freely)
  → for discrete/analog alarms not requiring acknowledgment


...m [SIMATIC PC station] ▸ HMI_RT [WinCC RT Professional] ▸ HMI alarms
Alarm classes
Alarm groups
Alarm text blocks
g

Alarm classes

| Display name | Acknowledgment model | Log | Bac... | Text ... |
|---|---|---|---|---|
| Blast furnace alarm | Alarm with simple acknowledgment | ☑ | ▨ | 🟥 |
| Blast furnace fault | Alarm with simple acknowledgment | ☑ | ▨ ... | 🟥 ... |
| Diagnostics | Alarm without Outgoing without acknowledgment | ☑ | ⬜ ... | ⬛ ... |
| Errors | Alarm with simple acknowledgment | ☑ | 🟥 ... | ⬛ ... |
| MyGlobalClas | Alarm with simple acknowledgment | ☑ | 🟥 ... | ⬛ ... |
| System | Alarm without Outgoing without acknowledgment | ☑ | ⬜ ... | ⬛ ... |
| Warnings | Alarm without acknowledgment | ☑ | ⬜ ... | ⬛ ... |
| <Add new> | | | | |

Each alarm must be assigned to an alarm class. Above all, alarm classes decide whether an alarm needs to be acknowledged and the appearance of alarms displayed on the HMI system.

WinCC has predefined alarm classes but also provides the option of configuring user-defined alarm classes.

**Predefined "Errors" and "Warnings" alarm classes**

These have been created for free use.

Faults For discrete and analog alarms that indicate critical or hazardous operating and process states.
Alarms in this class must always be acknowledged.

Warnings For discrete and analog alarms that indicate regular operating statuses, process states and process operations.
Alarms in this class are user information and do not have to be acknowledged.

### Display name

Here, an individual name can be set for each alarm class which can then also be displayed in the alarm view if multiple alarm classes are shown in an alarm view.
Even with the predefined alarm classes, these display names can be individually adapted.

### Alarm acknowledgment

- made by the operator on the HMI system
  To acknowledge alarms, the operator uses the "Single acknowledgement" or "Group acknowledgement" buttons in the WinCC alarm view, a hotkey or self-defined button with a script in C or VBS.
- The connected controller<
  As an alternative, discrete alarms can also be acknowledged by the controller program using a defined data area. An alarm can be acknowledged by the PLC program by setting a particular bit within a tag.

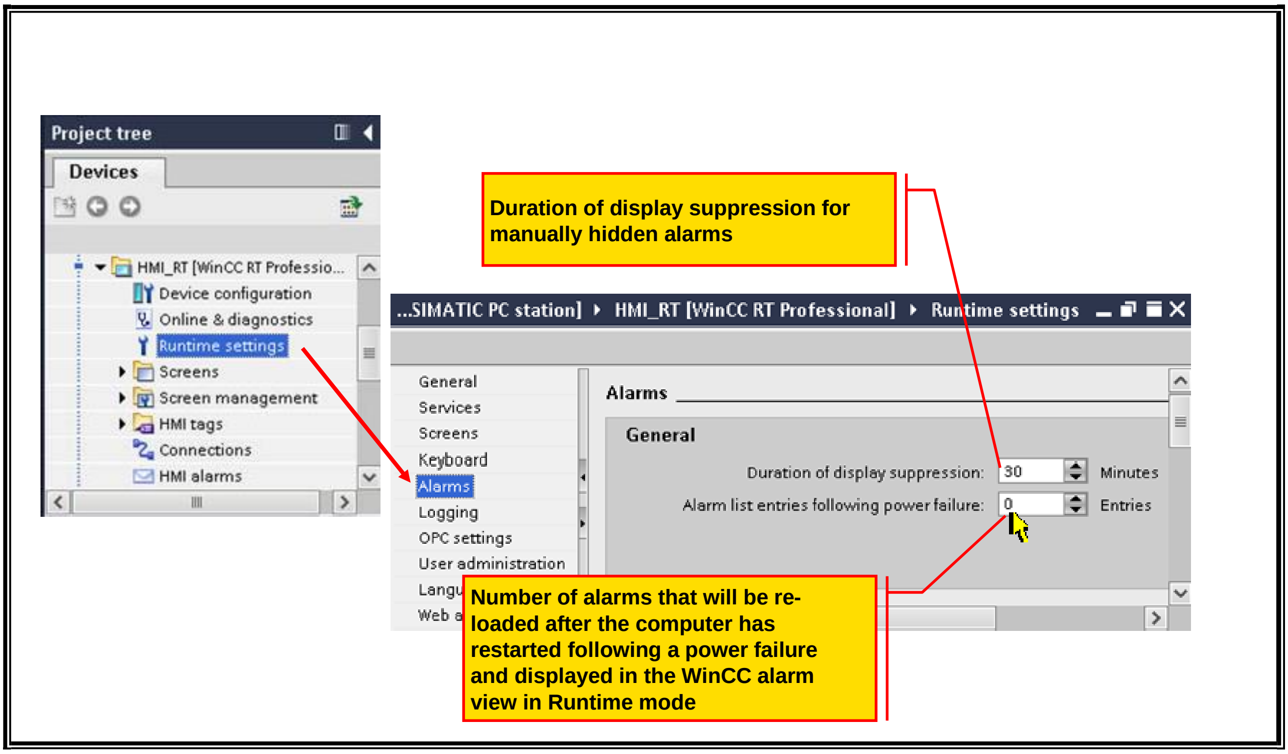
### Log

The alarms are stored in an alarm log if logging is enabled in the alarm class to which they belong. Alarm logs are created by the system. Each alarm event of an alarm is logged, for example the status change of the alarm from "Alarm incoming" to "Alarm acknowledged".

The logged alarms are stored in a circular log consisting of several individual segments. You specify the size of all segments and a single segment in the settings for logging under "Alarm log".

Backups can be configured for the regular backing up of the data.

## 7.2.4. Runtime settings for alarms



Duration of display suppression: 30 Minutes

Alarm list entries following power failure: 0 Entries

Duration of display suppression for manually hidden alarms

Number of alarms that will be re-loaded after the computer has restarted following a power failure and displayed in the WinCC alarm view in Runtime mode

The WinCC alarm system will always work with the default settings. Changing these default settings only serves to adapt the alarm system behavior to specific plant conditions. To achieve this, the required settings are made for alarms and logging in the "Runtime settings".

### Duration of display suppression

So as not to overload the plant operator with information, the display of alarms can be suppressed automatically or manually. If alarms are suppressed (hidden), they are logged but not displayed.

There are two ways of suppressing the display of incoming alarms:

Automatic suppression of the display

The alarms are not displayed depending on a certain system status. To do this, alarm suppression must be configured for certain plant statuses, for example plant startup or plant shutdown.

Manual suppression of the display

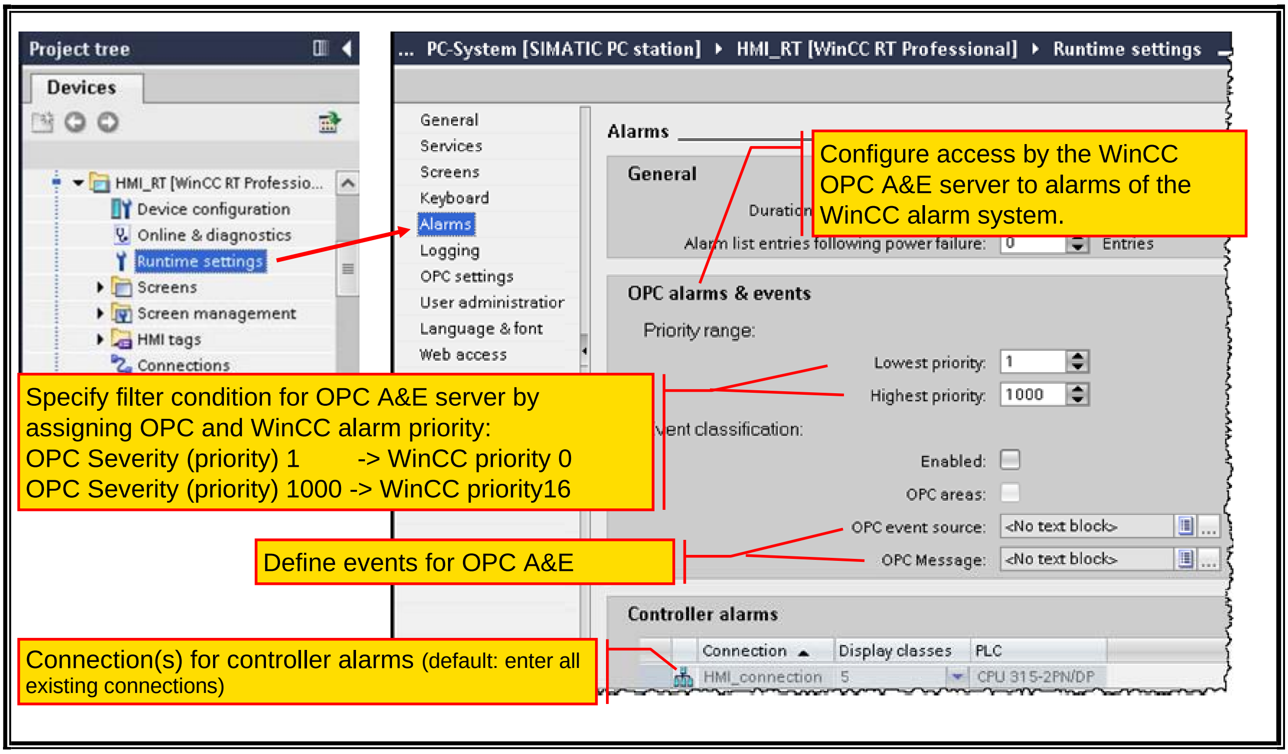
The user suppresses the display of an alarm in runtime when necessary using the "Hide alarm" button in the alarm view. When necessary, the user can display the alarm again using the "Show alarms" button.

After a selectable time ("Duration of display suppression"), the system shows the alarms again.

### Backtransfer of alarms following a computer restart after a power failure

To be able to display the latest alarms in the WinCC alarm view in runtime following a power failure and computer restart, the backtransfer of alarms must be configured.

This is configured using the "Alarms" entry in the area navigation of the "Runtime settings". Here, the number of alarms to be reloaded can be entered in the "Alarm list entries following power failure" parameter. The configured number of alarms is then reloaded from the alarm log following a power failure and computer restart and displayed in the WinCC alarm view in runtime mode.



Alarms

General

Duration

Alarm list entries following power failure: 0 Entries

OPC alarms & events

Priority range:

Lowest priority: 1

Highest priority: 1000

Event classification:

Enabled: □

OPC areas: □

OPC event source: <No text block>

OPC Message: <No text block>

Controller alarms

| Connection | Display classes | PLC |
|---|---|---|
| HMI_connection | 5 | CPU 315-2PN/DP |

Configure access by the WinCC OPC A&E server to alarms of the WinCC alarm system.

Specify filter condition for OPC A&E server by assigning OPC and WinCC alarm priority:
OPC Severity (priority) 1 -> WinCC priority 0
OPC Severity (priority) 1000 -> WinCC priority16

Define events for OPC A&E

Connection(s) for controller alarms (default: enter all existing connections)

### WinCC OPC Alarms & Events server

The WinCC OPC A&E server allows access to alarms of the WinCC alarm system as well as to logged alarms.

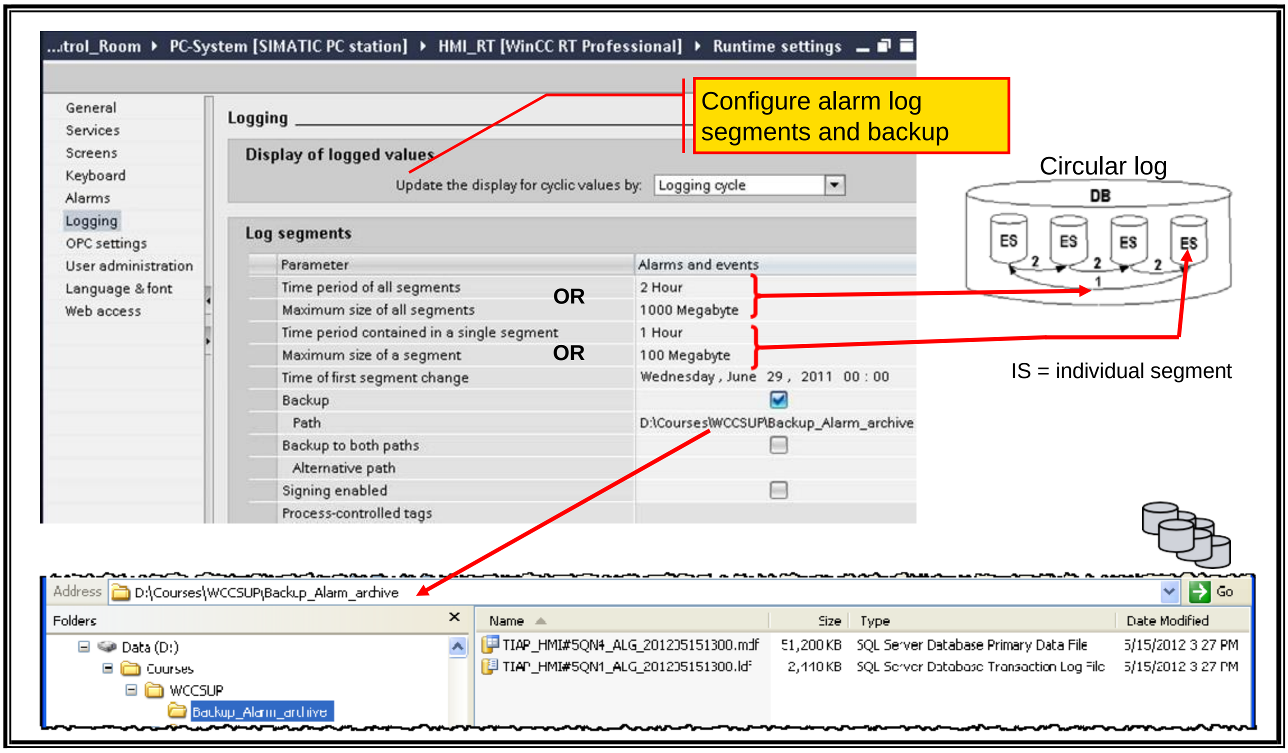
Using filter settings, you can specify which alarms are displayed on the OPC A&E client. These filter settings include the alarm priorities 0 - 16 of WinCC.

