IDM UID
AG42FC

VERSION CREATED ON / VERSION / STATUS
20 Jun 2013 / 1.4 / Signed

EXTERNAL REFERENCE

Report

CODAC Alarm Automated Actions - Test Plan

Test Plan of CODAC Alarm Automated Actions

Approval Process			
	Name	*Action*	*Affiliation*
Author	**Utzel N.**	**20-Jun-2013:signed**	**IO/DG/DIP/CHD/CSD/CDC**
Co-Authors			
Reviewers			
Approver			
Document Security: level 1 (IO unclassified) *RO: Di Maio Franck*			
Read Access	**RO, project administrator, LG: SOPRA extra, AD: ITER, AD: External Collaborators, AD: Division - Control System Division, AD: Section - CODAC, AD: Auditors, AD: ITER Management Assessor**		

Change Log				
Title (Uid)	***Version***	***Latest Status***	***Issue Date***	***Description of Change***
CODAC Alarm Automated Actions - Test Plan (AG42FC_v1_4)	v1.4	Signed	20 Jun 2013	Check for SEVERE messages in the log files
CODAC Alarm Automated Actions - Test Plan (AG42FC_v1_3)	v1.3	Signed	04 Jun 2013	Updated SVN demo unit
CODAC Alarm Automated Actions - Test Plan (AG42FC_v1_2)	v1.2	Signed	28 Sep 2012	Change due to the last version of ClientModel that does not notify clients when the alarm configuration has slightly changed. As a consequence, saving a new configuration does not imply the execution of an automated action
CODAC Alarm Automated Actions - Test Plan (AG42FC_v1_1)	v1.1	Signed	21 Sep 2012	2 additional test cases: Alarm Server Configuration Reload and Flurry of Alarms
CODAC Alarm Automated Actions - Test Plan (AG42FC_v1_0)	v1.0	Signed	07 Sep 2012	

CODAC Alarm Automated Actions

Software Test Plan (STP)
Based on QA Template Version <1.0>

This document describes the tests that should be performed for CODAC Alarm Automated Actions – Alarm Notifier - in order to be installed as part of Core System release. Different test cases are described, as well as and test pass-fail criteria.

Contents

1 Introduction

1.1 Purpose

This document describes the tests that should be performed for CSS BEAST - Best Ever Alarm System Toolkit – Notifier in order to be installed as part of CODAC Core System. These tests will ultimately compare the capabilities of BEAST Automated Actions against these described in CODAC System Requirement (SRD) Document [IDM 28C2HL].

Particular functions to be tested are the automated actions configuration and their execution when the configured alarm state changes. The main components of the system as shown on Figure 1-I - BEAST Automated Actions Architecture.

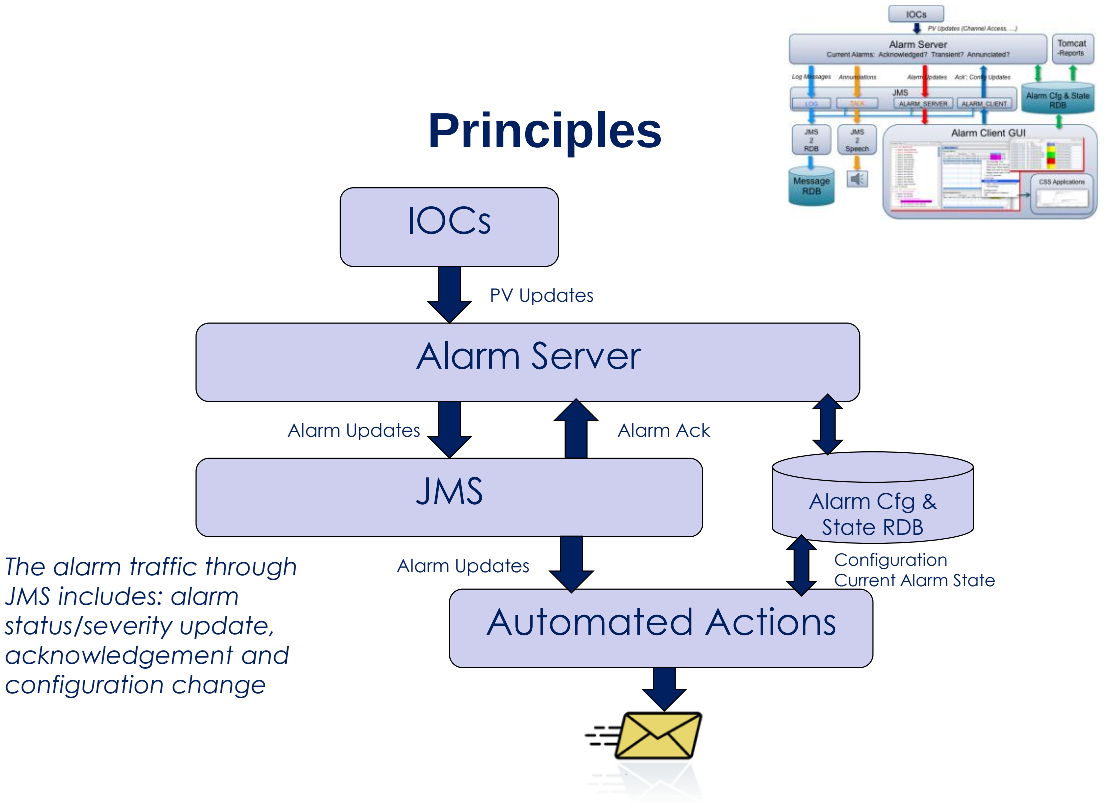


Figure 1-I - BEAST Automated Actions Architecture

The configuration of an automated action has three parts:

- Title: The title to be displayed as a context menu item. The max length of title is 30 characters.
- Delay: Delay in seconds before the notification is performed. If the alarm is acknowledged or clears before this delay, no action will be performed.
- Details: Details of the automated action to be executed. The exact format of the notification detail depends on the type of notification:
    - Email Notifications: start with "mailto:", using the RFC 6068 URL schema. Examples:
        - mailto:user@my.org
        - mailto:list1@example.com,list2@example.fr

        - mailto:user1@my.org?cc=support@my.org&bcc=john.doe@my.com
        - mailto:user@my.org?cc=rf@my.org&subject=*RF Source 1 in error&body={0} Alarm raised - Water below {1} m3
        - Within the configuration, the text {0} will be replaced with the alarm severity and {1} with the value that triggered the alarm.
    - External Command Notifications: start with "cmd:". Examples:
        - cmd:caput SomePV 1
        - cmd:my_script *

1.2 Scope

The test items are:

- The operational version of BEAST,
- The data, including all the configuration data needed to run the alarm system,
- The documentation, including the online help and the release notes.

The installation and uninstallation of the components are not part of this test plan.

1.3 System/Software overview and key features

Best Ever Alarm System Toolkit (BEAST) - is a distributed alarm system consisting of:

- Alarm Server that monitors alarm triggers in the control system,
- Alarm Notifier for Automated Actions,
- Relational Database for configuration and logging,
- CSS user interface for viewing current alarms as a table or hierarchical tree.

1.4 References

CODAC Quality assurance plan - https://user.iter.org/?uid=6J7RW4

2 DETAILS OF THE TESTING PROCESS

2.1 Definition of test levels

The described component tests will focus on the desired features of CODAC Alarm Automated Actions.

2.2 Test administration

2.2.1 Anomaly resolution and reporting

Anomaly Reports shall be submitted in Bugzilla.

2.2.2 Test reporting requirements

The test logs shall be generated to record the outcome of test procedures as described in section *.4 and *.5 of the level test plans.

2.2.3 Test deliverables

The test deliverables include:

- Component Test Logs / Reports
- Anomaly Reports with Bugzilla bug references.

Test input data are registered in SVN code repository.

No other test tool is needed.

The test reports may be submitted on ITER IDM.

3 COMPONENT TEST PLAN

3.1 Scope

3.1.1 Test items and their identifiers

CODAC Alarm Automated Actions includes the following unit:

- m-css with the following product:
  - org.csstudio.iter.alarm.beast.notifier.product/

3.1.2 Features to be tested

The main CODAC Alarm Automated Actions features to be tested are:

- Automatic email sending when configured alarm state changes,
- Automatic command execution.

