

Configuring External Alarms

Remote Site Management

DESCRIPTION

PRELIMINARY

Copyright

© Ericsson AB 2012. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
2	Configuring External Alarms Using the ESC Manager	3
3	Configuring External Alarms Using the ESC CLI	5
3.1	Configuration Example 1	5
3.2	Configuration Example 2	7
4	Additional Information	11
4.1	MO Class Descriptions	11
4.2	Configuration Work Flow	12
4.3	External Alarm Signal and portId Pairing	12
	Reference List	13



PRELIMINARY



1 Introduction

This document describes how to configure Remote Site Management (RSM) nodes for monitoring external alarm circuits. For more information on the feature see *Feature ESC Measurements*, Reference [1].

PRELIMINARY



PRELIMINARY



2 Configuring External Alarms Using the ESC Manager

In the ESC Manager, do the following to configure monitoring of an external alarm circuit:

1. In the Site Management panel, right click on an ESC in the domain tree and select **Create > Alarm port....**
2. In the **Create new Alarm port** dialog box, set the alarm port attributes according to site-specific instructions and according to the information in Section 4 on page 11. (This step automatically creates an **Alarm entry** instance as a child to the **Alarm port** instance.)
3. Do the following to set the attributes of the **Alarm entry** instance:
 - Click on the **plus** button to expand the Alarm port tree
 - Click on the **Alarm entry** instance.
 - Set relevant attributes in the **Component** panel.



PRELIMINARY



3 Configuring External Alarms Using the ESC CLI

This example describes how to configure burglar alarms through the Command Line Interface (CLI) of an ESC. The CLI is accessible through port 22 on the IP address of the ESC, using an SSH client.

All commands that are given are described in *Command Description*, Reference [3]. MO classes are described in Section 4 on page 11.

3.1 Configuration Example 1

This example shows the front door switch in a Battery Backup System (BBS) 6101 connected to an ESC through an OVP.

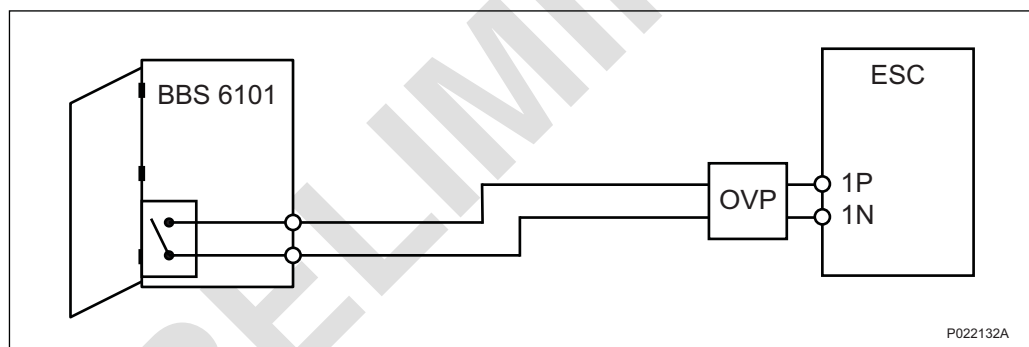


Figure 1 BBS 6101 Door Switch Connected to ESC through OVP

3.1.1 Example Work Flow

In the example text shown in next section, the following commands are given:

- Starting a transaction, `ea1`
- Creating the instance `AlarmPort=Door` (MO-DN `SDN=0,AlarmPort=Door.`)

Note: This operation automatically creates instance `ACTIVE` of MO class `AlarmEntry`, as a child to instance `AlarmPort=Door`. (MO-DN `SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE.`)

- Setting two attributes to the instance `AlarmPort=Door`:
 - `portId=1`
 - `inputActive=activeWhenClosed`
- Displaying all attributes (`getMOAttribute`) for MO instance `SDN=0,AlarmPort=Door`



- Setting five attributes to instance AlarmEntry=ACTIVE
- Committing the transaction ea1
- Ending the transaction ea1
- Displaying all attributes (getMOAttribute) for MO instance SDN=0,AlarmPort=Door,AlarmEntry=active

3.1.2

Example CLI Text

```
> startTransaction ea1
OperationSucceeded
> createMO ea1 SDN=0,AlarmPort=Door
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Door portId=1
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Door\
inputActive=activeWhenClosed
OperationSucceeded
> getMOAttribute ea1 SDN=0,AlarmPort=Door
SDN=0,AlarmPort=Door;
operationalState=disabled;
availabilityStatus=noStatus;
depEquipment="";
portId=1;
activeExternalAlarm=false;
inputActive=activeWhenClosed;
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE \
eventType=equipmentAlarm
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE \
perceivedSeverity=minor
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE \
probableCause=x733EquipmentMalfunction
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE \
specificProblem="door open"
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE \
additionalText="BBS 6201 Front Door"
OperationSucceeded
> commit ea1
OperationSucceeded
> endTransaction ea1
OperationSucceeded
> getMOAttribute SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE
SDN=0,AlarmPort=Door,AlarmEntry=ACTIVE;
eventType=equipmentAlarm;
perceivedSeverity=minor;
probableCause=x733EquipmentMalfunction;
```



```
specificProblem="door open";
additionalText="BBS 6201 Front Door";
filterAlgorithm=none;
filterTime=1;
OperationSucceeded
>
```