In addition to this, events must be defined such as the "OPC event source" and the "OPC message" so that it is possible to access the alarms filtered by the priority range from the OPC A&E client.

### Controller alarms

In this area, the connections required for the controller alarms (alarm number method / alarms in correct chronological order) are managed. Controller alarms are configured in STEP 7. If WinCC is operated in the STEP 7 environment, controller alarms are available in WinCC.

The displayed controller alarms can be filtered using "display classes". The "display classes" are configured in the controller for the controller alarms.



...trol_Room ▸ PC-System [SIMATIC PC station] ▸ HMI_RT [WinCC RT Professional] ▸ Runtime settings

Configure alarm log segments and backup

Logging

Display of logged values

Update the display for cyclic values by: Logging cycle

Log segments

| Parameter | | Alarms and events |
|---|---|---|
| Time period of all segments | OR | 2 Hour |
| Maximum size of all segments | | 1000 Megabyte |
| Time period contained in a single segment | OR | 1 Hour |
| Maximum size of a segment | | 100 Megabyte |
| Time of first segment change | | Wednesday , June 29 , 2011 00 : 00 |
| Backup | | ☑ |
| Path | | D:\Courses\WCCSUP\Backup_Alarm_archive |
| Backup to both paths | | □ |
| Alternative path | | |
| Signing enabled | | □ |
| Process-controlled tags | | |

IS = individual segment

Address D:\Courses\WCCSUP\Backup_Alarm_archive

| Name | Size | Type | Date Modified |
|---|---|---|---|
| TIAP_HMI#5QN4_ALG_201205151300.mdf | 51,200 KB | SQL Server Database Primary Data File | 5/15/2012 3 27 PM |
| TIAP_HMI#5QN1_ALG_201205151300.ldf | 2,110 KB | SQL Server Database Transaction Log File | 5/15/2012 3 27 PM |

## Logging of alarms / log segments

In the "Logging" area navigation in the runtime settings, you configure the logging settings for alarms and events. Here, you specify the period for which the data of "alarms and events" are logged and the point in time as of which the data can be overwritten. In addition to this, the backup function is also enabled here and the path for swapping out the log segments for backup is entered.

## Time period / Size for all segments

The "time period of all segments" or "Maximum size of all segments" relates to the entire log. If one or other limit is exceeded, old segments are deleted.

The following limits apply to the segment settings:

- Smallest unit for the time period for all segments: 1 hour
- Smallest unit for the size for all segments: 32 megabytes (and at least twice as large as a single segment)

## Time period / Size of a single segment

The "Time period contained in a single segment" or "Maximum size of a segment" relates to the segment files. This means that the log can be distributed in several database files (segments).

## The time of (first) segment change

Here, a defined start time can be entered, for example if there is a segment time period setting of one day, each segment will start at midnight. If the project is started the first time at 4 pm, the first segment covers a period of 4 pm to midnight. Following this, each segment covers the time from midnight to midnight.

Each time there is a change in the configuration, a new segment is started in RT.

### Note on the time for swapping out

The time for swapping out is 15 min after the log segment (time period) is completed or the maximum log segment size has been reached.

Example of runtime data:

COMPUTER_PROJECT_ALG_<period_from>_<period_to>.LDF
COMPUTER_PROJECT_ALG_<period_from>_<period_to>.MDF
TIAP_HMI#WHEF_ALG_201109230800_201109230816.ldf
TIAP_HMI#WHEF_ALG_201109230080_201109230816.mdf

### Backup activated

Activates swapping out of archive data to the directories specified under "Path" and / or "Alternative path".

### Backup to both paths

Activates swapping out of archive data to both directories "Path" and "Alternative path".

### Alternative path

The "Alternative path" setting is used when, for example the network path for the backup is not available, for example due to network failure.

### Signing enabled

If signing and backup are activated, each archive backup file is signed when it is swapped out. If the file is then reconnected to WinCC, it is possible to recognize whether or not the file has been changed since it was swapped out.

### Deleting a backup

To delete a log backup, delete the relevant LDF and MDF file in the Windows Explorer.

### Evaluating swapped out log backup files with the WinCC/DataMonitor option

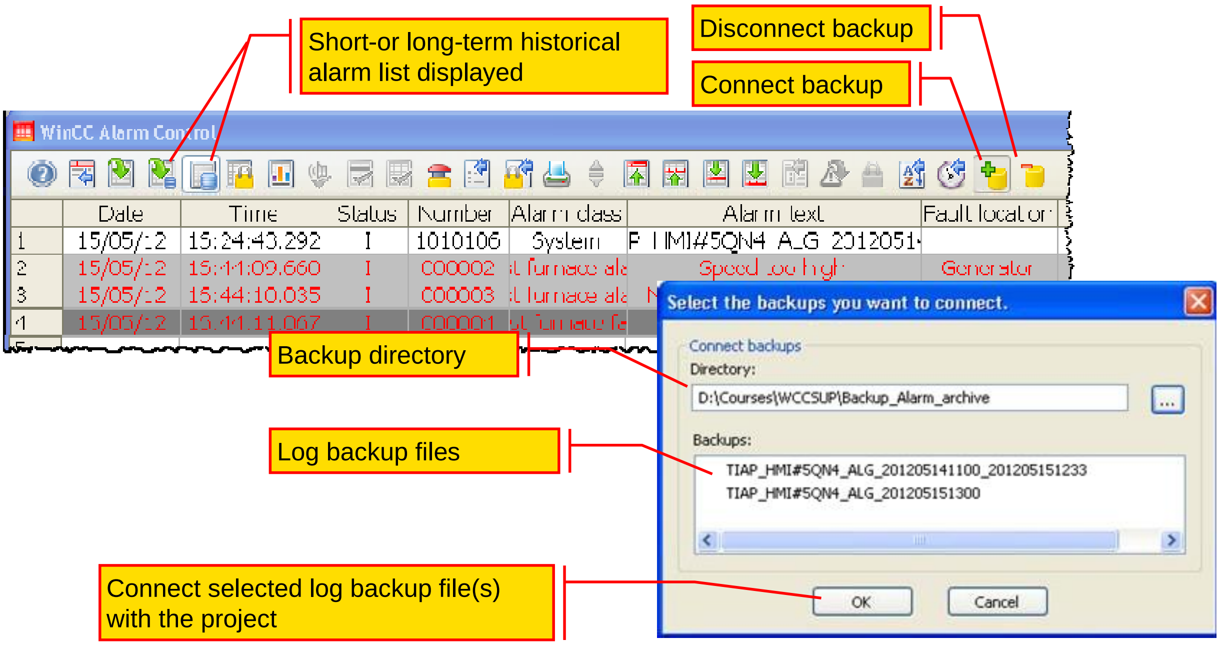
With the WinCC/DataMonitor option, swapped out log backup files can be evaluated in a Web-based environment. With the WinCC/DataMonitor option, WinCC process data can also be made available to all function levels of a company via the Internet / intranet.

### Access to swapped out log backup files in Runtime mode

There are various ways of reading in again and then displaying/evaluating in the WinCC alarm view:

- in the WinCC alarm view using the "Connect backup" and "Disconnect backup" buttons
- copying swapped out log segments to the "Common Archiving" subdirectory of the current Runtime project in Runtime mode. The copying can be performed manually with the Windows Explorer or using a VB script

Select alarm log backup files in Runtime mode and use/evaluate in the alarm view


Short-or long-term historical alarm list displayed
Disconnect backup
Connect backup
Backup directory
Log backup files
Connect selected log backup file(s) with the project
WinCC Alarm Control

| | Date | Time | Status | Number | Alarm class | Alarm text | Fault location |
|---|---|---|---|---|---|---|---|
| 1 | 15/05/12 | 15:24:43.292 | I | 1010106 | System | P HMI#5QN4_ALG_2012051 | |
| 2 | 15/05/12 | 15:44:09.660 | I | 000002 | t furnace ala | Speed too high | Generator |
| 3 | 15/05/12 | 15:44:10.035 | I | 000003 | t furnace ala | N | |
| 4 | 15/05/12 | 15:44:11.067 | I | 000001 | t furnace fa | | |

Select the backups you want to connect.

Connect backups

Directory: D:\Courses\WCCSUP\Backup_Alarm_archive

Backups:

TIAP_HMI#5QN4_ALG_201205141100_201205151233
TIAP_HMI#5QN4_ALG_201205151300

OK Cancel

### Historical alarm list (short-term)

With the "Historical alarm list (short-term)" button, logged and current alarms can be displayed. The new incoming alarms are updated immediately in the short-term historical alarm list.

### Historical alarm list (long-term)

With the "Historical alarm list (long-term)" button, only the logged alarms are displayed. In the long-term historical alarm list, a comment can be entered for each alarm.

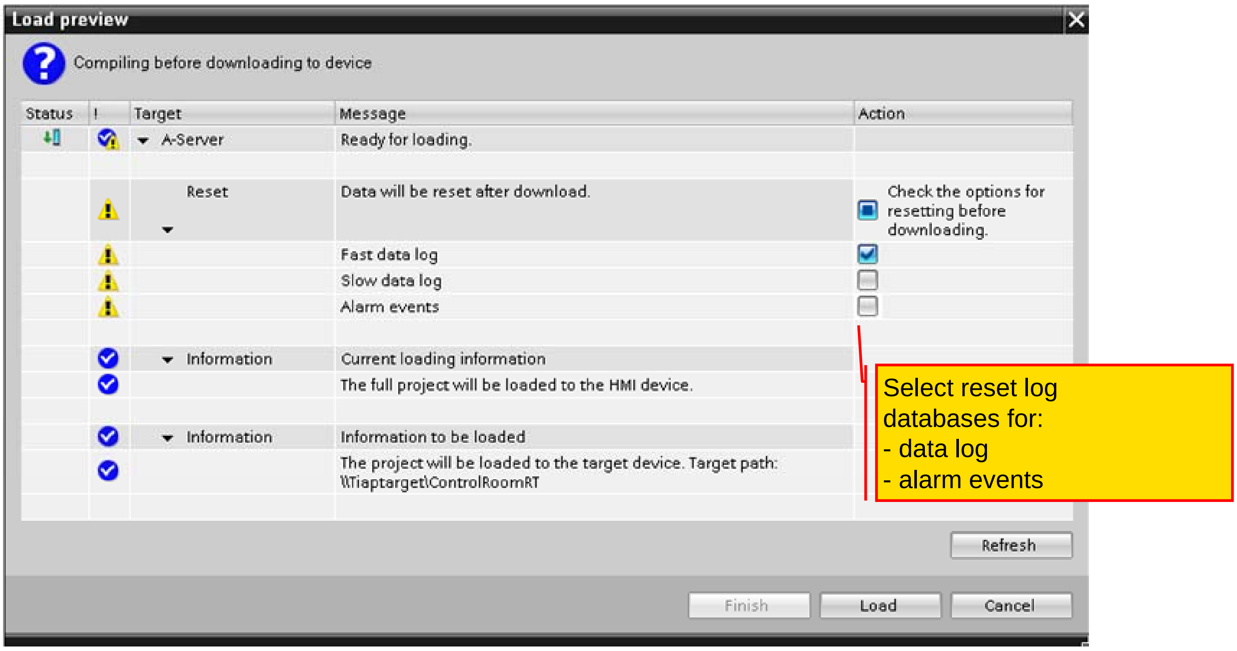
### Connect backup

The "Connect backup" button opens the dialog for selecting the required swapped out log backup files. One or more log backup files can be selected. Clicking the "OK" button in the selection dialog connects the swapped out log backup files to the project. The content of the newly connected log backup files is then visible in the short-term or long-term historical alarm list of the alarm view and can, if necessary be evaluated.

### Disconnect backup

The "Disconnect backup" button opens a dialog for selecting the archive backup files to be disconnected from the project. One or more log backup files can be selected. Clicking the "OK" button in the selection dialog disconnects the swapped out log backup files from the project. This function can also be used to disconnect/delete log backup files copied into the "Common Archiving" folder.

**Log reset when loading project (Online > Expanded download to device)**



Compiling before downloading to device

| Status | ! | Target | Message | Action |
|---|---|---|---|---|
| ↓ | ✓ | ▾ A-Server | Ready for loading. | |
| | ⚠ | Reset ▾ | Data will be reset after download. | ■ Check the options for resetting before downloading. |
| | ⚠ | | Fast data log | ☑ |
| | ⚠ | | Slow data log | □ |
| | ⚠ | | Alarm events | □ |
| | ✓ | ▾ Information | Current loading information | |
| | ✓ | | The full project will be loaded to the HMI device. | |
| | ✓ | ▾ Information | Information to be loaded | |
| | ✓ | | The project will be loaded to the target device. Target path: \\Tiaptarget\ControlRoomRT | |

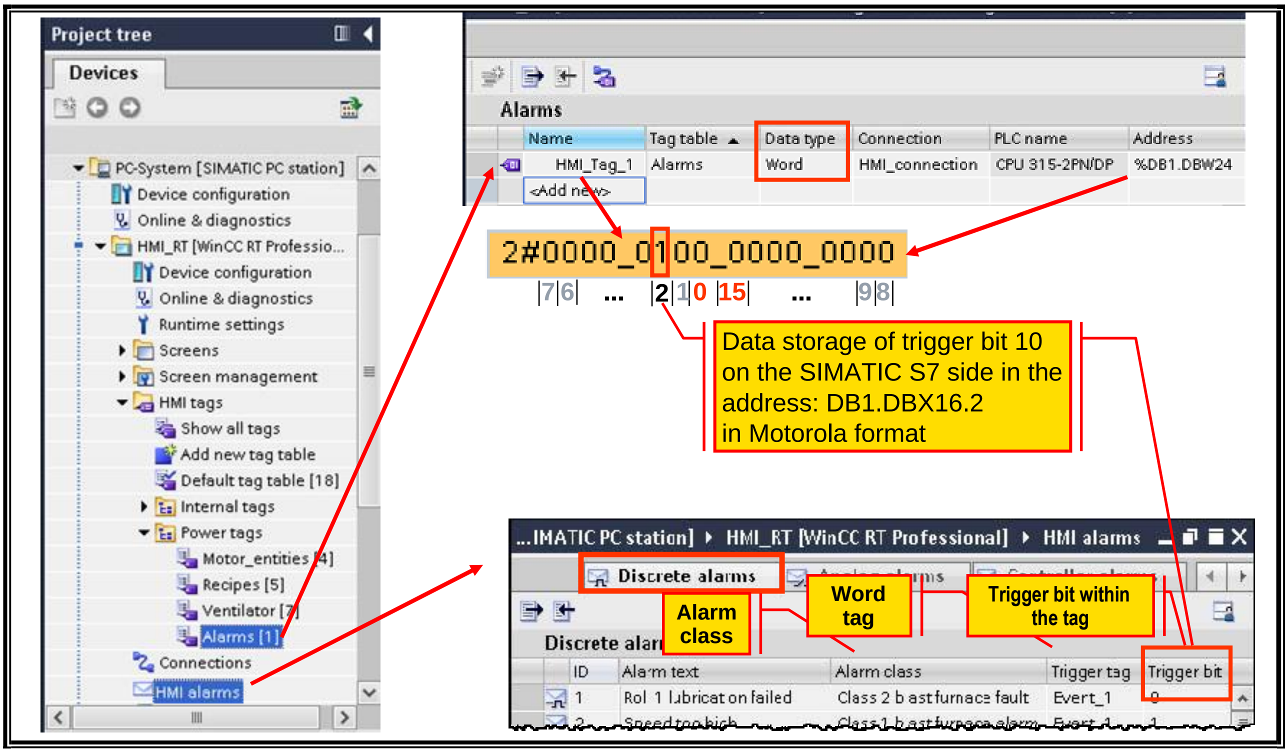
Select reset log databases for:
- data log
- alarm events

### Resetting the logging database

The settings for resetting the logging database are in the "Extended download" dialog. To use this function, the HMI system is selected in the project tree followed by the menu command "Online > Extended download to device". In the following dialog, the interface to the HMI system needs to be selected. The "Load" button opens the next dialog "Load preview". Here, the logs to be deleted are selected. The "Load" button starts the load procedure and the reset (deletion of the log database files).