3.1.3 Features not to be tested

Some basic external commands like caput are part of the test plan but the execution of external scripts or the SMS calls are excluded for now.

3.2 Approach

3.2.1 Testing Methods

The overall approach for the level of testing is the Black box method to test the functionality of CODAC Alarm Automated Actions.

3.2.2 Item pass/fail criteria

Each major anomaly found determines whether each test item has passed or failed testing.

3.3 Environment / Infrastructure

Core System in its development role version should be installed on a CODAC standard machine. Access to SVN is required.

3.4 Component Test Procedures

	3.4.1 CFG01 - Alarm Configuration Import	

Prerequisite	In a Linux console, create a working directory, download and start a demo IOC:
	0. $ rm -Rf ~/.css
	1.$ mkdir test
	2.$ cd test
	3.$ svn co https://svnpub.iter.org/codac/iter/codac/dev/units/m-css/trunk/applications/plugins/org.csstudio.alarm.beast.notifier/demo/
	`Checked out revision xxx.`
	`A    demo/ca.ini`
	`A    demo/demo.xml`
	`A    demo/demo.db`
	`Checked out revision xxx.`
	4.$ cd demo
	5.$ softIoc -s -d demo.db
	`Starting iocInit`
	`############################################################################`
	`## EPICS R3.14.12.2 $Date: Mon 2011-12-12 14:09:32 -0600$`
	`## EPICS Base built May 30 2012`
	`############################################################################`
	`iocRun: All initialization complete`
	`epics>`	

Test Cases	1. Positive confirmation of the alarm configuration loaded	
Procedure	In a new Linux console, import the alarm configuration for a “demo” alarm server: 1.$ cd test/demo 2. Edit the configuration file demo.xml $ gedit demo.xml& And replace “your_email@site.org” with your email address. Three occurrences should be replaced. Save the new configuration file 3.$ alarm-configtool -root demo -import -file demo.xml	

Pass Criteria	The output of the command should be:
	`Alarm Config Tool <current Core System version>`
	`Reading RDB configuration of 'demo'`
	`Deleting existing RDB configuration for 'demo'`
	`Importing configuration 'demo' from demo.xml`
	`Loading /demo/RFQ`
	`Loading /demo/RFQ/Vacuum`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control/Pump 2 Control`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control/Pump 2`
	`Control/RFQ_Vac:Pump2:Start_State`	

```
Loading /demo/RFQ/Vacuum/Pump 1 Control/Pump 2 Control/RFQ_Vac:Pump2:Pressure
Loading /demo/RFQ/Vacuum/Pump 1 Control/RFQ_Vac:Pump1:Start_State
Loading /demo/RFQ/Vacuum/Pump 1 Control/RFQ_Vac:Pump1:Pressure
```

3.4.2 SRV01 - Alarm Server and Notifier Startup

Prerequisite

1. Demo IOC running
2. Alarm Configuration Imported successfully

Test Cases

1. Positive confirmation of the demo alarm server and notifier started

Procedure

In the previous Linux console, start the "demo" alarm server and notifier:

1.$ alarm-server -root demo &

2.$ alarm-notifier -root demo -pluginCustomization ca.ini &

Pass Criteria

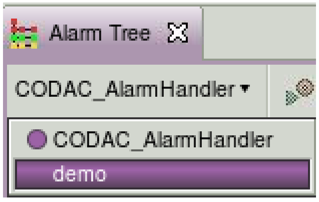
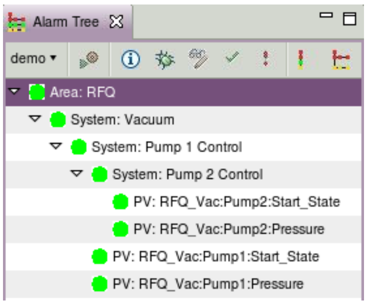
1. The output of the command should be:

```
INFO [Thread 10] org.csstudio.alarm.beast.server.Application (start) - Alarm
Server <version> started for 'demo' configuration
Alarm Server <version>
Configuration Root: demo
JMS Server Topic:   demo_SERVER
JMS Client Topic:   demo_CLIENT
JMS Talk Topic:     demo_TALK
JMS Global Topic:   GLOBAL_SERVER
INFO [Thread 11] org.apache.activemq.transport.failover.FailoverTransport
(doReconnect) - Successfully connected to tcp://localhost:61616
Read 4 PVs in x.xx seconds: xx.xx PVs/sec
<… many Info messages …>
```

2. The output of the command should be:

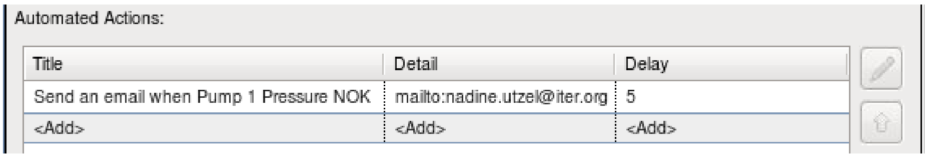
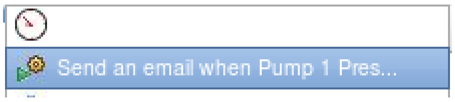
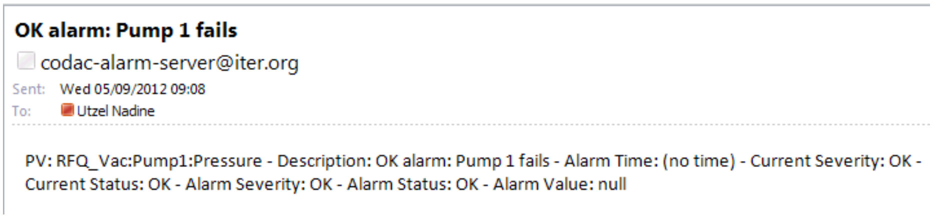
```
INFO [Thread 10] org.csstudio.alarm.beast.notifier.Application (start) - Alarm
Notification <version> started for 'demo' configuration
Alarm Notification <version>
Configuration Root: demo
JMS Server Topic:   demo_SERVER
JMS Client Topic:   demo_CLIENT
JMS Global Topic:   GLOBAL_SERVER
Notifier timer threshold: 3
Notifier thread threshold: 3
INFO [Thread 10] org.csstudio.alarm.beast.notifier.AlarmNotifier (start) -
Alarm Notifier started
<… many Info messages …>
```

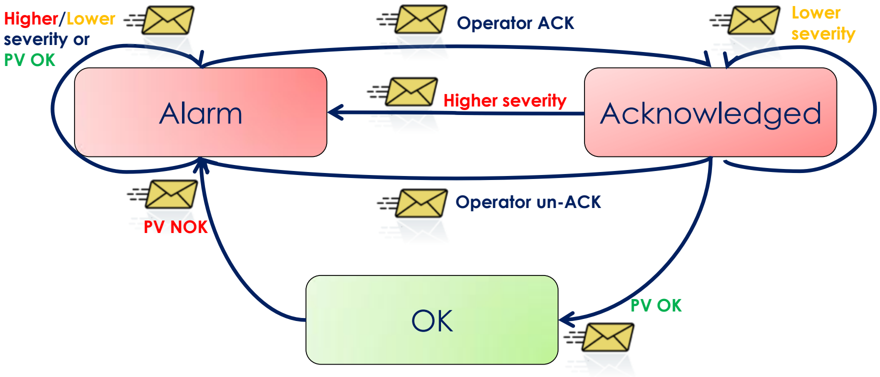
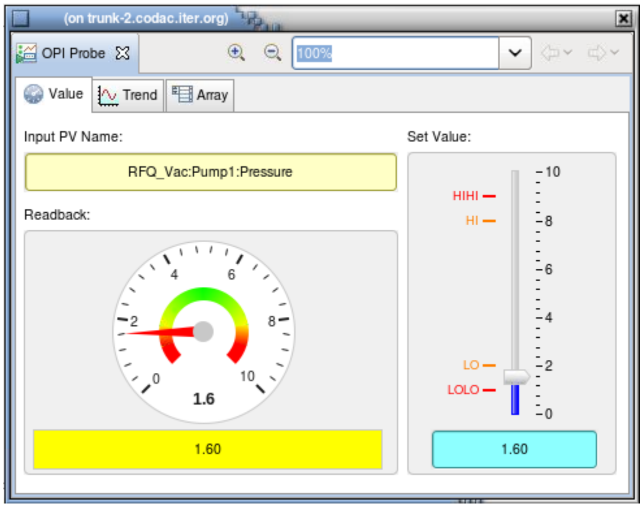
3.4.3 DSP01 - Alarm Notification

