Alarming

Types of alarms

There are 2 types of alarms in the D2000 system:

- **System alarms**: separate objects of Alarm type. These acquire a value according to the fulfillment of the condition for the occurrence/end of the alarm, or according to blocking and acknowledgment. If an alarm occurs, it is always stored in the logging database.
- **Process alarms**: objects of the I/O Tag, Eval Tag, or Switch types contain a tab for configuring alarms. For different types of values, it is possible to define multiple alarms depending on reaching the limits (for *Integer* and *Real* values) or on the value itself (for *Boolean* and *Quaternary input* val ues). Alarms cannot be defined for *Text*, *Absolute Time,* and *Relative Time* values. If an alarm occurs, it is possible to configure whether it will also be saved in the logging database.

User handling of alarms and graphic representation

The operator can do the following actions with alarms in the D2000 HI:

- **Acknowledge (confirm)**: the alarm is marked as acknowledged, or it disappears (if the acknowledgment was mandatory and the alarming condition has already disappeared).
- **Block**: the alarm is transferred to the blocked alarms. Blocking the alarm is also reflected in the configuration (the *Blocked alarm* flag is set in the configuration of the relevant object).
- **Unblock**: the blocked alarm is transferred to the active alarms
- **Suspend**: for a defined period, it will be transferred to blocked alarms, but only for the logged-in user - it will still appear as an active alarm to other users.

Alarms are displayed in the D2000 HI in the user window for alarm management in the tabs:

- **Active alarms**
- **Blocked alarms**
- **All alarms**

In addition, logical groups that are defined as *Alarm groups* are displayed in the user window for alarm management. Thus, individual alarms can be filtered according to the hierarchical tree of logical groups, while each alarm can also belong to several logical groups.

Status and transition process alarms

Process alarms can be:

- **status process alarms**: The alarm is active when the value of the object has a defined value (e.g. Boolean value - alarms PA_On, PA_Off) or the limit has a defined value (e.g. Integer value - alarms PA_HL, PA_VHL). This alarm can be configured to **require acknowledgment** - this means that the information about it will remain in the D2000 HI in the alarm management window even if it has already disappeared and the operator must confirm it.
  If the alarm does not require acknowledgment, it will disappear spontaneously if the condition for its occurrence (value, limit) disappears.
- **transition process alarms**: The alarm occurs when the value of the object changes to a defined value (e.g. Boolean value - alarms PA_ToOn, PA _ToOff) or when the limits change to a defined value (e.g. Integer value - alarms PA_ToHL, PA_ToVHL). This alarm **always requires acknowledgment** (if it were not, it would disappear immediately).

Criticality of alarms

Alarms are categorized by criticality:

- **non-critical** (less important)
- **critical** (more important)

The criticality of the system/process alarm is a configuration property, it is configured on the corresponding object. In the case of objects of Alarm type, this is a single setting, in the case of objects of the I/O Tag, Eval Tag, or Switch types, the alarm type can be configured for each defined process alarm.

Alarms in expressions and in the script

The value of objects of the I/O Tag, Eval Tag, or Switch types has attributes related to alarms. These attributes can be used within expressions in eval tags, scripts, and calculated historical values.

- Active process alarm (\ALF)
- Process alarm type (\ALV)
- Acknowledged process alarm (\ALQ)
- Process alarm value assignment time (\ALT)
- Indication of active critical alarm (\ALC)

Objects of Alarm type obtain the following values usable in expressions:

- @Normal
- @Alarm
- @Kvitt
- @Block
- @NoKvit

The number of active alarms in the system is given by the system variable ProcAlarmsNr.

The Signal_Trigger system variable generates a True pulse for alarms that have the "Raise SIGNAL" parameter enabled.

Note: also system user variables SystemError and SystemWarning with values of type text can be considered as "alarms" - their value has the critical alarm flag, the alarm type is *SysPrAl* and they are displayed in the user window for alarm management. These alarms are set by various processes of the D2000 system when multiple types of system errors occur (incorrect object configuration, duplicate address of the I/O tags, insufficient disk space). These are alarms intended for the system administrator.