## 7.2.5. Discrete alarms

### 7.2.5.1. Trigger tags and their bit assignment for discrete alarm texts



Alarms

| Name | Tag table | Data type | Connection | PLC name | Address |
|---|---|---|---|---|---|
| HMI_Tag_1 | Alarms | Word | HMI_connection | CPU 315-2PN/DP | %DB1.DBW24 |
| <Add new> | | | | | |

2#0000_0100_0000_0000

|7|6| ... |2|1|0 |15| ... |9|8|

Data storage of trigger bit 10 on the SIMATIC S7 side in the address: DB1.DBX16.2 in Motorola format

...IMATIC PC station] ▸ HMI_RT [WinCC RT Professional] ▸ HMI alarms

Discrete alarms

| ID | Alarm text | Alarm class | Trigger tag | Trigger bit |
|---|---|---|---|---|
| 1 | Roll 1 lubrication failed | Class 2 blast furnace fault | Event_1 | 0 |
| 2 | Speed too high | Class 1 blast furnace alarm | Event_1 | 1 |

To generate a discrete alarm, there must be a bit edge change. These bits are managed in trigger tags. These trigger tags are edited using the tag editor or can be created directly when configuring a discrete alarm.

On the SIMATIC side, tag storage is always organized on a byte basis. If, however, two bytes are put together to form a word, for example DB1.DBW16 (byte 16, byte 17), the data storage on the SIMATIC S7 side will be in the Motorola format. In this format (big endian), the byte with the most significant bits is stored first, in other words at the lowest memory address (in the example above in byte 16 of data block DB1).

**Trigger tags**

When using the trigger tags, remember the following:

- permitted data types (HMI or S7 side) for trigger tags:

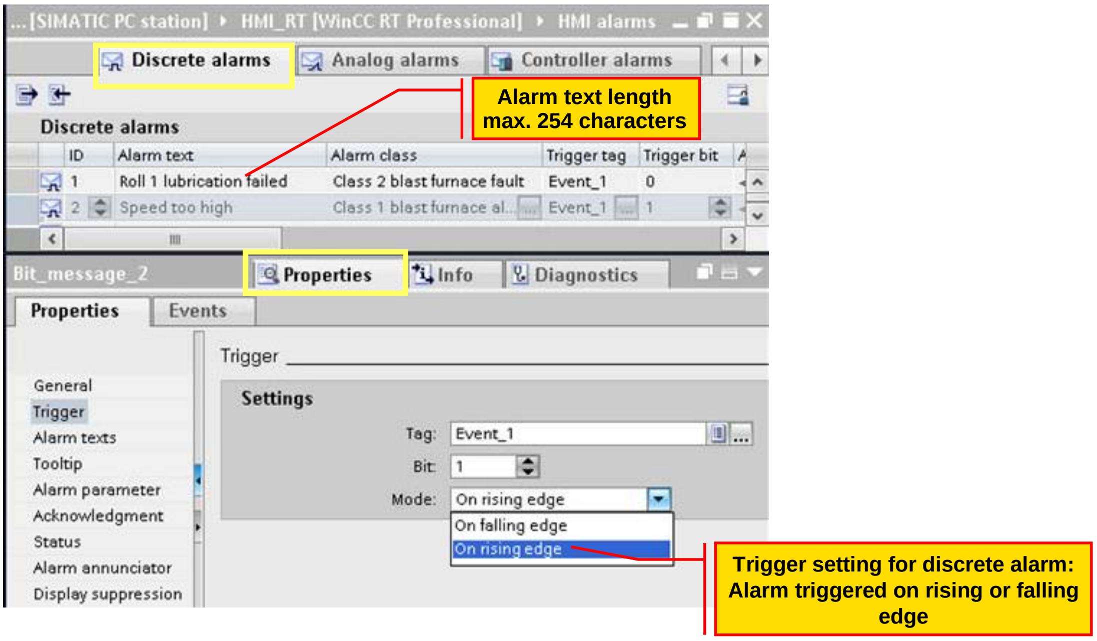
| | |
|---|---|
| - HMI: "Bool", binary tag | S7: Bool |
| - HMI: "USInt", unsigned 8-bit value | S7: Byte |
| - HMI: "UInt", unsigned 16-bit value | S7: Word |
| - HMI: "UDInt", unsigned 32-bit value | S7: DWord |

- use a bit of a trigger tag only for one alarm
- use trigger tag only as a trigger tag

☞ For performance reasons it is advisable not to use scattered address areas for the trigger tags (use contiguous data areas as far as possible), so that all trigger tags can be read from the controller by the HMI device with as few read requests as possible.

### 7.2.5.2. Triggering discrete alarms (edge setting)

**Edge selection for triggering discrete alarms**



...[SIMATIC PC station] ▸ HMI_RT [WinCC RT Professional] ▸ HMI alarms

Discrete alarms

| ID | Alarm text | Alarm class | Trigger tag | Trigger bit |
|---|---|---|---|---|
| 1 | Roll 1 lubrication failed | Class 2 blast furnace fault | Event_1 | 0 |
| 2 | Speed too high | Class 1 blast furnace al... | Event_1 | 1 |

Trigger

Settings

Tag: Event_1

Bit: 1

Mode: On rising edge

Alarm text length max. 254 characters

Trigger setting for discrete alarm: Alarm triggered on rising or falling edge

The edge change of a bit in the trigger tag is the trigger for an alarm. On a rising edge, the alarm response is as follows:

- "0" → "1" Incoming alarm
- "1" → "0" Outgoing alarm

Corresponding trigger variables are assigned to the alarm texts.

☞ Each bit of a trigger tag may only be assigned to one alarm.

**Configuring a new alarm**

A new alarm is created by clicking in the next free line. An alarm class and trigger tag (word tag with bit number) must be assigned to this.

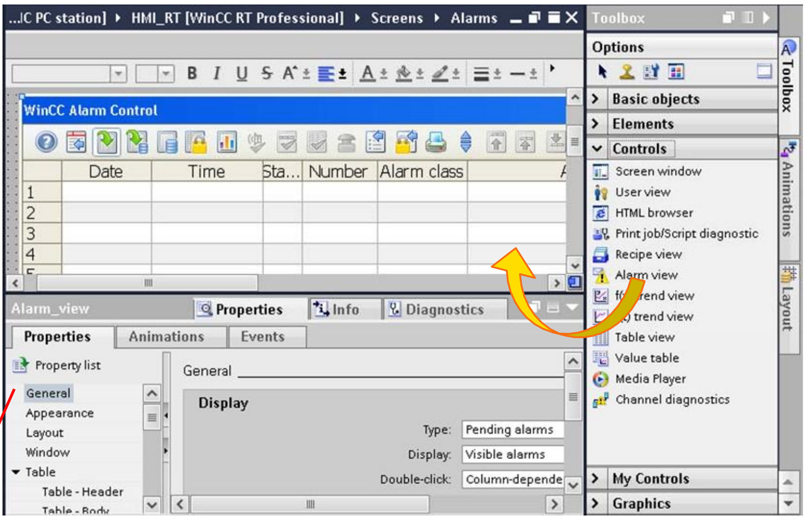
**Alarm text**

The alarm text contains a description of the alarm. The alarm text can contain output fields for the current values of tags. The alarm retains the instantaneous value of the inserted tags at the time at which the alarm status changes.

## 7.3. Displaying alarms

**Display of pending / logged alarms**

- Drag the control to the screen
- Set the required data view:
  - alarm list
  - short-term historical alarm list
  - long-term historical alarm list
  - alarm statistics
  - suppressed alarms
- Versatile parameter assignment:
  - alarm line structure
  - toolbar
  - evaluation functions
- User-defined views, e.g. filters for plant areas
- Configurable in Runtime mode



Type: Pending alarms
Display: Visible alarms
Double-click: Column-depende

**Configure the required alarm view, filter, toolbar etc**

Several alarm views can be configured for different alarm classes and in different screens or windows.

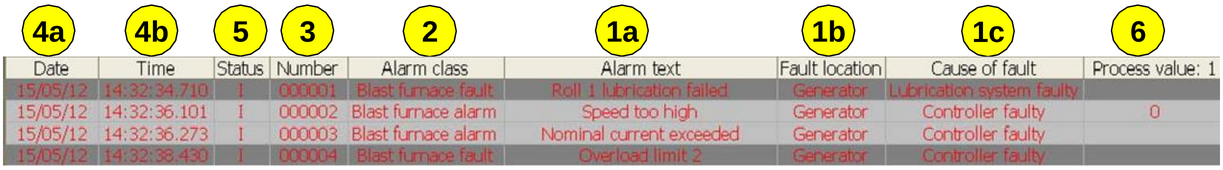
**Alarm view**

- Display of currently pending alarms or alarms from the alarm log
- Selection of alarms according to alarm class
- Can be configured in a screen or screen window
- Is only visible if the corresponding screen is open
- Optional display mode: table | alarm line

### 7.3.1. Content of the alarm view and alarm window

**Basic alarm list components (blocks), display in Runtime mode:**

**1** - Alarm text, max. 10 user text blocks
**2** - Alarm class for alarm properties
**3** - Alarm number
**4** - Time stamp
**5** - Status
**6** - Process value, max. 10 parameter blocks



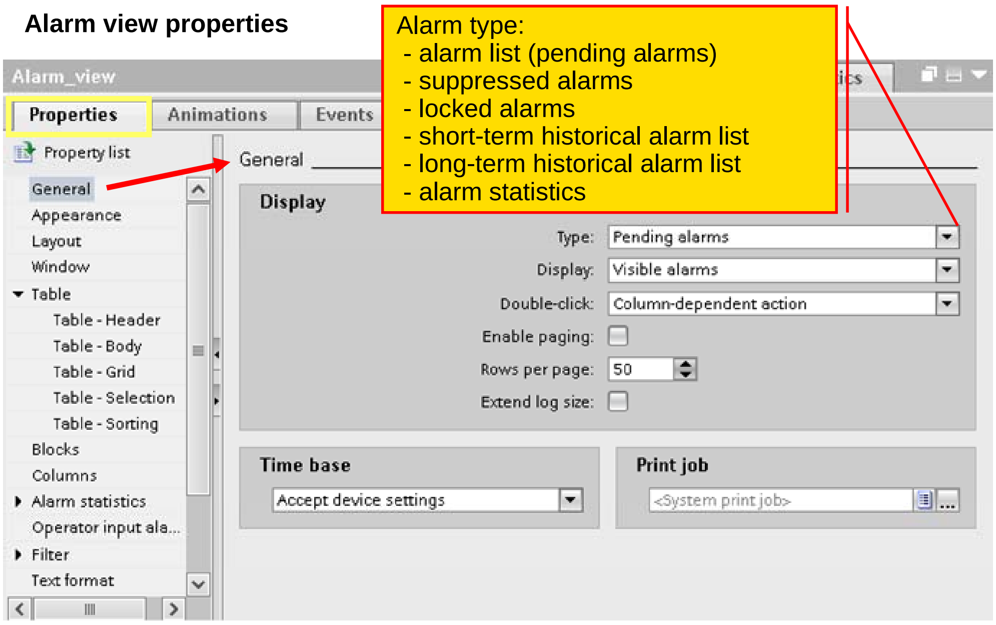
| Date | Time | Status | Number | Alarm class | Alarm text | Fault location | Cause of fault | Process value: 1 |
|---|---|---|---|---|---|---|---|---|
| 15/05/12 | 14:32:34.710 | I | 000001 | Blast furnace fault | Roll 1 lubrication failed | Generator | Lubrication system faulty | |
| 15/05/12 | 14:32:36.101 | I | 000002 | Blast furnace alarm | Speed too high | Generator | Controller faulty | 0 |
| 15/05/12 | 14:32:36.273 | I | 000003 | Blast furnace alarm | Nominal current exceeded | Generator | Controller faulty | |
| 15/05/12 | 14:32:38.430 | I | 000004 | Blast furnace fault | Overload limit 2 | Generator | Controller faulty | |

**Alarm properties**

- Acknowledgment
- Color scheme
- Alarm status (I - Incoming, O - Outgoing, IO - without acknowledgment incoming/outgoing, A - acknowledged)
- Archiving

### 7.3.2. Alarm view properties

**Alarm view properties**



Alarm type:
- alarm list (pending alarms)
- suppressed alarms
- locked alarms
- short-term historical alarm list
- long-term historical alarm list
- alarm statistics

Type: Pending alarms
Display: Visible alarms
Double-click: Column-dependent action
Enable paging: □
Rows per page: 50
Extend log size: □
Time base: Accept device settings
Print job: <System print job>

**Alarm view properties**

The settings for the position, geometry, style, color and font of the object are made in the Inspector window. The following settings need to be selected for an alarm list display (list of currently pending alarms):

General/Display:

| | |
|---|---|
| Type: | Pending alarms |
| Display: | All alarms |
| Double-click: | e.g. loop-in alarm: Specifies that double-clicking on an alarm, triggers a configured screen change. |

General/Time base: Accept device settings.
Here, you specify the time base with the alarms are output.