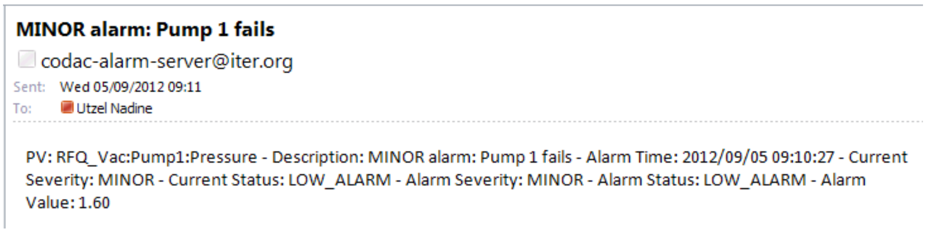
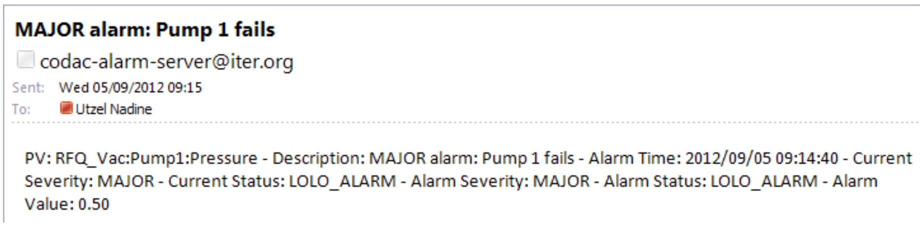
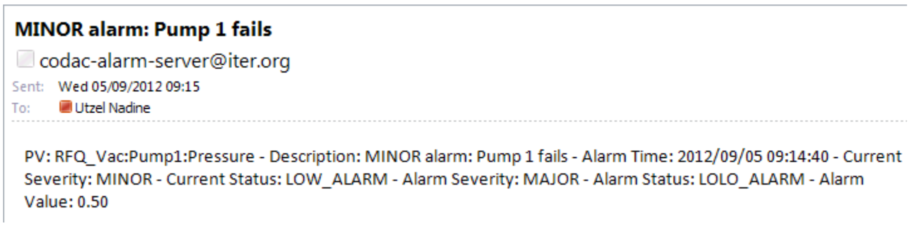
Prerequisite	1. Demo IOC running 2. Alarm Configuration imported 3. Alarm server and notifier started	
Test Cases	1. Positive confirmation of alarms triggered	
Procedure	In the previous Linux console, start the Operator Interface to monitor the alarms triggered: 1.$ css& 2. Browse to select the working directory test 3. Exit the Welcome screen by clicking on Workbench icon 3. Open the Alarm Perspective: menu Window -> Open Perspective -> Other… and select Alarm 4. Change the root element of the Alarm Tree View using the arrow near CODAC_AlarmHandler and select demo:  Alarm Tree CODAC_AlarmHandler ▾ CODAC_AlarmHandler demo	
Pass Criteria	1. The Alarm Tree should reflect the demo server structure and current alarms state:  Alarm Tree demo ▾ Area: RFQ System: Vacuum System: Pump 1 Control System: Pump 2 Control PV: RFQ_Vac:Pump2:Start_State PV: RFQ_Vac:Pump2:Pressure PV: RFQ_Vac:Pump1:Start_State PV: RFQ_Vac:Pump1:Pressure	

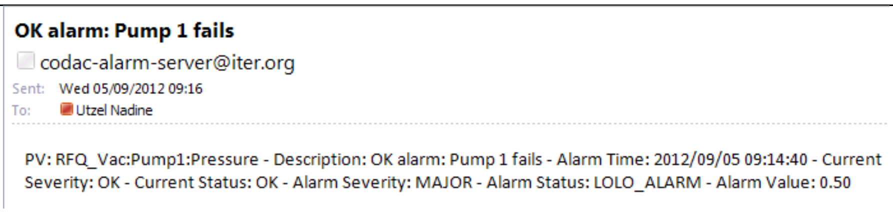
3.4.4 DSP02 - Automated Action Manual Configuration

Prerequisite	1. Demo IOC running 2. Alarm Configuration Imported successfully 3. Alarm server and notifier started 4. Alarm Operator Interface started	
Test Cases	1. Configure an alarm to executed automated actions when triggered	

Procedure	In CSS, log in as codac-dev to be able to modify dynamically the alarm configuration: 1. File -> Log in… 2. Enter codac-dev as User Name and type the Password. Click on OK to validate 3. In the Alarm Tree view, right-click on PV: “Area: RFQ” -> “System: Vacuum” -> “System: Pump 1” ->“PV: RFQ_Vac:Pump1:Pressure” and select the option Configure Item 4. In the Automated Actions section, add a Title “Send an email when Pump 1 Pressure NOK”, the Detail of the Automated Action “mailto:your_email@site.org” and a Delay of “5” seconds. Then click on OK to validate the new configuration of the alarm  Automated Actions: Title Detail Delay Send an email when Pump 1 Pressure NOK mailto:nadine.utzel@iter.org 5 <Add> <Add> <Add> 5. Test manually the automated action by right-clicking from the Alarm Tree on “PV: RFQ_Vac:Pump1:Pressure” and selecting the option “Send an email when Pump 1 Pres…”  Wait for few seconds and check your inbox for the test alarm email	
Pass Criteria	1. The email automatically sent when the configured alarm changes:  **OK alarm: Pump 1 fails** codac-alarm-server@iter.org Sent: Wed 05/09/2012 09:08 To: Utzel Nadine PV: RFQ_Vac:Pump1:Pressure - Description: OK alarm: Pump 1 fails - Alarm Time: (no time) - Current Severity: OK - Current Status: OK - Alarm Severity: OK - Alarm Status: OK - Alarm Value: null	
	3.4.5 SRV02 - PV Alarm State Change Notification	
Prerequisite	1. Demo IOC running 2. Alarm Configuration imported 3. Alarm server and notifier started 4. Alarm Operator Interface started	
Test Cases	1. Positive notification by email of all alarm state changes	

	When an automated action is triggered?  Higher/Lower severity or PV OK Operator ACK Lower severity Alarm Higher severity Acknowledged PV NOK Operator un-ACK OK PV OK	
Procedure	In CSS, when the alarm state of a PV changes, an email is sent. From the Alarm Tree View: 1. Select “Area: RFQ” -> “System: Vacuum” -> “System: Pump 1 Control”-> “PV: RFQ_Vac:Pump1: Pressure” 2. Right click on the item and select Process Variable -> OPI Probe A. Set the Value 1.6 to generate a MINOR LOW_ALARM  (on trunk-2.codac.iter.org) Input PV Name: RFQ_Vac:Pump1:Pressure Readback: 1.60 Set Value: 1.60 OPI Probe 100% Value Trend Array 1.6 0 2 4 6 8 10 HIHI HI LO LOLO Wait for ~5 seconds and check your inbox for a MINOR alarm email B. Set the Value 0.5 to generate a MAJOR LOLO_ALARM, wait for ~5 seconds and check your inbox for a MAJOR alarm email C. Set the Value 1.3 to generate a MINOR LOW_ALARM, wait for ~5 seconds and check your inbox for a MINOR alarm email D. Set the Value 3.2 to generate a OK Alarm, wait for ~5 seconds and check your inbox for a OK alarm email E. From the Alarm Tree, right-click on “PV: RFQ_Vac:Pump1: Pressure” and select Acknowledge. Wait for ~5 seconds and check your inbox for a OK alarm email F. Set the Value 1.9 to generate a MINOR LOW_ALARM, wait for ~5 seconds and	