3.2 Configuration Example 2

This example shows a smoke detector, KEA 550 06/2, connected to an ESC. The first connector pair, 1P and 1N, is connected to the alarm relay contact in the smoke detector. The second connector pair, 2P and 2N, is connected to a relay contact that monitors the operational function of the smoke detector.

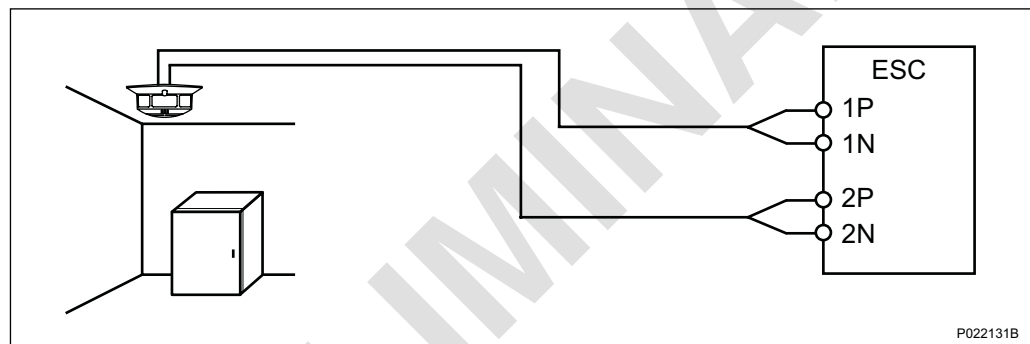


Figure 2 KEA 550 06/2 Smoke Detector Connected to ESC

3.2.1 Example Work Flow

In the example text shown in next section, the following commands are given:

- Starting a transaction, ea1
- Creating the instance AlarmPort=DetectorFault, (MO-DN SDN=0, AlarmPort=DetectorFault)

Note: This operation automatically creates instance ACTIVE of MO class AlarmEntry, as a child to instance AlarmPort=DetectorFault. (MO-DN SDN=0, AlarmPort=DetectorFault, AlarmEntry=ACTIVE.)

- Setting two attributes:
 - portId=2
 - inputActive=activeWhenOpen
- Creating the instance AlarmPort=Fire, MO-DN SDN=0, AlarmPort=Fire

Note: This operation automatically creates instance ACTIVE of MO class AlarmEntry, as a child to instance AlarmPort=Fire. (MO-DN SDN=0, AlarmPort=Fire, AlarmEntry=ACTIVE.)



- Setting two attributes:
 - portId=1
 - inputActive=activeWhenOpen
- Setting seven attributes to the alarm entry of MO class instance AlarmPort=DetectorFault (MO-DNSDN=0,AlarmPort=DetectorFault,AlarmEntry=ACTIVE):
 - eventType=equipmentAlarm
 - perceivedSeverity=normal
 - probableCause=m3100FireDetectorFailure
 - specificProblem=FireDetectorFailure
 - additionalText="Fire detector failure"
 - filterAlgorithm=time
 - filterTime=3
- Setting seven attributes to the alarm entry of MO class instance AlarmPort=Fire (MO-DNSDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE)
- Committing the transaction ea1
- Ending the transaction ea1
- Displaying all attributes (getMOAttribute) for MO instance SDN=0,AlarmPort=DetectorFault
- Displaying all attributes (getMOAttribute) for MO instance SDN=0,AlarmPort=Fire
- Displaying all attributes (getMOAttribute) for MO instance SDN=0,AlarmPort=DetectorFault,AlarmEntry=ACTIVE
- Displaying all attributes (getMOAttribute) for MO instance SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE

3.2.2

Example CLI Text

```
> startTransaction ea1
OperationSucceeded
> createMO ea1 SDN=0,AlarmPort=DetectorFault
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault portId=2
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault \
inputActive=activeWhenOpen
OperationSucceeded
> createMO ea1 SDN=0,AlarmPort=Fire
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire portId=1
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire \
inputActive=activeWhenOpen
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault, \
AlarmEntry=ACTIVE eventType=equipmentAlarm
OperationSucceeded
```



```

> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault,\
AlarmEntry=ACTIVE perceivedSeverity=normal
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault,\
AlarmEntry=ACTIVE probableCause=m3100FireDetectorFailure
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault,\
AlarmEntry=ACTIVE specificProblem="FireDetectorFailure"
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault,\
AlarmEntry=ACTIVE additionalText="Fire detector failure"
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault,\
AlarmEntry=ACTIVE filterAlgorithm=time
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=DetectorFault,\
AlarmEntry=ACTIVE filterTime=3
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE \
eventType=environmentalAlarm
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE \
perceivedSeverity=critical
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE \
probableCause=m3100Fire
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE \
specificProblem="Fire"
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE \
additionalText="Fire detected"
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE \
filterAlgorithm=time
OperationSucceeded
> setMOAttribute ea1 SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE \
filterTime=3
OperationSucceeded
> commit ea1
OperationSucceeded
> endTransaction ea1
OperationSucceeded
> getMOAttribute SDN=0,AlarmPort=DetectorFault
SDN=0,AlarmPort=DetectorFault;
operationalState=disabled;
availabilityStatus=noStatus;
depEquipment="";
portId=2;
activeExternalAlarm=false;
inputActive=activeWhenOpen;

```



```
OperationSucceeded
> getMOAttribute SDN=0,AlarmPort=Fire
SDN=0,AlarmPort=Fire;
operationalState=disabled;
availabilityStatus=noStatus;
depEquipment="";
portId=1;
activeExternalAlarm=false;
inputActive=activeWhenOpen;
OperationSucceeded
> getMOAttribute SDN=0,AlarmPort=DetectorFault,\
AlarmEntry=ACTIVE
SDN=0,AlarmPort=DetectorFault,AlarmEntry=ACTIVE;
eventType=equipmentAlarm;
perceivedSeverity=minor;
probableCause=m3100FireDetectorFailure;
specificProblem="FireDetectorFailure";
additionalText="Fire Detector failure";
filterAlgorithm=time;
filterTime=3;
OperationSucceeded
> getMOAttribute SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE
SDN=0,AlarmPort=Fire,AlarmEntry=ACTIVE;
eventType=environmentalAlarm;
perceivedSeverity=critical;
probableCause=m3100Fire;
specificProblem="Fire";
additionalText="Fire detected";
filterAlgorithm=time;
filterTime=3;
OperationSucceeded
>
```



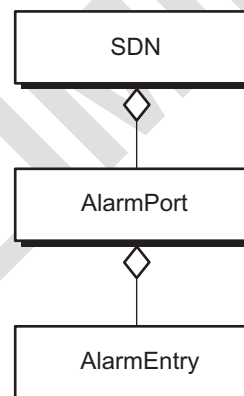
4 Additional Information

This section provides additional information about RSM node configuration.

4.1 MO Class Descriptions

This section describes the MO classes that are used when configuring external alarms. For a full list of MO Class attributes and their detailed descriptions, refer to document *MO Class Descriptions*, Reference [2].

MO classes `AlarmPort` and `AlarmEntry` are used to configure the RSM node for monitoring external alarm circuits. Figure 3 shows the containment hierarchy of the MO classes.



P023150A

Figure 3 MO Class Containment Hierarchy

MO class `SDN` represents the basic settings and ID attributes of the RSM node. (In the ESC Manager, the `SDN` MO class is represented by an ESC object in the domain tree of the Site Management panel.)

The main purposes of MO class `AlarmPort` are to dedicate a hardware port to the MO class instance (using attribute `portId` as described in Section 4.3 on page 12) and to monitor the state of the external alarm circuit. Attribute `inputActive` defines whether and opened or closed external alarm circuit will raise the alarm. Attribute `activeExternalAlarm` indicates if the alarm is raised or not.

MO class `AlarmEntry` represents attributes of the alarm entries that can be assigned to `AlarmPort` MO class instances. The main purpose of MO class `AlarmEntry` is to describe an alarm event. Another important feature is the



alarm filter function of attribute `filterAlgorithm`, which permits filtering of unstable or flickering alarm signals.

4.2 Configuration Work Flow

To configure the RSM node for monitoring external alarm circuits, follow this general work flow:

1. Create an `AlarmPort` MO class instance and couple it to an external alarm port, using the `portId` attribute.
2. Set the `inputActive` attribute.
3. Set relevant attributes of the `AlarmEntry` MO class instance that is automatically created as a child to the `AlarmPort` MO class instance.

4.3 External Alarm Signal and portId Pairing

The `portId` attribute of MO class `AlarmPort` designates a hardware port to the MO class instance. Information about pairing between `portId` values and hardware ports of the ESC and the EPP can be found in *Ericsson Site Controller Description*, Reference [4].



Reference List

- [1] *Feature ESC Measurements*, 25/1553 - LZA 701 0003
- [2] *MO Class Descriptions*, 32/1543 - LZA 701 0003
- [3] *Command Description*, 22/1551 - LZA 701 0003
- [4] *Ericsson Site Controller Description*, 46/1531-LZA 701 0003

PRELIMINARY