Layout: Here, the position and size of the alarm view is defined.

Window: Settings title: e.g. "Blast furnace-W"

Table: activate display of the column and row headers.
Specifies the properties (color, column labeling, row numbers) of the table in the alarm view.

Blocks: Adopt project settings
With these settings, the blocks created in the alarm configuration are selected for further use in the column configuration.

Columns: Specifies the columns of the table in the alarm view. Each column corresponds to an alarm text block. Here all available alarm text blocks should be selected. Here, only the alarm text blocks selected under "Blocks" are displayed.

Toolbar: At least the following buttons should be available in the taskbar:
Alarm list, short and long-term archive, acknowledgment (single/group acknowledgment), selection dialog (filter), auto scrolling and for navigatio in the alarm lists: First alarm, next alarm, previous alarm, latest alarm

Status bar: Specifies the elements of the status bar. In runtime mode, these elements show, for example, information about the number of pending alarms.

Text format: Specifies the font and size.

## 7.3.3. Statistical functions for alarms in the alarm view

The alarm statistics represent statistical calculations relating to logged alarms in the alarm view. The alarm statistics also show a selection of the configured alarm text blocks. If the contents are dynamic, the alarm text blocks display the data for the last incoming alarm. The columns of the alarm statistics can be put together individually.

The alarm statistics shows the following statistical calculations:

**Alarm frequency**

The system counts how often an alarm with the "Incoming" status exists in the log. If the alarm number does not exist, this alarm number is not entered in the statistics.

**Total display duration of an alarm in seconds**

As the basis of the calculation, the following periods between the alarm statuses are available:

"Incoming/incoming" (sum +/+),
"Incoming/outgoing (sum +/-)
"Incoming/first acknowledgement" (sum +/*1)
"Incoming/second acknowledgement" (sum +/*2)

Each type of calculation is displayed when necessary in a column in the alarm statistics.

**Average display duration of an alarm in seconds**

As the basis of the calculation, the following periods between the alarm statuses are available:

"Incoming/outgoing" (average +/+),
"Incoming/outgoing" (average +/-)
"Incoming/first acknowledgement" (average +/*1)
"Incoming/second acknowledgement" (average +/*2)

There is a column in the alarm statistics for each type of calculation.

### 7.3.4. Filtering the content of the alarm view / alarm window

The display of alarms can be filtered based on the alarm text. The filter only applies to the display on the HMI device. All alarms are retained in the alarm buffer.

WinCC provides the option of configuring various filters for the alarm view:

Standard filter (cannot be modified in Runtime mode)

In Runtime, the operator has no access to a standard filter. A configured standard filter can, when necessary, be copied during configuration and used in other alarm views.

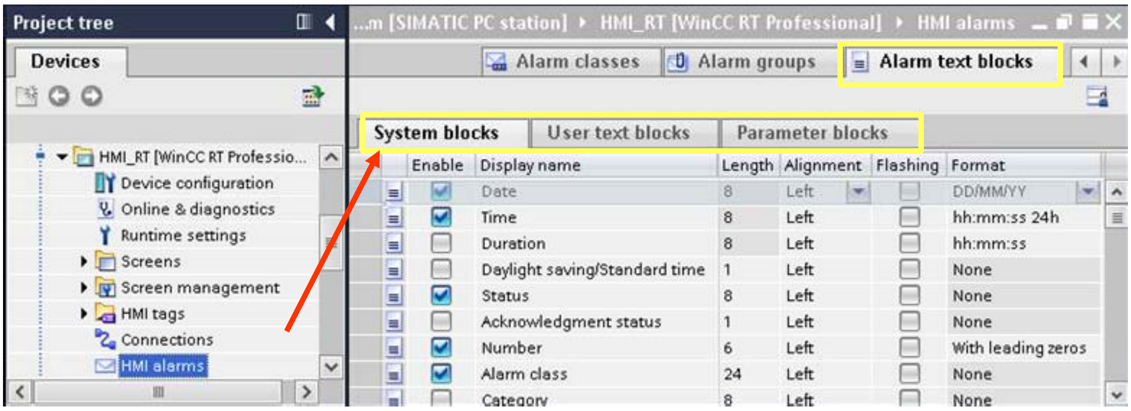
User-defined filter (can be modified in Runtime mode)

When necessary, the operator can modify a user-defined filter in Runtime. A user-defined filter cannot be copied. A user-defined filter can therefore only be used in one alarm view.

## 7.4. Exercise 1: Create alarm text blocks

**Procedure**

### 1. Open HMI alarms editor



| Enable | Display name | Length | Alignment | Flashing | Format |
|---|---|---|---|---|---|
| ☑ | Date | 8 | Left | □ | DD/MM/YY |
| ☑ | Time | 8 | Left | □ | hh:mm:ss 24h |
| □ | Duration | 8 | Left | □ | hh:mm:ss |
| □ | Daylight saving/Standard time | 1 | Left | □ | None |
| ☑ | Status | 8 | Left | □ | None |
| □ | Acknowledgment status | 1 | Left | □ | None |
| ☑ | Number | 6 | Left | □ | With leading zeros |
| ☑ | Alarm class | 24 | Left | □ | None |
| □ | Category | 8 | Left | □ | None |

- **Configure alarm text blocks**
  **→ System blocks**
  **→ User text blocks**
  **→ Parameter blocks**

**Task description**

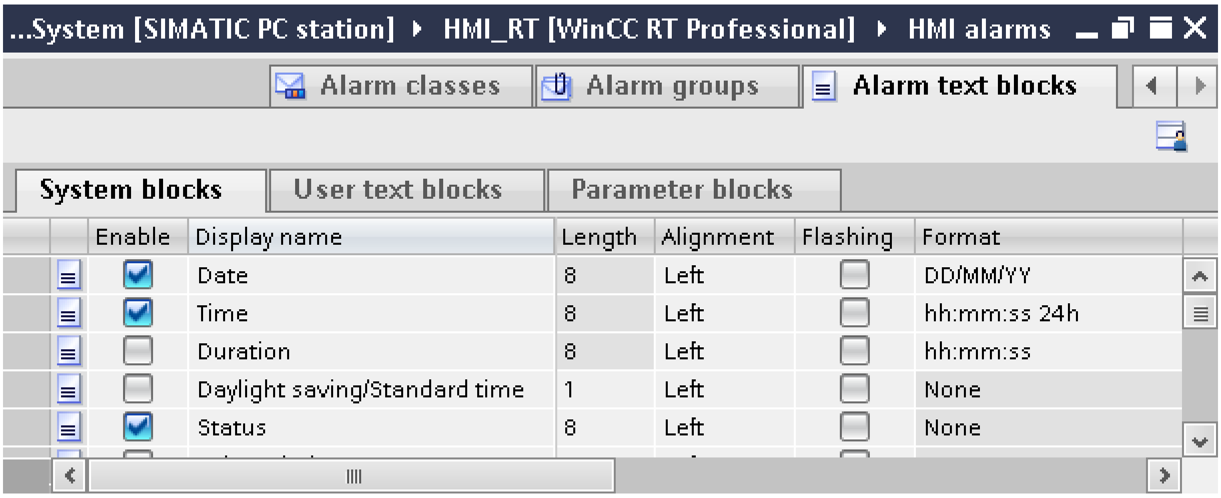
The Control_Room project needs to be expanded with alarms. In the first step, the various alarm text blocks are created/edited.

**1. Configuring alarm text blocks**

Open the HMI alarms editor to configure the system, user and parameter blocks.

*>> Project tree > HMI device > HMI alarms > "Alarm text blocks" tab*

### 2. Configuring system alarm text blocks

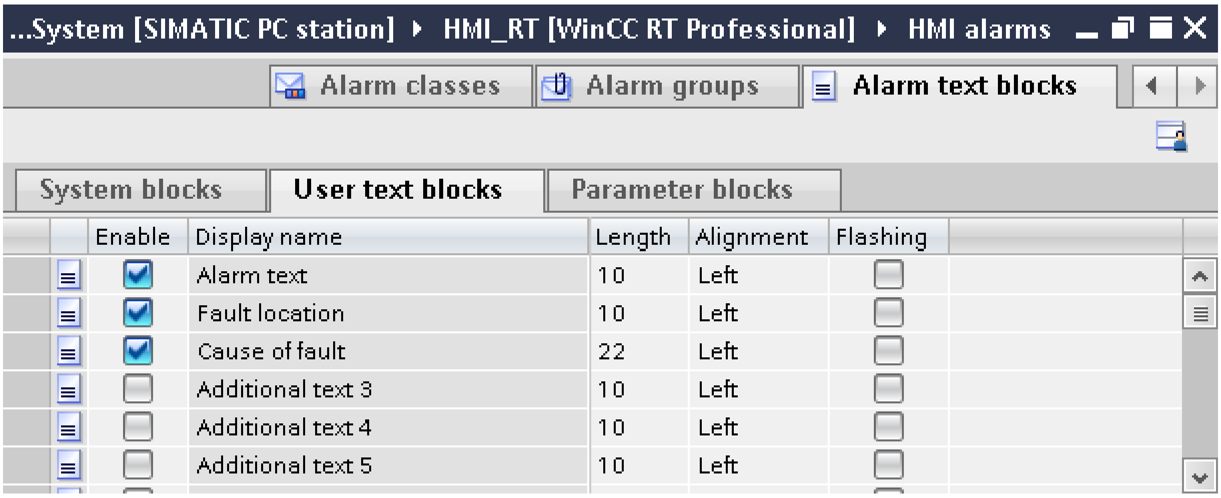


| Enable | Display name | Length | Alignment | Flashing | Format |
|---|---|---|---|---|---|
| ☑ | Date | 8 | Left | ☐ | DD/MM/YY |
| ☑ | Time | 8 | Left | ☐ | hh:mm:ss 24h |
| ☐ | Duration | 8 | Left | ☐ | hh:mm:ss |
| ☐ | Daylight saving/Standard time | 1 | Left | ☐ | None |
| ☑ | Status | 8 | Left | ☐ | None |

Configure the system blocks as shown in the figure above.

>> Project tree > HMI device > HMI alarms > "Alarm text blocks" tab > "System blocks" tab

### 3. Configuring user text blocks

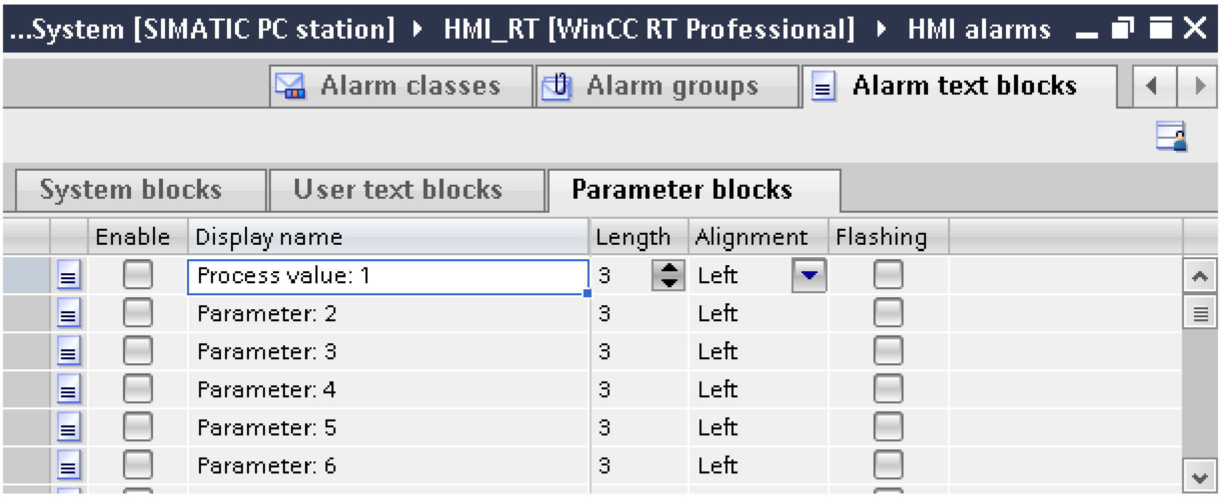


| Enable | Display name | Length | Alignment | Flashing |
|---|---|---|---|---|
| ☑ | Alarm text | 10 | Left | ☐ |
| ☑ | Fault location | 10 | Left | ☐ |
| ☑ | Cause of fault | 22 | Left | ☐ |
| ☐ | Additional text 3 | 10 | Left | ☐ |
| ☐ | Additional text 4 | 10 | Left | ☐ |
| ☐ | Additional text 5 | 10 | Left | ☐ |

Configure the user text blocks as shown in the figure above.

>> Project tree > HMI device > HMI alarms > "Alarm text blocks" tab > "User text blocks" tab

### 4. Configuring parameter blocks



...System [SIMATIC PC station] ▸ HMI_RT [WinCC RT Professional] ▸ HMI alarms

| Enable | Display name | Length | Alignment | Flashing |
|---|---|---|---|---|
| ☐ | Process value: 1 | 3 | Left | ☐ |
| ☐ | Parameter: 2 | 3 | Left | ☐ |
| ☐ | Parameter: 3 | 3 | Left | ☐ |
| ☐ | Parameter: 4 | 3 | Left | ☐ |
| ☐ | Parameter: 5 | 3 | Left | ☐ |
| ☐ | Parameter: 6 | 3 | Left | ☐ |

Configure the parameter blocks as shown in the figure above.