	then acknowledge the alarm as point E. Wait for ~5 seconds and check your inbox for 2 x MINOR alarm emails G. Set the Value 0.8 to generate a MAJOR LOLO_ALARM, wait for ~5 seconds and check your inbox for a MAJOR alarm email. Then acknowledge the alarm as point E. Wait for ~5 seconds and check your inbox for a second MAJOR alarm email H. Set the Value 1.1 to generate a MINOR LOW_ALARM, wait for ~5 seconds and check your inbox for a MINOR alarm email I. From the Alarm Tree, right-click on “PV: RFQ_Vac:Pump1: Pressure” and select Un-Acknowledge. Wait for ~5 seconds and check your inbox for a MINOR alarm email J. Acknowledge the active alarm as point E, wait ~5 seconds and set the Value 2.7 to set the PV within its alarm limits. Check your inbox for 2 x notification emails K. Set the Value 1.1 to generate a MINOR LOW_ALARM, BUT before the 5 seconds, make the PV recover by setting the value to 4. Check that no notification email has been sent.	
Pass Criteria	A. Email notification when PV OK -> MINOR LOW_ALARM at 1.6  **MINOR alarm: Pump 1 fails** □ codac-alarm-server@iter.org Sent: Wed 05/09/2012 09:11 To: Utzel Nadine PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:10:27 - Current Severity: MINOR - Current Status: LOW_ALARM - Alarm Severity: MINOR - Alarm Status: LOW_ALARM - Alarm Value: 1.60 B. Email notification when PV MINOR LOW_ALARM -> MAJOR LOLO_ALARM at 0.5  **MAJOR alarm: Pump 1 fails** □ codac-alarm-server@iter.org Sent: Wed 05/09/2012 09:15 To: Utzel Nadine PV: RFQ_Vac:Pump1:Pressure - Description: MAJOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:14:40 - Current Severity: MAJOR - Current Status: LOLO_ALARM - Alarm Severity: MAJOR - Alarm Status: LOLO_ALARM - Alarm Value: 0.50 C. Email notification when PV MAJOR LOLO_ALARM at 0.5 -> MINOR LOW_ALARM (current severity/status and alarm severity/status are different)  **MINOR alarm: Pump 1 fails** □ codac-alarm-server@iter.org Sent: Wed 05/09/2012 09:15 To: Utzel Nadine PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:14:40 - Current Severity: MINOR - Current Status: LOW_ALARM - Alarm Severity: MAJOR - Alarm Status: LOLO_ALARM - Alarm Value: 0.50 D. Email notification when PV MAJOR LOLO_ALARM at 0.5 -> OK (alarm severity/status are still MAJOR/LOLO_ALARM as the alarm has not been acknowledged yet)	



OK alarm: Pump 1 fails

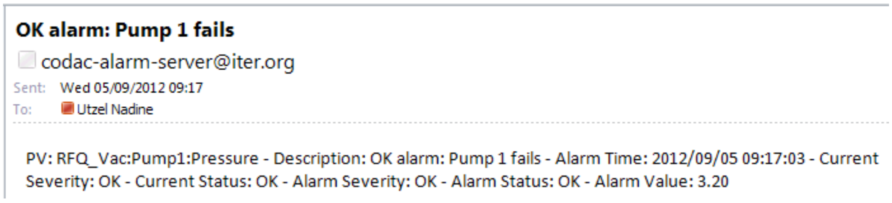
codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:16

To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: OK alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:14:40 - Current Severity: OK - Current Status: OK - Alarm Severity: MAJOR - Alarm Status: LOLO_ALARM - Alarm Value: 0.50

E. Email notification when PV OK at 3.2 has been acknowledged



OK alarm: Pump 1 fails

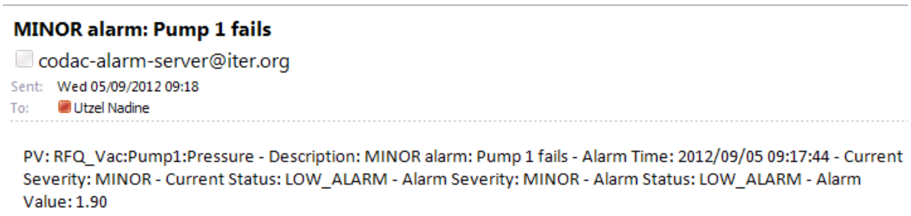
codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:17

To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: OK alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:17:03 - Current Severity: OK - Current Status: OK - Alarm Severity: OK - Alarm Status: OK - Alarm Value: 3.20

F. 2 x notification emails – one when the MINOR alarm is raised at 1.9 and one when this active alarm has been acknowledged (Alarm Severity: MINOR_ACK)



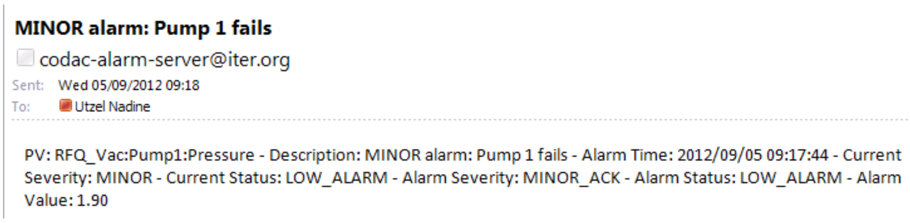
MINOR alarm: Pump 1 fails

codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:18

To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:17:44 - Current Severity: MINOR - Current Status: LOW_ALARM - Alarm Severity: MINOR - Alarm Status: LOW_ALARM - Alarm Value: 1.90



MINOR alarm: Pump 1 fails

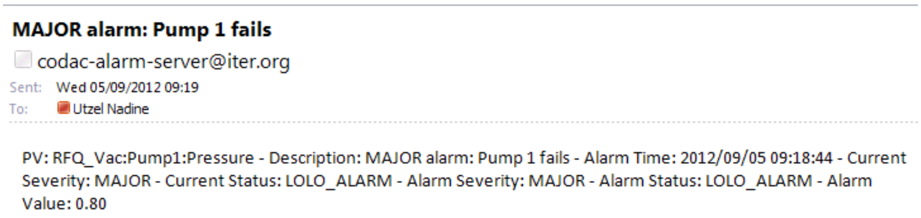
codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:18

To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:17:44 - Current Severity: MINOR - Current Status: LOW_ALARM - Alarm Severity: MINOR_ACK - Alarm Status: LOW_ALARM - Alarm Value: 1.90

G. Email notification when PV MINOR Acknowledged -> MAJOR active alarm at 0.8. As the severity is highest, the alarm becomes active again



MAJOR alarm: Pump 1 fails

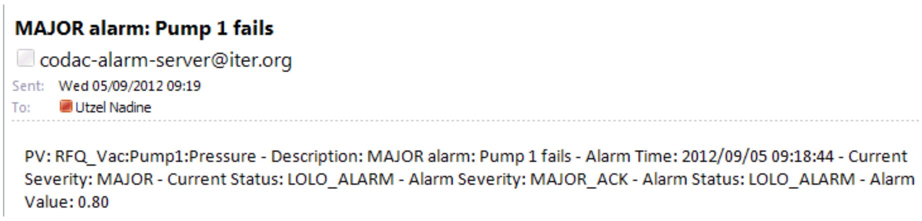
codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:19

To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MAJOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:18:44 - Current Severity: MAJOR - Current Status: LOLO_ALARM - Alarm Severity: MAJOR - Alarm Status: LOLO_ALARM - Alarm Value: 0.80