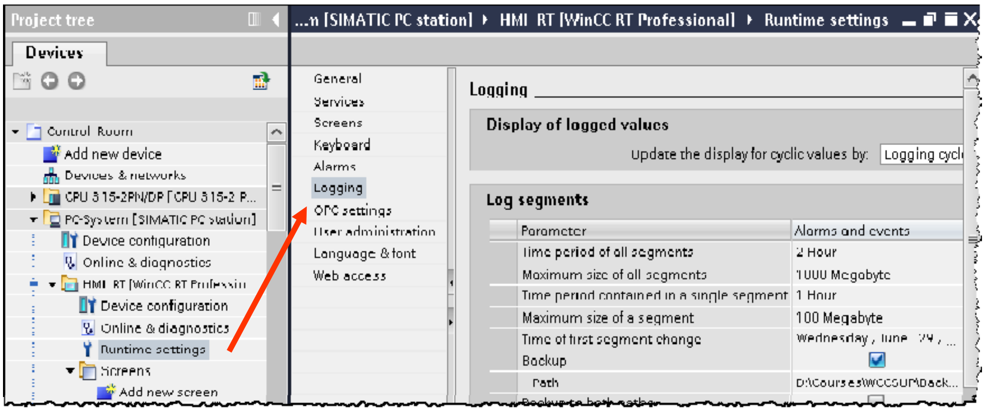
>> Project tree > HMI device > HMI alarms > "Alarm text blocks" tab > "User text blocks" tab

Don't forget to save your project.

## 7.5. Exercise 2: Alarm logging

### Procedure

**1. Open Runtime settings / define segment sizes for alarm logging**



Logging

Display of logged values

Update the display for cyclic values by: Logging cycl

Log segments

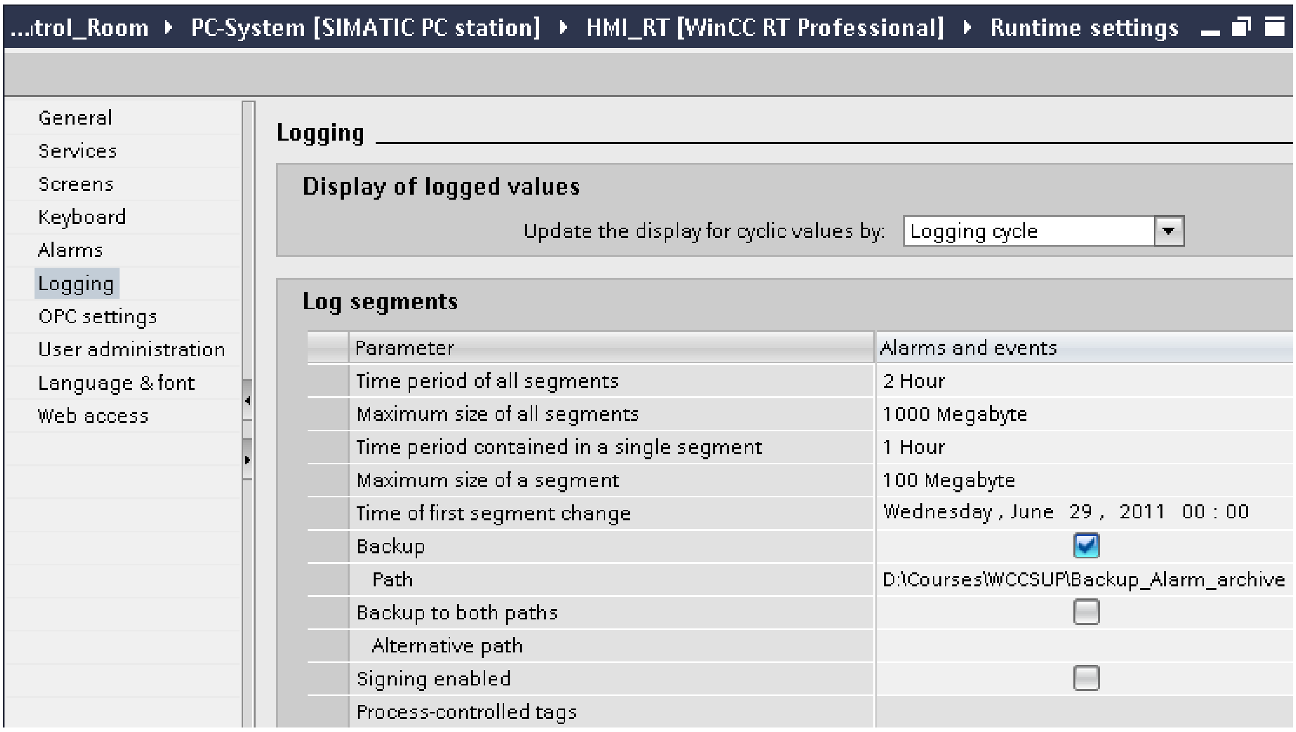
| Parameter | Alarms and events |
|---|---|
| Time period of all segments | 2 Hour |
| Maximum size of all segments | 1000 Megabyte |
| Time period contained in a single segment | 1 Hour |
| Maximum size of a segment | 100 Megabyte |
| Time of first segment change | Wednesday , June 29 , ... |
| Backup | ☑ |
| Path | D:\Courses\WCCSUP\Back... |

- **Enable backup function**
- **Configure backtransfer following power failure**
- **Enable alarm logging in the Runtime settings**

### Task description

The Control_Room project needs to be expanded with alarms. In the second step, the alarm logging is edited.

### 1. Open Runtime settings / define segment sizes for alarm logging



Logging

Display of logged values

Update the display for cyclic values by: Logging cycle

Log segments

| Parameter | Alarms and events |
|---|---|
| Time period of all segments | 2 Hour |
| Maximum size of all segments | 1000 Megabyte |
| Time period contained in a single segment | 1 Hour |
| Maximum size of a segment | 100 Megabyte |
| Time of first segment change | Wednesday , June 29 , 2011 00 : 00 |
| Backup | ☑ |
| Path | D:\Courses\WCCSUP\Backup_Alarm_archive |
| Backup to both paths | □ |
| Alternative path | |
| Signing enabled | □ |
| Process-controlled tags | |

Use the alarm logging parameters shown in the figure above for the segment sizes.

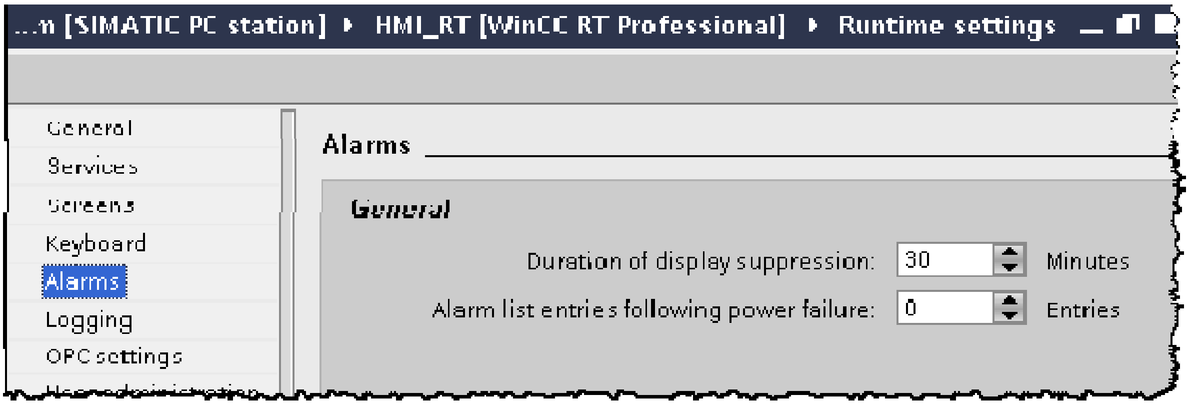
### 2. Configuring the backup function

Create the following directory for the backup files of the alarm log: "D:\Courses\WCCSUP\Backup_Alarm_archive"

Enable the backup function as shown in the figure above and use the path you created earlier that is also shown in the figure above "D:\Courses\WCCSUP\Backup_Alarm_archive" to install the alarm log backup files.

### 3. Configuring backtransfer following power failure

To be able to display the latest alarms in the WinCC alarm view in runtime following a power failure and computer restart, the backtransfer of alarms must be configured.



...n [SIMATIC PC station] ▸ HMI_RT [WinCC RT Professional] ▸ Runtime settings

Alarms

General

Duration of display suppression: 30 Minutes

Alarm list entries following power failure: 0 Entries

With the "Alarm list entries following power failure" parameter, you enter the number of alarms to be reloaded. The configured number of alarms is then reloaded from the alarm log following a power failure and computer restart and displayed in the WinCC alarm view in runtime mode.

Use the value 30 entries following power failure as shown in the figure above.

>> Project tree > HMI device > Runtime settings > Logging

### 4. Enabling alarm logging in the Runtime settings

Enable alarm logging in the Runtime settings of the "HMI system" using "Services > Start sequence of WinCC > Alarm logging in runtime".

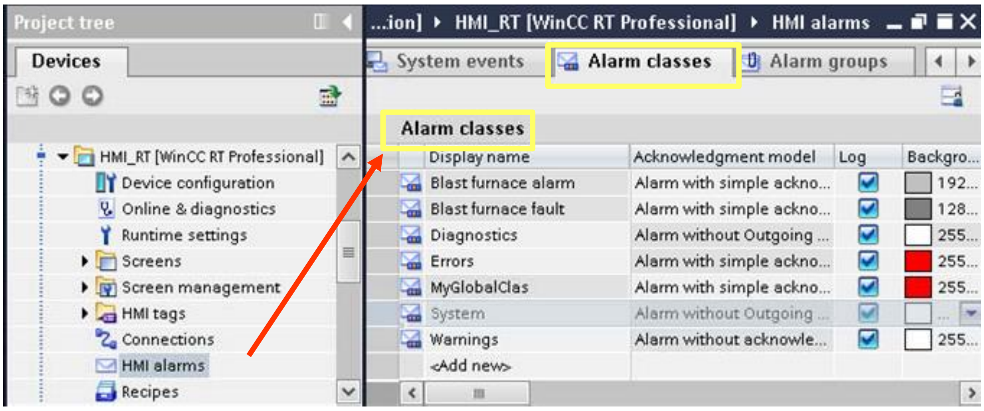
Don't forget to save your project.

STOP

## 7.6. Exercise 3: Alarm classes

**Procedure**

### 1. Open HMI alarms editor



Alarm classes

| Display name | Acknowledgment model | Log | Backgro... |
|---|---|---|---|
| Blast furnace alarm | Alarm with simple ackno... | ☑ | 192... |
| Blast furnace fault | Alarm with simple ackno... | ☑ | 128... |
| Diagnostics | Alarm without Outgoing ... | ☑ | 255... |
| Errors | Alarm with simple ackno... | ☑ | 🟥 255... |
| MyGlobalClas | Alarm with simple ackno... | ☑ | 🟥 255... |
| System | Alarm without Outgoing ... | ☑ | ... |
| Warnings | Alarm without acknowle... | ☑ | 255... |
| <Add new> | | | |

### 2. Configure two new alarm classes:

→ Blast furnace alarm

→ Blast furnace fault

**Task description**

The Control_Room project needs to be expanded with alarms. In the third step, two new alarm classes of your own will be created/edited.

**1. Configure two new alarm classes of your own**

Configure two new alarm classes of your own "Blast furnace alarm" and "Blast furnace fault" using "Add alarm classes". To display the alarm statuses and the classes, the alarms will be displayed in different colors. To do this, you need to define the background and text color.

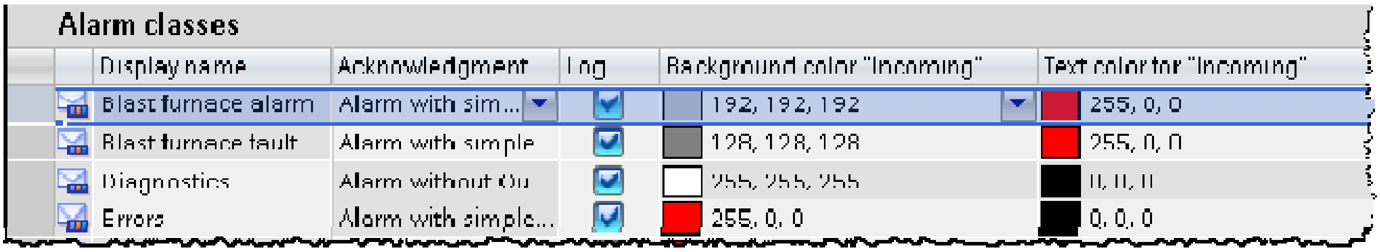
In addition to this, select "Alarm with simple acknowledgement" for the acknowledgement concept. This alarm requires an acknowledgement as soon as the event that triggers the alarm occurs. The alarm remains pending until it is acknowledged.

Logging needs to be activated for both classes so that the alarm log can later be seen in the WinCC alarm view.

*>> Project tree > HMI device > HMI alarms > "Alarm classes" tab*

## 2. Editing the background and text color of alarm classes

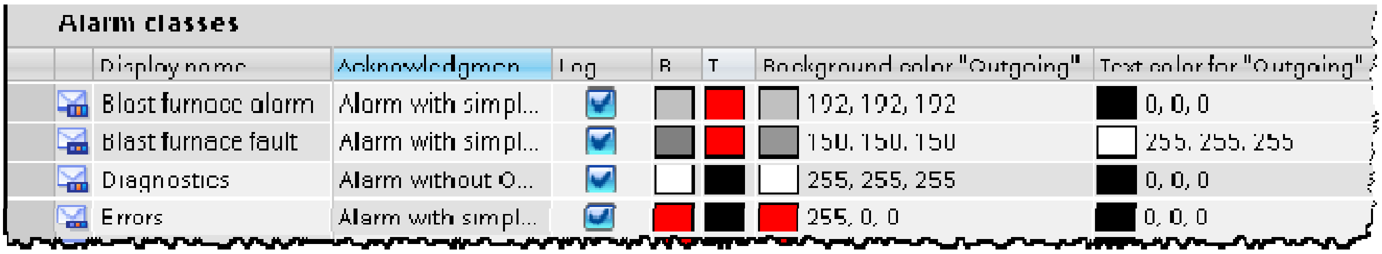
Use the color settings for "Background color Incoming" and "Text color for Incoming" as shown in the figure below.