Email notification after acknowledging the previous MAJOR alarm



MAJOR alarm: Pump 1 fails

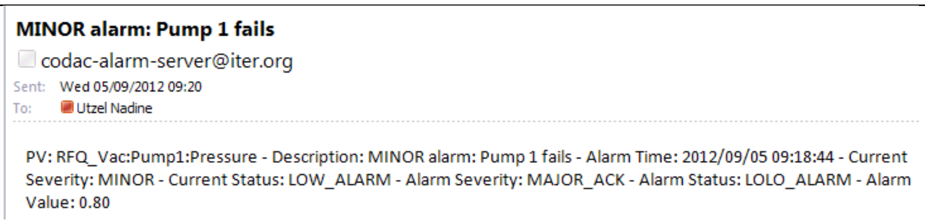
codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:19

To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MAJOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:18:44 - Current Severity: MAJOR - Current Status: LOLO_ALARM - Alarm Severity: MAJOR_ACK - Alarm Status: LOLO_ALARM - Alarm Value: 0.80

H. Email notification when PV MAJOR at 0.8 acknowledged -> MINOR alarm. As the severity is lowest, the alarm is still acknowledged



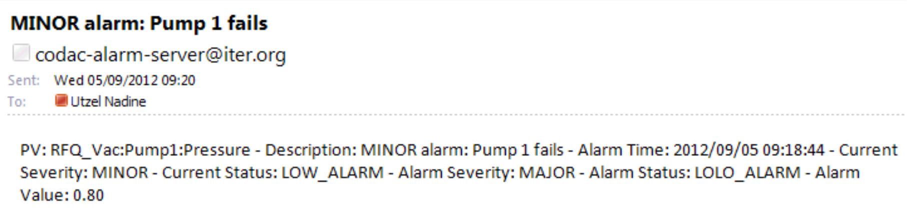
MINOR alarm: Pump 1 fails

codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:20
To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:18:44 - Current Severity: MINOR - Current Status: LOW_ALARM - Alarm Severity: MAJOR_ACK - Alarm Status: LOLO_ALARM - Alarm Value: 0.80

I. Email notification when PV MAJOR at 0.8 acknowledged -> un-acknowledged (Alarm Severity from MAJOR_ACK -> MAJOR and current severity is MINOR)



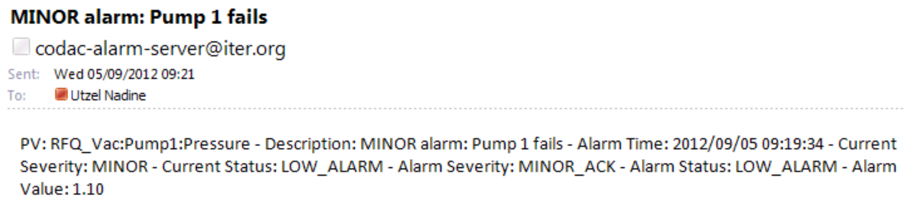
MINOR alarm: Pump 1 fails

codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:20
To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:18:44 - Current Severity: MINOR - Current Status: LOW_ALARM - Alarm Severity: MAJOR - Alarm Status: LOLO_ALARM - Alarm Value: 0.80

J. 2 x notification emails – one when the active MINOR alarm has been acknowledged and one OK alarm when the PV is within its alarm limits at 2.7.

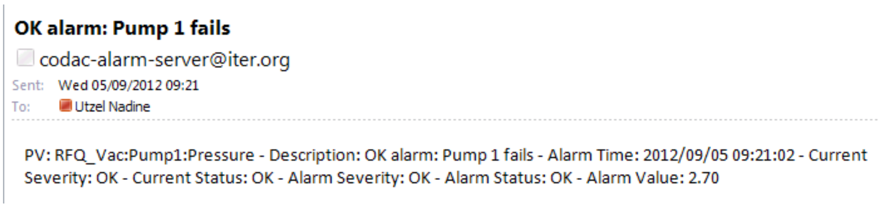


MINOR alarm: Pump 1 fails

codac-alarm-server@iter.org

Sent: Wed 05/09/2012 09:21
To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:19:34 - Current Severity: MINOR - Current Status: LOW_ALARM - Alarm Severity: MINOR_ACK - Alarm Status: LOW_ALARM - Alarm Value: 1.10



OK alarm: Pump 1 fails

codac-alarm-server@iter.org

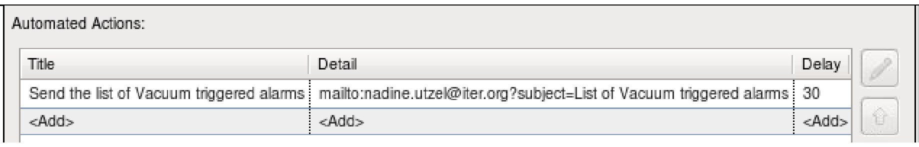
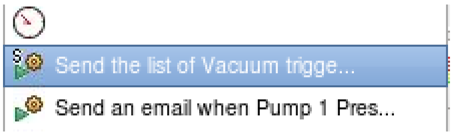
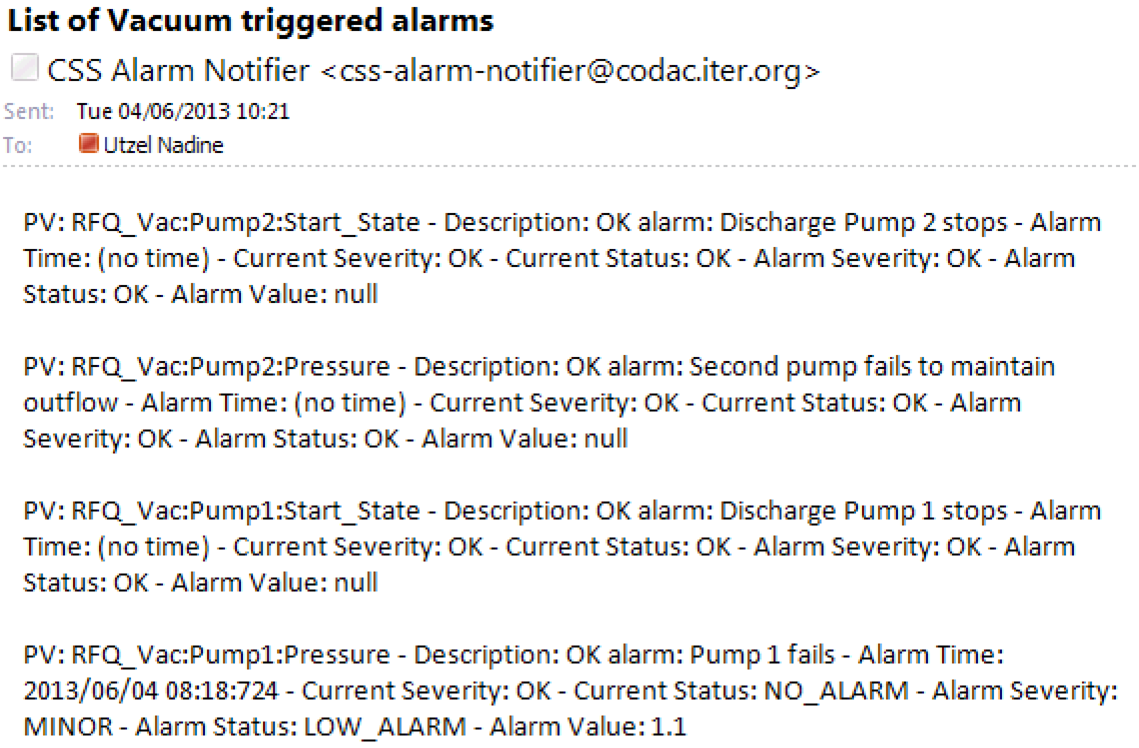
Sent: Wed 05/09/2012 09:21
To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: OK alarm: Pump 1 fails - Alarm Time: 2012/09/05 09:21:02 - Current Severity: OK - Current Status: OK - Alarm Severity: OK - Alarm Status: OK - Alarm Value: 2.70

K. No email is sent as the MINOR alarm has recovered before the delay of 5 seconds.

3.4.6 SRV03 - Component Alarm State Change Notification

Prerequisite	1. Demo IOC running 2. Alarm Configuration imported 3. Alarm server and notifier started 4. Alarm Operator Interface started	
Test Cases	1. Positive notification by email of the list of sub-alarm state changes	
Procedure	1. From the Alarm Tree, right-click on Area: RFQ -> System: Vacuum and select Configure Item 2. Add an Automated Action with Title = “Send the list of Vacuum triggered alarms”, Detail = “mailto:your_email@site.org?subject=*List of Vacuum triggered alarms” and Delay = “30” seconds.	