Alarm classes

| Display name | Acknowledgment | Log | Background color "Incoming" | Text color for "Incoming" |
|---|---|---|---|---|
| Blast furnace alarm | Alarm with sim... | ☑ | ▨ 192, 192, 192 | 🟥 255, 0, 0 |
| Blast furnace fault | Alarm with simple | ☑ | ▨ 128, 128, 128 | 🟥 255, 0, 0 |
| Diagnostics | Alarm without Qu | ☑ | ⬜ 255, 255, 255 | ⬛ 0, 0, 0 |
| Errors | Alarm with simple... | ☑ | 🟥 255, 0, 0 | ⬛ 0, 0, 0 |

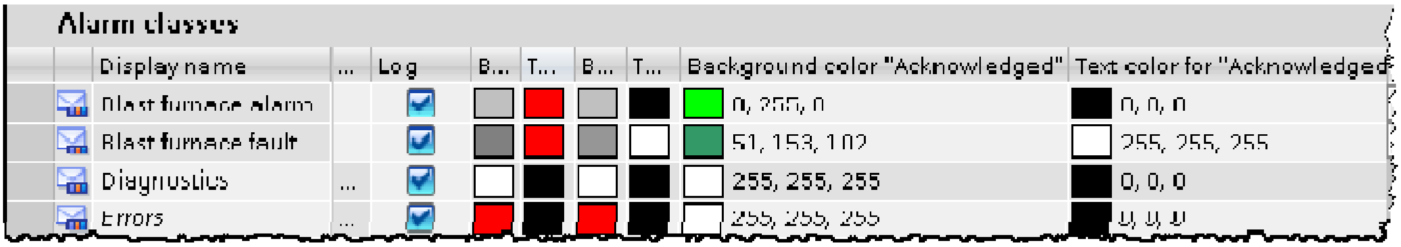
Use the color settings for "Background color Outgoing" and "Text color for Outgoing" as shown in the figure below.



Alarm classes

| Display name | Acknowledgmen | Log | B | T | Background color "Outgoing" | Text color for "Outgoing" |
|---|---|---|---|---|---|---|
| Blast furnace alarm | Alarm with simpl... | ☑ | ▨ | 🟥 | ▨ 192, 192, 192 | ⬛ 0, 0, 0 |
| Blast furnace fault | Alarm with simpl... | ☑ | ▨ | 🟥 | ▨ 150, 150, 150 | ⬜ 255, 255, 255 |
| Diagnostics | Alarm without Q... | ☑ | ⬜ | ⬛ | ⬜ 255, 255, 255 | ⬛ 0, 0, 0 |
| Errors | Alarm with simpl... | ☑ | 🟥 | ⬛ | 🟥 255, 0, 0 | ⬛ 0, 0, 0 |

Use the color settings for "Background color Acknowledged" and "Text color for Acknowledged" as shown in the figure below.



Alarm classes

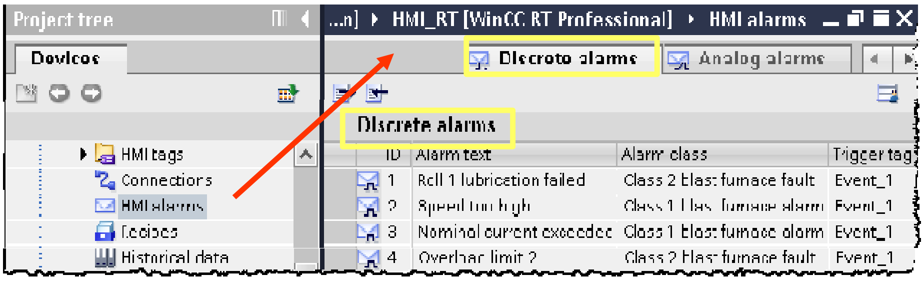
| Display name | ... | Log | B... | T... | B... | T... | Background color "Acknowledged" | Text color for "Acknowledged |
|---|---|---|---|---|---|---|---|---|
| Blast furnace alarm | | ☑ | ▨ | 🟥 | ▨ | ⬛ | 🟩 0, 255, 0 | ⬛ 0, 0, 0 |
| Blast furnace fault | | ☑ | ▨ | 🟥 | ▨ | ⬜ | 🟩 51, 153, 102 | ⬜ 255, 255, 255 |
| Diagnostics | ... | ☑ | ⬜ | ⬛ | ⬜ | ⬛ | ⬜ 255, 255, 255 | ⬛ 0, 0, 0 |
| Errors | ... | ☑ | 🟥 | ⬛ | 🟥 | ⬛ | ⬜ 255, 255, 255 | ⬛ 0, 0, 0 |

Don't forget to save your project.

## 7.7. Exercise 4: Alarms

### Procedure

1. **Create trigger tag "Event_1"**



| ID | Alarm text | Alarm class | Trigger tag |
|---|---|---|---|
| 1 | Roll 1 lubrication failed | Class 2 Blast furnace fault | Event_1 |
| 2 | Speed too high | Class 1 Blast furnace alarm | Event_1 |
| 3 | Nominal current exceeded | Class 1 Blast furnace alarm | Event_1 |
| 4 | Overbar limit 2 | Class 2 Blast furnace fault | Event_1 |

2. **Open the HMI alarms editor and create four new discrete alarms**
3. **Enter message texts**
4. **Assign alarm classes "Blast furnace alarm" and "Blast furnace fault" to the discrete alarms**
5. **Select trigger tag and trigger bit**
6. **Add cause of fault and fault location**

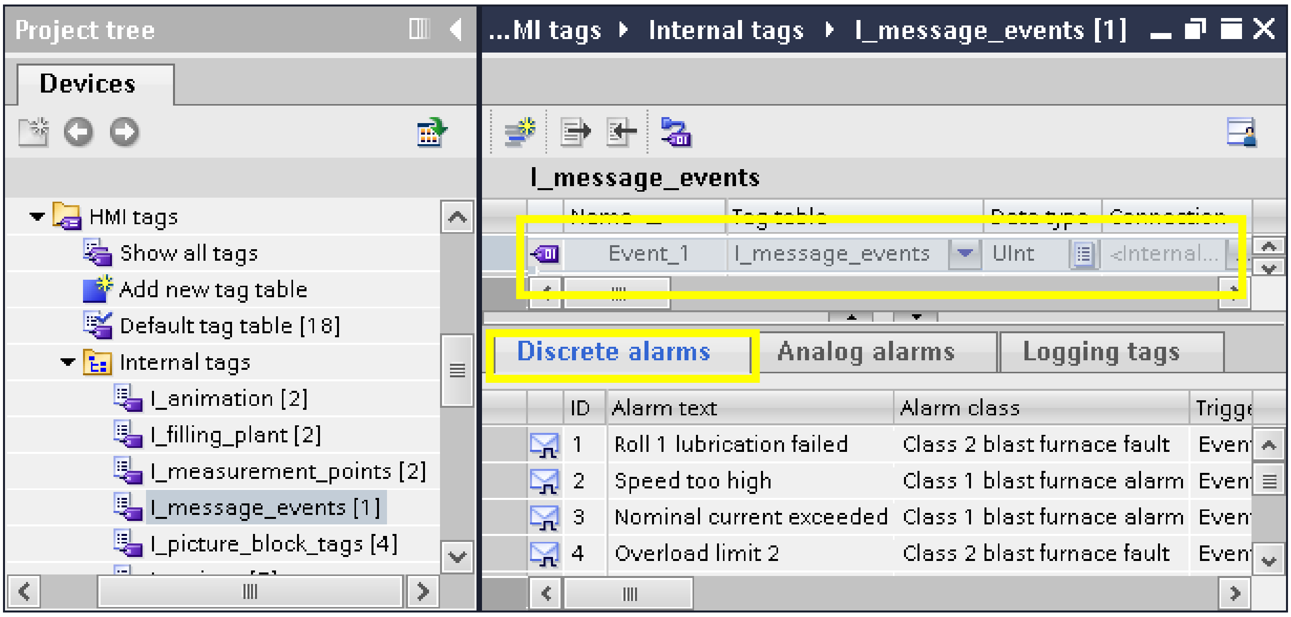
### Task description

The Control_Room project needs to be expanded with alarms. In the next steps, the alarms will be created/edited. Alarms 1-4 will be triggered in Runtime mode by bits 0-3 of the alarm tag "Event_1".

### 1. Create trigger tag "Event_1"

Create the internal tag "Event_1" of the data type "UInt" (HMI: unsigned 16-bit value, S7: word). To make the project easier to understand, the tags can be stored structured in groups and tag tables. In the figure below, the "Internal tags" group and the tag table "I_message_events" are used to store the "Event_1" tag. If alarm classes have already been created you can configure alarms for the tags directly. Logging can also be edited directly with the tags.

>> Project tree > HMI device > HMI tags



I_message_events

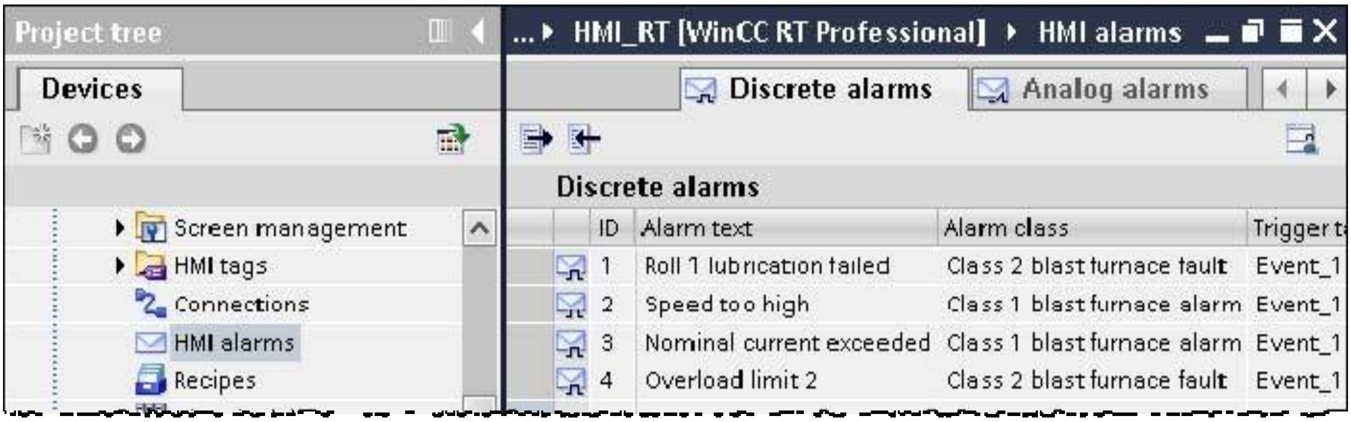
| Name | Tag table | Data type | Connection |
|---|---|---|---|
| Event_1 | I_message_events | UInt | <Internal... |

| ID | Alarm text | Alarm class | Trigge |
|---|---|---|---|
| 1 | Roll 1 lubrication failed | Class 2 blast furnace fault | Even |
| 2 | Speed too high | Class 1 blast furnace alarm | Even |
| 3 | Nominal current exceeded | Class 1 blast furnace alarm | Even |
| 4 | Overload limit 2 | Class 2 blast furnace fault | Even |

### 2. Create four new discrete alarms

Create four new discrete alarms using "Add" in the HMI alarms editor.

>> Project tree > HMI device > HMI alarms > tab: Discrete alarms



Discrete alarms

| ID | Alarm text | Alarm class | Trigger t |
|---|---|---|---|
| 1 | Roll 1 lubrication failed | Class 2 blast furnace fault | Event_1 |
| 2 | Speed too high | Class 1 blast furnace alarm | Event_1 |
| 3 | Nominal current exceeded | Class 1 blast furnace alarm | Event_1 |
| 4 | Overload limit 2 | Class 2 blast furnace fault | Event_1 |

### 3. Entering alarm texts

Use the alarm texts as shown in the figure above in the Alarm text column:

- Alarm 1: Roll 1 lubrication failed
- Alarm 2: Speed too high
- Alarm 3: Nominal current exceeded
- Alarm 4: Overload limit 2

**4. Assign alarm classes**

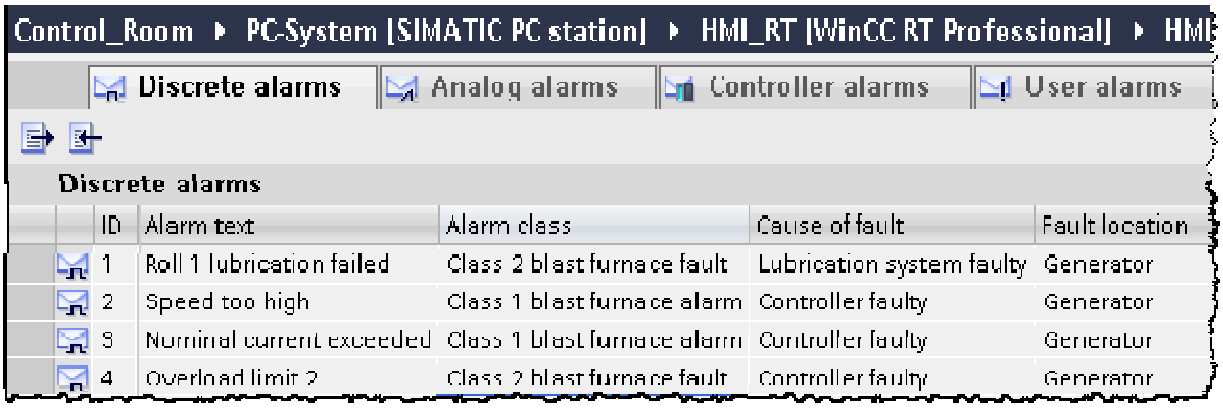
Assign the alarm classes to the alarms as shown in the figure above in the Alarm class column.

**5. Select trigger tag and trigger bit**

Select the trigger tag "Event_1" for all discrete alarms and use trigger bits 0-3 for the alarms with the IDs 1-4.

**6. Add cause of fault and fault location**

Expand the four discrete alarms with the information regarding the "Fault location" and "Cause of fault" as shown in the figure below.



Control_Room ▸ PC-System [SIMATIC PC station] ▸ HMI_RT [WinCC RT Professional] ▸ HMI

Discrete alarms

| ID | Alarm text | Alarm class | Cause of fault | Fault location |
|---|---|---|---|---|
| 1 | Roll 1 lubrication failed | Class 2 blast furnace fault | Lubrication system faulty | Generator |
| 2 | Speed too high | Class 1 blast furnace alarm | Controller faulty | Generator |
| 3 | Nominal current exceeded | Class 1 blast furnace alarm | Controller faulty | Generator |
| 4 | Overload limit 2 | Class 2 blast furnace fault | Controller faulty | Generator |

Don't forget to save your project.

## 7.8. Exercise 5: Configure the alarm view

**Procedure**

1. **Take the "Alarms"screen from the global library "Library_Pictures"and include it in the "Control_Room" project.**
2. **Insert the "WinCC alarm view" in the "Alarms" screen**
3. **Set the properties of the WinCC alarm view**
   - **General**
   - **Window**
   - **Table**
   - **Blocks**
   - **Columns**
   - **Toolbar**
   - **Status bar**
4. **Insert a screen navigation button for the "Alarms" screen in the "Start" screen**

### WinCC alarm view parameters:

| | |
|---|---|
| **General** | Display:<br>- Type: "Pending alarms"<br>- Display: "All alarms"<br>timebase: Adopt "device settings |
| **Window** | Settings title: "Blast furnace-W." |
| **Table** | |
| **Blocks** | Apply project settings |
| **Columns** | |
| **Toolbar** | General:<br>- Display toolbar / tooltips<br>- Alignment top<br>Buttons:<br>- Enable all buttons |
| **Status bar** | General: Display status bar / tooltips<br>Elements: Display all elements |

- Header
  - Column headers "Show column headers" "Sizeable"
  - Row headers "Show row headers"
- Selection Selection range
  - Type: "Single elements"
  - Border: "Line"
- Sorting settings:
  - Sorting: "Sort on double-click"
  - Sorting order: "Top/Bottom/None"
  - Show sorting icon, index and button

-Date
-Time
-Number
-Status
-Class
-Alarm text
-Fault location
-Cause of fault
-Process value: 1

### Task description

The project control room needs to be expanded by an alarm screen to display alarms. Current or logged alarms are shown in the alarm view. If necessary, alarms of all alarm classes are displayed in an alarm view. In the next steps, an alarm screen from a global library will be inserted in the Control_Room project. The alarm screen "Alarms" will have a WinCC alarm view added.

Alarms 1-4 will be triggered in Runtime mode by bits 0-3 of the alarm tag "Event_1".

### Procedure

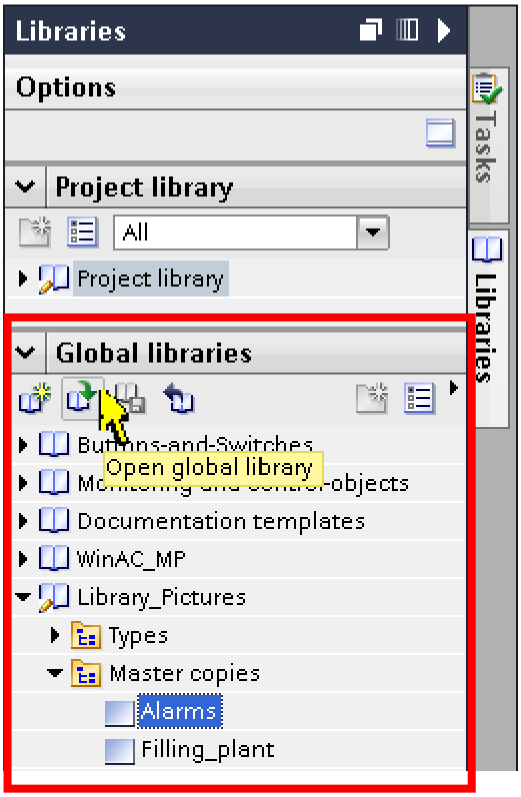
### 1. Insert the "Alarms" screen

You will find the "Alarms" screen in the global library "Library_Pictures". To import the screen, open this global library.

*Task card >> Libraries > Global libraries> Open global library*

The "Library_Pictures" library is in the folder: D:\Courses\WCCSUP\Libraries. Library files have the file extension "al11".

Drag the "Alarms" screen from the global library to the list of screens in your project. To do this, drag the "Alarms" screen from the global library to the Screens area of your project while holding down the left mouse button.



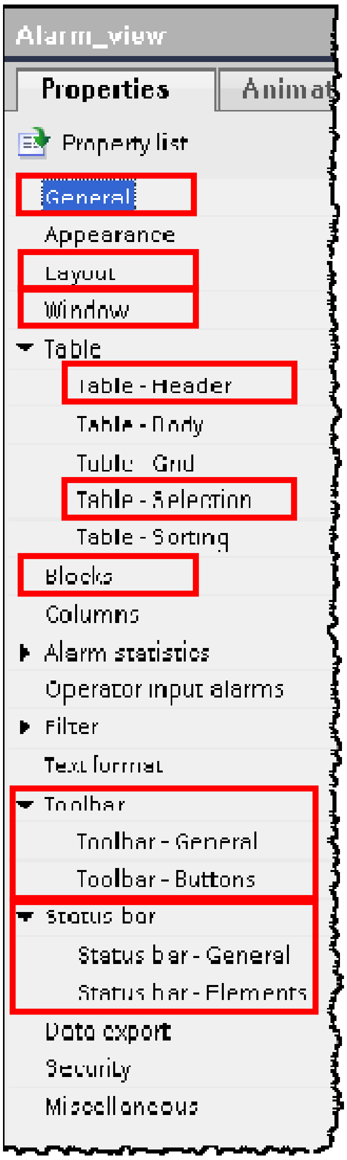
### 2. Insert the "WinCC alarm view" in the "Alarms" screen

From the "Toolbox" task card, insert the "Alarm view" object in the screen.

*Task card >> Toolbox > Controls > Alarm view*

### 3. Assign parameters for "WinCC alarm view" in the "Alarms" screen

You assign parameters for the WinCC alarm view in the Inspector window using the "Properties page":



Use the following settings for the WinCC alarm view (Properties page):

**General / Display:**
>> Inspector window > Properties tab+*Properties* > *General* > ***Display***
**- Type:** "Pending alarms"
**- Display:** "All alarms"

**General/Time base:**
>> Inspector window > Properties tab+*Properties* > *General* >
**Time base:** "accept device settings"

**Layout / Position & size:**
>> Inspector window > Properties tab+Properties > ***Layout*** > **Position & size** X: 50, Y: 110, width: 1180 and height: 510.

***Window / Settings***
*>> Inspector window > Properties tab+Properties > Window >* ***Settings***
- **Title style:** "Standard"
- **Title:** "Blast furnace-W."
- **Moveable, Closable:, Sizeable:** deactivated

**Table:**

*>> Inspector window > Properties tab+Properties > Table >*
***Table – Header*** >
**- Column headers:**
**Show column headers** activated
**Sizeable** activated
**- Row headers: Show row headers** activated

*>> Inspector window > Properties tab+Properties > Table >*
***Table – Selection*** > **Selection range**
- **Type:** "Single elements" (one alarm row can be selected)
- **Border:** "Line"

*>> Inspector window > Properties tab+Properties > Table >*
***Table – Sorting*** >
**Settings:**
**- Sorting:** "Sort on double-click"
**- Sorting order:** "Top/Bottom/None"
**- Show sorting icon, index and button:** Enable

**Blocks:**
>> Inspector window > Properties tab+*Properties* > **Blocks**
- **Blocks: Accept project settings** enabled

**Columns:**
>> Inspector window > Properties tab+*Properties* > **Columns**
- **Columns**: Enable all columns (Date, Time, Number, Status, Class, Alarm text, Fault location, Cause of fault, Process value: 1)

**Toolbar:**
>> Inspector window > Properties tab+Properties > **Toolbar** >
**General:**
**- Toolbar / Show tooltips**
**- Alignment top**

**Buttons:**
- Enable all buttons

**Status bar:**
>> Inspector window > Properties tab+Properties > **Status bar** >
**Status bar - General:**
**- Display status bar / Show tooltips** activated

**Status bar - Elements:**
**-** Show all elements

### 4. Insert a screen navigation button for the "Alarms" screen in the "Start" screen

Open the "Start" screen. Copy the button for selecting the "Motor" screen and paste the copy at position X: 571 and Y: 951. Change the button label from "Motor" to "Alarms".

Open the function list for the **Press left mouse button** event for the button. In "ActivateScreenInScreenWindow", change the "New screen name" parameter from "Motor" to "Alarms". Save your project.

## 7.9. Exercise 6: Function test WinCC alarm view

**Procedure**

1. **Check "alarm logging in runtime" in the Runtime settings**
2. **Call up WinCC Professional Runtime simulation**
3. **Trigger alarms using the buttons in the "Alarms" screen**
4. **Monitor the alarm status of the alarms**
5. **Acknowledge alarms**
6. **Call up various alarm list displays**
   **- "Alarm list"**
   **- "Short-term historical alarm list"**
   **- "Long-term historical alarm list"**

**Task description**

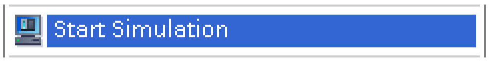
The created alarm screen needs to be tested.

**1. Check alarm logging in the Runtime settings**

Check alarm logging in the Runtime settings of the "HMI system" using "Services > Start sequence of WinCC > Alarm logging in runtime". Without this function, no alarms are displayed in the short-term or long-term historical data list in the WinCC alarm view.

**2. Test your project**

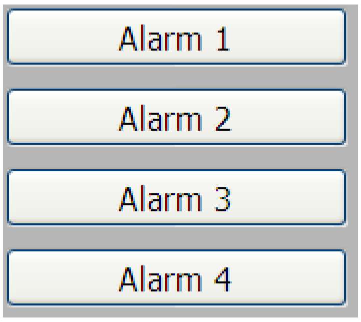
Call up Runtime simulation using the shortcut menu of "PC system" (right mouse button on "PC system"). You will find the "PC system" in: >> Project tree > Control_Room > PC system



When you start the simulation, the project is compiled and Runtime mode started.

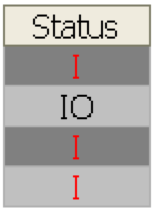
### 3. Trigger alarms

Trigger the alarms you have configured using the buttons (alarm 1 to 4) in the "Alarms" screen. The buttons have a toggle function to change over the bits that trigger the alarms in the "Event_1" tag.



### 4. Monitor the alarm status

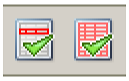
Watch the display in the "Status" column (Incoming, Outgoing...)



| Status |
|---|
| I |
| IO |
| I |
| I |

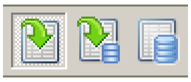
### 5. Acknowledging alarms

Acknowledge the alarms with the "Single acknowledgement" and "Group acknowledgement" buttons. Watch the display in the "Status" column (Incoming, Outgoing...)



### 6. Call up different alarm list types

Call up the various alarm list types:
- "Alarm list"
- "Historical data list short-term"
- "Historical data list long-term"



## 7.10. Exercise 7: Loop in alarm function

**Procedure**

1. **Open discrete alarms in the HMI Alarms editor**
2. **Link the "Loop-in alarm" event of "Bit_message_2" with the C function "Loop_in_Alarm"**
3. **Set parameters for the "Loop_in_Alarm" C function for the "Loop_in_Alarm" event**
4. **Test "Loop_in_Alarm" in Runtime mode**

**Task description**

The alarm system you have created needs to be expanded and tested with the "Loop-in alarm" function.
With the "Loop-in alarm" function, it is possible to open a predefined screen directly from an alarm row.
The "Loop_in_Alarm" C function required for this has already been created in the "Runtime scripting" section. You will find this user-defined script in Scripts > "C scripts"

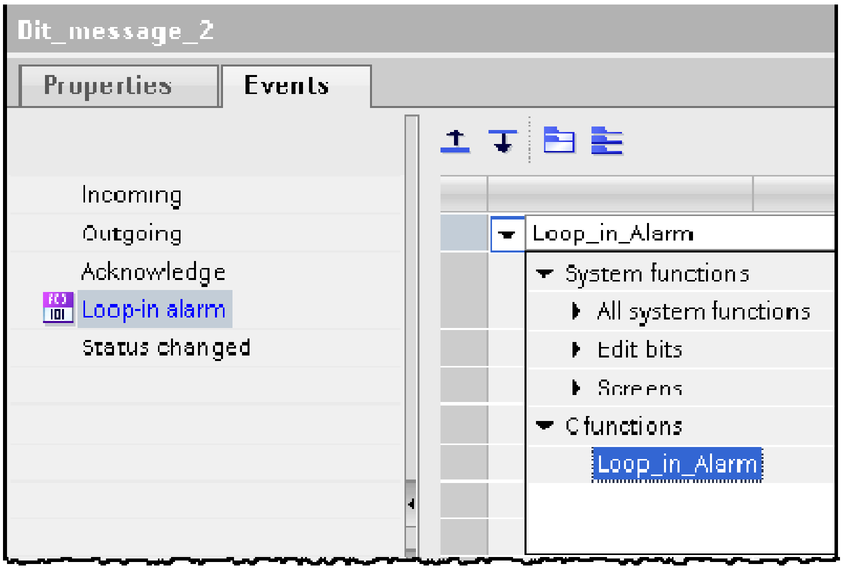
**1. Open discrete alarms in the HMI Alarms editor**

Open the discrete alarms in the "HMI Alarms" editor

>> Project tree > HMI device > HMI alarms > tab: Discrete alarms

### 2. Link the "Loop-in alarm" event of "Bit_message_2" with the C function "Loop_in_Alarm"

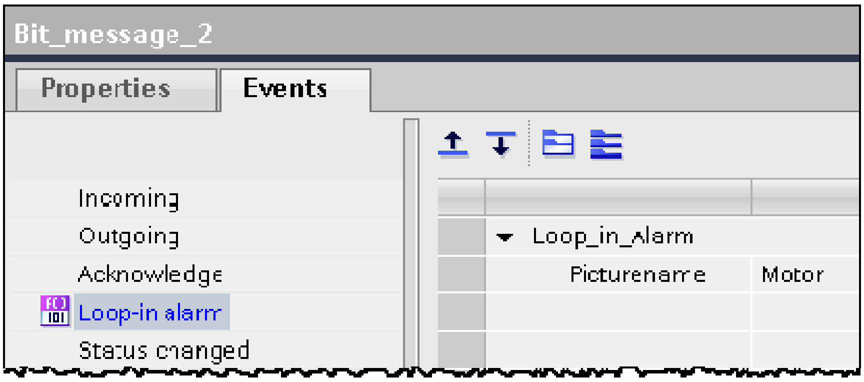
In the "HMI Alarms" editor, select the second alarm (Bit_message_2 / ID: 2) and link the "Loop-in alarm" event with the "Loop_in_Alarm" C function



### 3. Set the "Loop_in_Alarm" C function for the "Loop-in alarm" event

Select the screen name "Motor" for the "Picturename" parameter of the "Loop_in_Alarm" function. When the function is called, the "Motor" screen opens.