<table>
<tr><td></td><td>

Automated Actions:
<table>
<tr><th>Title</th><th>Detail</th><th>Delay</th></tr>
<tr><td>Send the list of Vacuum triggered alarms</td><td>mailto:nadine.utzel@iter.org?subject=List of Vacuum triggered alarms</td><td>30</td></tr>
<tr><td>&lt;Add&gt;</td><td>&lt;Add&gt;</td><td>&lt;Add&gt;</td></tr>
</table>
3. Test manually the new automated action from the Alarm Tree, right-click on “PV: RFQ_Vac:Pump1: Pressure”, select “Send the list of Vacuum trigge…” option

Send the list of Vacuum trigge...
Send an email when Pump 1 Pres...
4. Wait some seconds and check your inbox for a new alarm email
</td><td></td></tr>
<tr><td>Pass Criteria</td><td>
1. Email notification for the Vacuum component with the current state of all sub-alarms

List of Vacuum triggered alarms

CSS Alarm Notifier <css-alarm-notifier@codac.iter.org>

Sent: Tue 04/06/2013 10:21

To: Utzel Nadine

PV: RFQ_Vac:Pump2:Start_State - Description: OK alarm: Discharge Pump 2 stops - Alarm Time: (no time) - Current Severity: OK - Current Status: OK - Alarm Severity: OK - Alarm Status: OK - Alarm Value: null

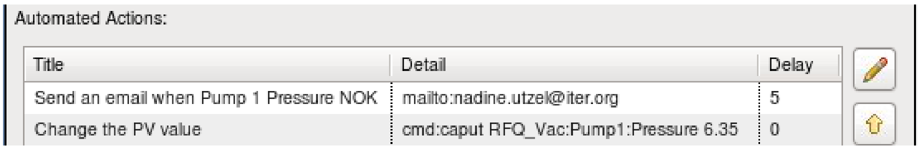
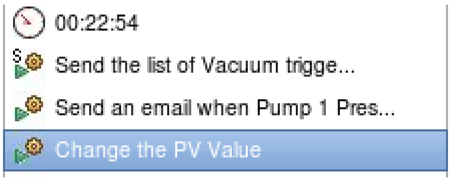
PV: RFQ_Vac:Pump2:Pressure - Description: OK alarm: Second pump fails to maintain outflow - Alarm Time: (no time) - Current Severity: OK - Current Status: OK - Alarm Severity: OK - Alarm Status: OK - Alarm Value: null

PV: RFQ_Vac:Pump1:Start_State - Description: OK alarm: Discharge Pump 1 stops - Alarm Time: (no time) - Current Severity: OK - Current Status: OK - Alarm Severity: OK - Alarm Status: OK - Alarm Value: null

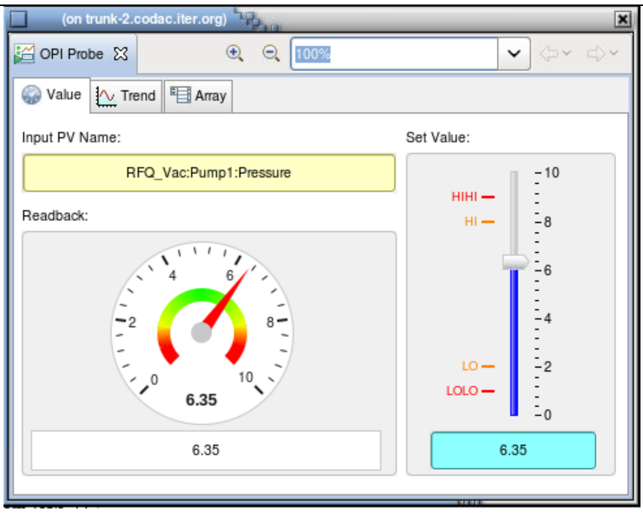
PV: RFQ_Vac:Pump1:Pressure - Description: OK alarm: Pump 1 fails - Alarm Time: 2013/06/04 08:18:724 - Current Severity: OK - Current Status: NO_ALARM - Alarm Severity: MINOR - Alarm Status: LOW_ALARM - Alarm Value: 1.1
</td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
</table>

3.4.7 SRV04 - Automated Action Syntax

Prerequisite	1. Demo IOC running 2. Alarm Configuration imported 3. Alarm server and notifier started 4. Alarm Operator Interface started	
Test Cases	1. Test of different syntax of automated action	
Procedure	1. From the Alarm Tree, right-click on “Area: RFQ -> System: Vacuum” -> “System:	

<table>
<tr>
<td></td>
<td>
Pump 1 Control” -> “PV: RFQ_VacPump1:Pressure”and select Configure Item
A. Add an Automated Action with Title = “Change the PV Value”, Detail = “cmd:caput RFQ_Vac:Pump1:Pressure 6.35” and Delay = “0” seconds

Automated Actions:
<table>
<tr><th>Title</th><th>Detail</th><th>Delay</th></tr>
<tr><td>Send an email when Pump 1 Pressure NOK</td><td>mailto:nadine.utzel@iter.org</td><td>5</td></tr>
<tr><td>Change the PV value</td><td>cmd:caput RFQ_Vac:Pump1:Pressure 6.35</td><td>0</td></tr>
</table>
After saving the configuration, execute manually the action with a right-click on “PV: RFQ_VacPump1:Pressure” and selecting the option “Change the PV Value”:

00:22:54
Send the list of Vacuum trigge...
Send an email when Pump 1 Pres...
Change the PV Value

B. Edit the alarm configuration as point A. select and delete “Change the PV Value” automated action using the Delete selected items button:
✖
Then modify the Detail of automated action “Send an email when Pump 1 Pressure NOK” and specify a person in copy and a specific subject & body: mailto:your_email@site.org?subject=Pump 1 fails to maintain outflow?cc=second_email@site.org&body={0} alarm - Water Pressure {1} bar. Execute manually the automated action by a right-clicking on the PV and selecting the option ”Send an email when Pump 1 Pres…”. Check if you have received an email with the specific subject and body as well as the person in copy
C. Edit again the same configuration and just add * before the subject and the body: mailto: your_email@site.org?subject=*Pump 1 fails to maintain outflow?cc=second_email@site.org&body=*{0} alarm - Water Pressure {1} bar. Execute manually the automated action by a right-clicking on the PV and selecting the option ”Send an email when Pump 1 Pres…”. Check if you have received an email with the new format
D. Edit again the same configuration, suppress the ?cc= information, add directly the person in the mailto list and add the field {0} in the subject to display the alarm severity: mailto:your_email@site.org,second_email@site.org?subject=*{0} Pump 1 fails to maintain outflow&body=*{0} alarm - Water Pressure {1} bar. Execute manually the automated action by a right-clicking on the PV and selecting the option ”Send an email when Pump 1 Pres…”. Check your inbox
</td>
<td></td>
</tr>
<tr>
<td>Pass Criteria</td>
<td>A. The PV value has changed to 6.35 due to the manual execution of the new automated action “Change the PV Value”</td>
<td></td>
</tr>
</table>



(on trunk-2.codac.iter.org)