Don't forget to save your project.

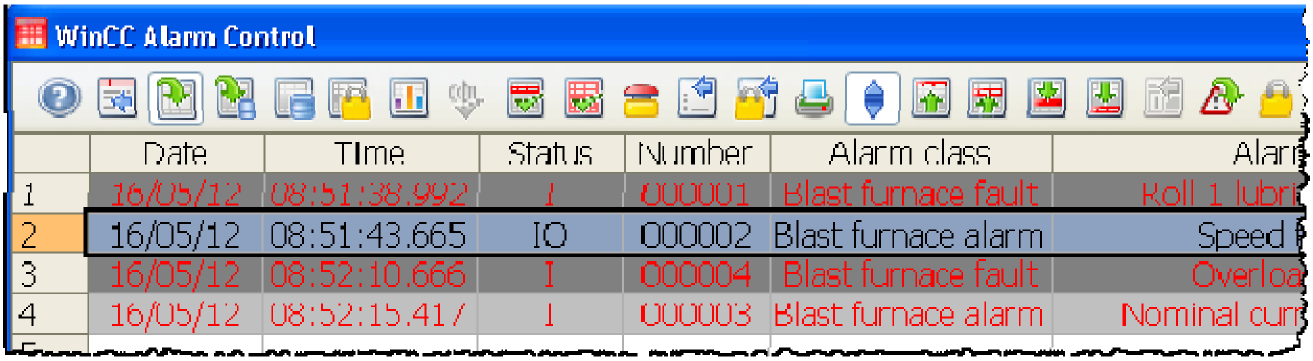


### 4. Test your project

Call-up Runtime simulation ("PC system"). Trigger the "Bit_message_1" to "Bit_message_4" using the "Alarm 1" to "Alarm 4" buttons.

Enable automatic scrolling so that you can select the displayed alarms in the alarm view with a mouse click.

Select the alarm with number "2" with a mouse click. The selected alarm is highlighted in color and displayed with a selection border.



WinCC Alarm Control

| | Date | Time | Status | Number | Alarm class | Alar |
|---|---|---|---|---|---|---|
| 1 | 16/05/12 | 08:51:38.992 | I | 000001 | Blast furnace fault | Roll 1 lubri |
| 2 | 16/05/12 | 08:51:43.665 | IO | 000002 | Blast furnace alarm | Speed |
| 3 | 16/05/12 | 08:52:10.666 | I | 000004 | Blast furnace fault | Overloa |
| 4 | 16/05/12 | 08:52:15.417 | I | 000003 | Blast furnace alarm | Nominal curr |

In addition to this, the "Loop-in-Alarm" button is enabled (change from gray to colored display).

To trigger the "Loop-in alarm" function, click the "Loop-in-Alarm" button. After clicking, the "Motor" screen is displayed.

## 7.11. Alarm groups

1. **An alarm group is a collection of individual alarms**

2. **Generating alarm groups from alarm classes:**
   **- with predefined alarm classes already available**
   **- with user-defined alarm classes automatic (system function)**

3. **Generating user-defined alarm groups:**
   **- generated by the user as necessary**
   **- can be structured hierarchically with up to 5 subgroups**

In a plant, there are many alarms from different areas and processes. To display the statuses of the plant areas clearly on the HMI device and to make the alarm system of your project easier to work with, alarm groups are used.

### Description

An alarm group is a collection of individual alarms. An alarm group has various tags, (status, lock tags, acknowledgement tags, display suppression tags) that each address a property of the alarms it contains. You specify the tags and their bits for a group. To display an alarm group on the HMI device, configure its visualization individually using the status tag of the alarm group.

### Usage

Use cases for alarm groups:

- Collect alarms with the same problem as the cause, for example "power failure"
- Sort and monitor alarms from a machine unit, for example "Drive XY fault"

- Process all alarms of a plant area together, for example
  - Acknowledge alarms
  - Lock alarms
  - Suppress the display of alarms

## Types of alarm groups

In WinCC, there are two types of alarm groups:

User-defined alarm groups:

User-defined alarm groups are created by the user as necessary. They contain alarms that require acknowledgement and, if necessary, subordinate alarm groups. User-defined alarm groups can, when necessary, be structured hierarchically with up to five subgroups.

Alarm groups from alarm classes

An alarm group has already been created for each predefined alarm class. When creating a new alarm class, an alarm group with the same name is generated by the system. All alarms of this alarm class are also included in the alarm group based on the alarm class. The changes to the configuration of the alarm classes are transferred to the "Alarm groups" tab. The configuration of the relevant alarm group is updated.

## Tags of an alarm group

When necessary, the following tags are assigned to an alarm group:

**Status tag:**

The status tag saves the alarm statuses of an alarm group.

The status tag saves the request for acknowledgement. The request for acknowledgement is pending in an alarm group when at least one of the alarms it contains is pending and requires acknowledgement. The status tag is used to visualize the alarm statuses and the request for acknowledgement of an alarm group. The status tag can be queried by other WinCC components.

**Lock tag:**

The lock tag saves the locked status of an alarm group.

The lock tag is used to evaluate the locked status of an alarm group.

**Acknowledgement tag:**

The acknowledgement tag saves the "Acknowledged" status of an alarm group.
The acknowledgement tag is used to acknowledge an alarm group. The acknowledgment tag can be queried by other WinCC components.

**Note**

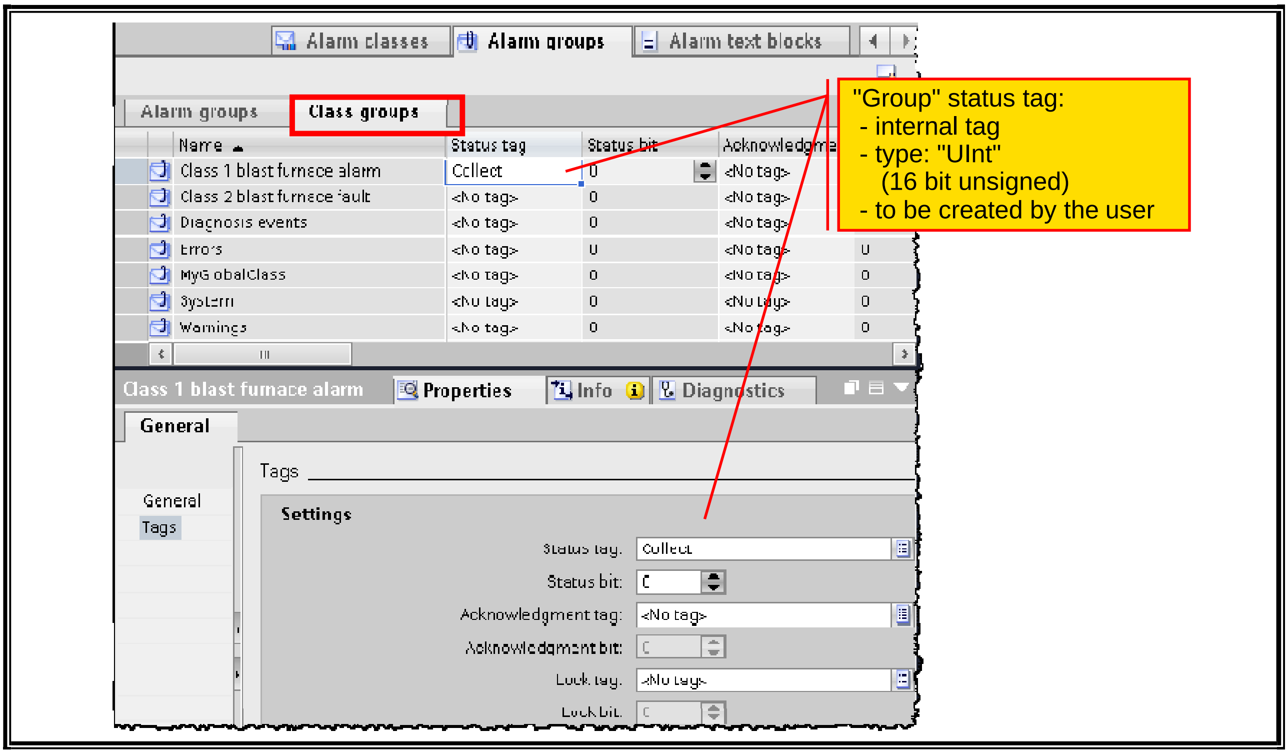
The acknowledgement bit of an acknowledgement tag is not automatically reset.

A button or function key with a system function is required to reset the acknowledgement bit.
The user then resets the acknowledgement bit of the acknowledgement tag.
The acknowledgement tag of an alarm group does not indicate the acknowledgement in the alarm view.

**Display suppression tag:**

The display suppression tag is used to automatically suppress the display of all alarms of an alarm group in a specific plant status.

The display suppression task returns the current value of the plant status as a value between 1 and 32. All alarms of the alarm group that have their corresponding bit number 1 to 32 set in the display suppression form are suppressed.



| Name | Status tag | Status bit | Acknowledgme |  |
|---|---|---|---|---|
| Class 1 blast furnece alarm | Collect | 0 | <No tag> |  |
| Class 2 blast furnece fault | <No tag> | 0 | <No tag> |  |
| Diagnosis events | <No tag> | 0 | <No tag> |  |
| Errors | <No tag> | 0 | <No tag> | 0 |
| MyGlobalClass | <No tag> | 0 | <No tag> | 0 |
| System | <No tag> | 0 | <No tag> | 0 |
| Warnings | <No tag> | 0 | <No tag> | 0 |

Status tag: Collect
Status bit: 0
Acknowledgment tag: <No tag>
Acknowledgment bit: 0
Lock tag: <No tag>
Lock bit: 0

"Group" status tag:
- internal tag
- type: "UInt"
  (16 bit unsigned)
- to be created by the user

Status tag

In this example, an internal tag "Group" of the data type "UInt" (unsigned 16 bit) is used as the status tag.

The two statuses ("Incoming / Outgoing") and the "Acknowledgement status" of the alarm group are stored in a status tag. Depending on the data type of the tag, up to 16 group alarms can be logged to a status tag. Each alarm group occupies 2 bits in the status tag.

The position of the bit for "activated/deactivated" states in the status tag is identified by the status bit. The position of the "acknowledgment bit" depends on the data type of the status tag.

The distance to the bit with the "Incoming / Outgoing" status is as follows:

- 4 bits with the "USInt" tag (8 bits unsigned)
- **8 bits** with the **"UInt"** tag (**16 bits** unsigned**)**
- 16 bits with the "UDInt" tag (32 bits unsigned)

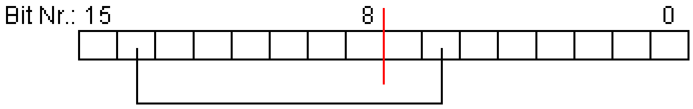
**Example**

Status tag **"Group"** of the data type "UInt" (16 bits unsigned)

If the status tag is of the data type "UInt" (16 bits unsigned) and if the **status bit = bit no. 6** was used for this group alarm,

- **Bit no. 6** of the status tag characterizes the status "Incoming / Outgoing" of this group alarm.
- **Bit no. 14** of the status tag indicates whether the group alarm contains an alarm requiring acknowledgment that has not yet been acknowledged.

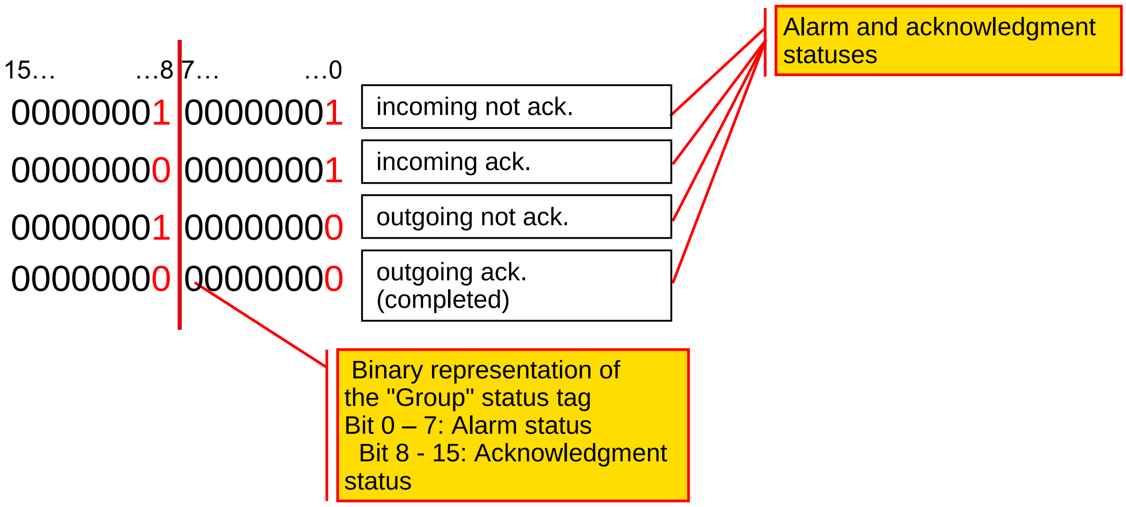
Bits "0-7" correspond accordingly to bits "8-15" of a "16-bit" status tag.



Binary representation of the "Group" status tag

Bit 0: 0 - the no alarm active, Bit 0: 1 - alarm pending

Bit 8: 0 - alarm is acknowledged, Bit 8: 1 - alarm is not acknowledged



| 15… …8 | 7… …0 | |
|---|---|---|
| 00000001 | 00000001 | incoming not ack. |
| 00000000 | 00000001 | incoming ack. |
| 00000001 | 00000000 | outgoing not ack. |
| 00000000 | 00000000 | outgoing ack. (completed) |

Alarm and acknowledgment statuses

Binary representation of the "Group" status tag Bit 0 – 7: Alarm status Bit 8 - 15: Acknowledgment status