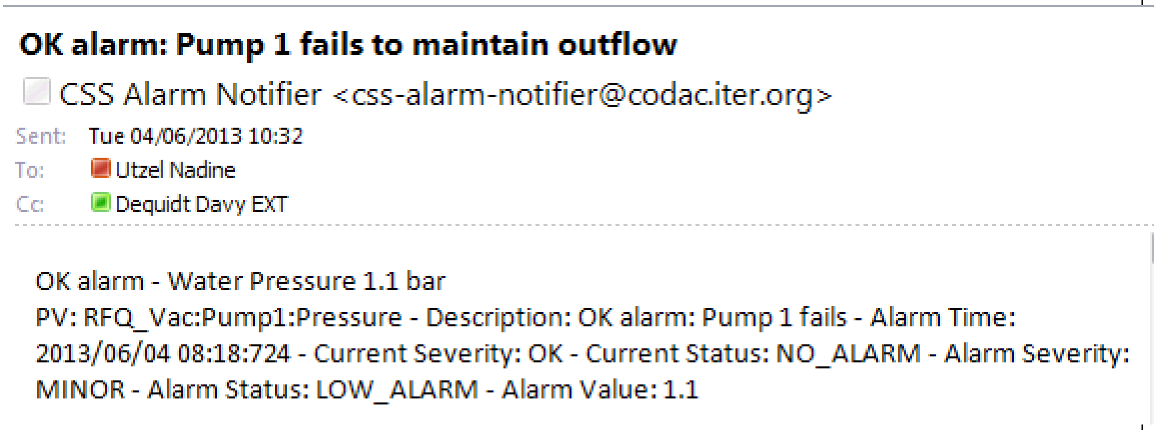
OPI Probe

Input PV Name: RFQ_Vac:Pump1:Pressure

Readback: 6.35

Set Value: 6.35

B. The Subject and the Body of the email are no more the generic ones and use the information entered in the Detail field of the automated action



OK alarm: Pump 1 fails to maintain outflow

CSS Alarm Notifier <css-alarm-notifier@codac.iter.org>

Sent: Tue 04/06/2013 10:32

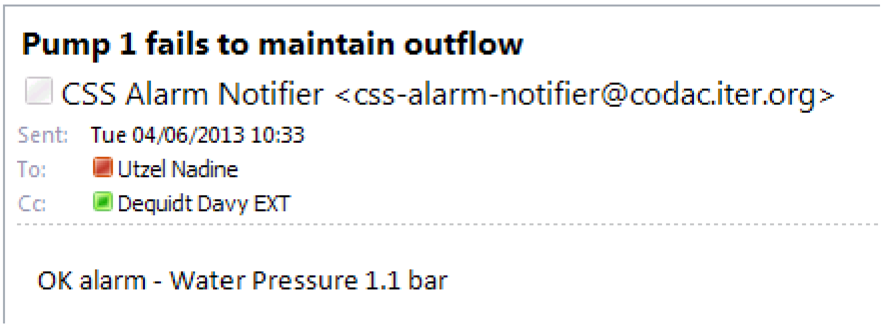
To: Utzel Nadine

Cc: Dequidt Davy EXT

OK alarm - Water Pressure 1.1 bar

PV: RFQ_Vac:Pump1:Pressure - Description: OK alarm: Pump 1 fails - Alarm Time: 2013/06/04 08:18:724 - Current Severity: OK - Current Status: NO_ALARM - Alarm Severity: MINOR - Alarm Status: LOW_ALARM - Alarm Value: 1.1

C. Check that starting the subject or the body with an asterisk, the notifier uses the information as given without prefixing the subject with the alarm severity and without adding in the body the full description of the alarm



Pump 1 fails to maintain outflow

CSS Alarm Notifier <css-alarm-notifier@codac.iter.org>

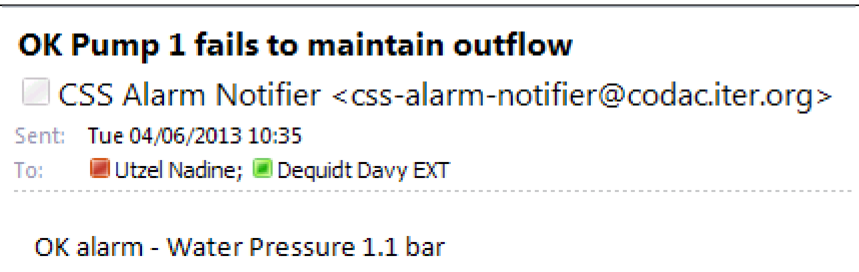
Sent: Tue 04/06/2013 10:33

To: Utzel Nadine

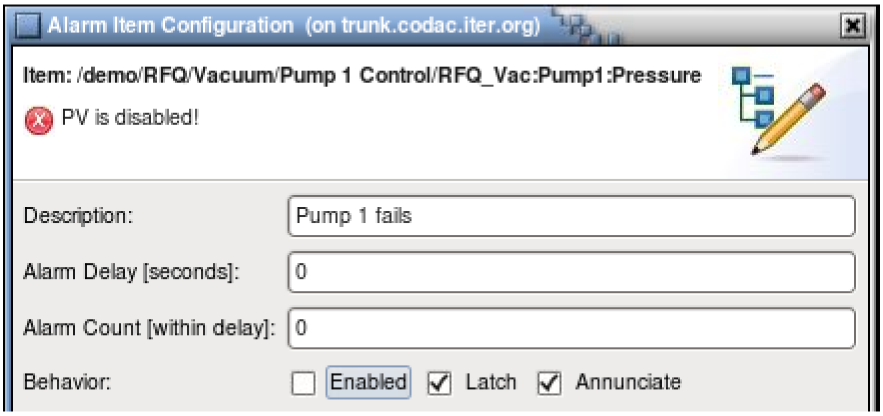
Cc: Dequidt Davy EXT

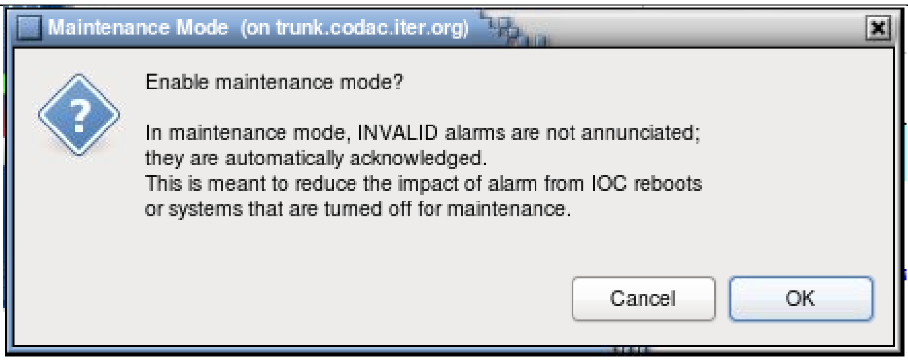
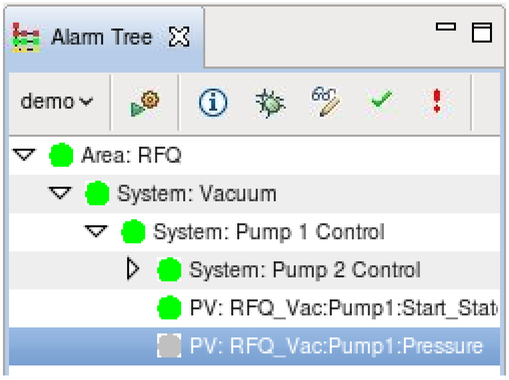
OK alarm - Water Pressure 1.1 bar

D. Check that the second person is directly in the "To:" list and that the subject contains the alarm severity

	 **OK Pump 1 fails to maintain outflow** CSS Alarm Notifier <css-alarm-notifier@codac.iter.org> Sent: Tue 04/06/2013 10:35 To: Utzel Nadine; Dequidt Davy EXT OK alarm - Water Pressure 1.1 bar	

3.4.8 SRV05 - Automated Action Disabled

Prerequisite	1. Demo IOC running 2. Alarm Configuration Imported successfully 3. Alarm server and notifier started 4. Alarm Operator Interface started	
Test Cases	1. When a PV Item in the Alarm Tree is disable, this should also disable the automated actions	
Procedure	A1. From the Alarm Tree, right-click on “Area: RFQ -> System: Vacuum” -> “System: Pump 1 Control” -> “PV: RFQ_VacPump1:Pressure”and select Configure Item A2. Uncheck the Behavior “Enabled”  Alarm Item Configuration (on trunk.codac.iter.org) Item: /demo/RFQ/Vacuum/Pump 1 Control/RFQ_Vac:Pump1:Pressure PV is disabled! Description: Pump 1 fails Alarm Delay [seconds]: 0 Alarm Count [within delay]: 0 Behavior: □ Enabled ☑ Latch ☑ Annunciate A3. Change the PV Value to 8.33 to be outside the limits, wait ~5 seconds and check that you have not received any email notification B1. Using the configuration tool, Enable again the Alarm Item. Within 5 seconds, switch to Maintenance Mode: from the Alarm Table click on the button to enable the Maintenance Mode and confirm your choice by clicking on OK	

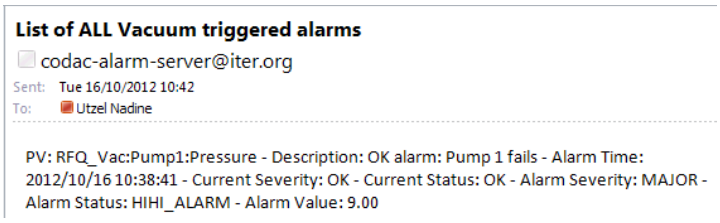
	 Maintenance Mode (on trunk.codac.iter.org) Enable maintenance mode? In maintenance mode, INVALID alarms are not annunciated; they are automatically acknowledged. This is meant to reduce the impact of alarm from IOC reboots or systems that are turned off for maintenance. B2. Change the PV Value to 9. Wait ~5 seconds but the raised MAJOR alarm should not be notified B3. Disable Maintenance Mode by clicking in the Alarm Table on the icon and confirm with OK	
Pass Criteria	A2. The Alarm Tree shows the PV in grey  Area: RFQ System: Vacuum System: Pump 1 Control System: Pump 2 Control PV: RFQ_Vac:Pump1:Start_Stat PV: RFQ_Vac:Pump1:Pressure A3. No alarm email once the PV Item is disabled B1. If after Enabling the alarm, you have not switch to Maintenance Mode within 5 seconds, you should have received a notification email for MINOR alarm - Water Pressure 8.33 bar B2. No alarm email in Maintenance Mode	

3.4.9 SRV06 - Alarm Server Configuration Reload

Prerequisite	1. Demo IOC running 2. Alarm Configuration Imported successfully 3. Alarm server and notifier started 4. Alarm Operator Interface started	
Test Cases	1. Send a JMS message to force the reload of the configuration of the alarm server and notification	
Procedure	From a Linux console, export the current alarm server configuration from the database to an XML file, modify it, import it again and send a JMS message to force the reload	

	of the configuration
	1. $ alarm-configtool -root demo -export -file demo_export.xml
	2. $ gedit demo_export.xml&
	3. Modify the Title of the automated action defined for the component name “Vacuum”: Send by email… and modify also the Detail: *List of ALL…
	`<component name="Vacuum">`
	`  <automated_action>`
	`  <title>Send by email the list of Vacuum triggered alarms</title>`
	`  <details>mailto:your_email@site.org?subject=*List of ALL Vacuum triggered alarms</details>`
	…
	Save the configuration file
	4. $ alarm-configtool -root demo -import -file demo_export.xml
	5. $ jms-send -url tcp://localhost:61616 -jms_user alarm -jms_pass $alarm -topic demo_SERVER -type alarm -text CONFIG
	6. From CSS Alarm Tree, right-click on “PV: RFQ_Vac:Pump1: Pressure”, the updated Title should appear as “Send by email the list of vacu…”
	7. From CSS using the OPI Probe tool, set the value of “PV: RFQ_Vac:Pump1: Pressure” to 4 in order to change the alarm state to OK. Check if the second alarm notification email received after ~30 seconds has the updated subject	

Pass Criteria	1. The output of the configuration export command should be:
	$ alarm-configtool -root demo -export -file demo_export.xml
	`Alarm Config Tool 3.1.1.codac_core_xx_xxx`
	`Reading RDB configuration of 'demo'`
	`Writing configuration 'demo' to demo_export.xml`
	4. The output of the configuration import command should be:
	$ alarm-configtool -root demo -import -file demo_export.xml
	`Alarm Config Tool 3.1.1.codac_core_xx_xxx`
	`Reading RDB configuration of 'demo'`
	`Deleting existing RDB configuration for 'demo'`
	`Importing configuration 'demo' from demo_export.xml`
	`Loading /demo/RFQ`
	`Loading /demo/RFQ/Vacuum`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control/Pump 2 Control`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control/Pump 2 Control/RFQ_Vac:Pump2:Start_State`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control/Pump 2 Control/RFQ_Vac:Pump2:Pressure`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control/RFQ_Vac:Pump1:Start_State`
	`Loading /demo/RFQ/Vacuum/Pump 1 Control/RFQ_Vac:Pump1:Pressure`	

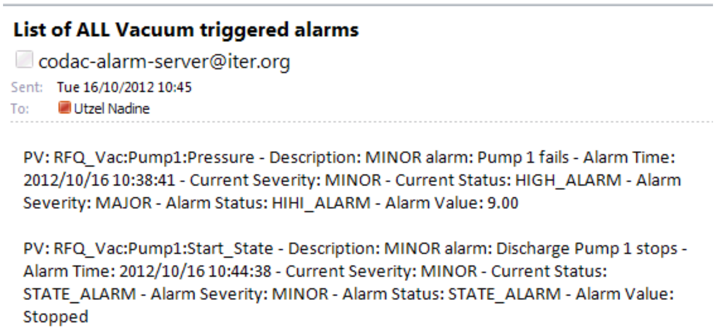
	5. After sending the JMS message, the output in the console should look like that:
	`jms-send -url tcp://localhost:61616 -jms_user alarm -jms_pass $alarm -topic -topic demo_SERVER -type alarm -text CONFIG`
	`JMSSender 3.0.0.codac_core_xx_xxx`
	`URL         : tcp://localhost:61616`
	`Topic       : demo_SERVER`
	`Type        : alarm`
	`Application: JMSSender`
	`INFO [Thread 14] org.csstudio.alarm.beast.ui.clientmodel.AlarmClientModel (readConfiguration) - Read 9 alarm tree items, 4 PVs in 0.05 seconds: 179.1 items/sec, 79.6 PVs/sec`
	`CONFIG [Thread 14] org.csstudio.alarm.beast.notifier.AlarmNotifier (handleNewAlarmConfiguration) - New Alarm Configuration`
	7. The subject of the alarm email contains the ALL string:
	
	List of ALL Vacuum triggered alarms
	codac-alarm-server@iter.org
	Sent: Tue 16/10/2012 10:42
	To: Utzel Nadine
	PV: RFQ_Vac:Pump1:Pressure - Description: OK alarm: Pump 1 fails - Alarm Time: 2012/10/16 10:38:41 - Current Severity: OK - Current Status: OK - Alarm Severity: MAJOR - Alarm Status: HIHI_ALARM - Alarm Value: 9.00	

	3.4.10 SRV07 - Flurry of Alarms	
Prerequisite	1. Demo IOC running 2. Alarm Configuration Imported successfully 3. Alarm server and notifier started 4. Alarm Operator Interface started	
Test Cases	1. When too many notifications are triggered in case of flurry of alarms, the queue of automated actions is cleared and only component level notifications are treated	
Procedure	From a Linux console, put 3 PVs in alarm state, this will trigger 4 automated actions, one for each PV and one for the component level. As the threshold in ca.ini is set to 3, a flush of ongoing action should be done: 1. $ caput RFQ_Vac:Pump1:Pressure 8; caput RFQ_Vac:Pump1:Start_Cmd Stop; caput RFQ_Vac:Pump2:Start_Cmd Stop 2. $ Check the number of received alarm emails	

Pass Criteria	1. The output of the caput commands should be:
	`Old : RFQ_Vac:Pump1:Pressure        4`
	`New : RFQ_Vac:Pump1:Pressure        8`
	`Old : RFQ_Vac:Pump1:Start_Cmd       Start`
	`New : RFQ_Vac:Pump1:Start_Cmd       Stop`	

```
Old : RFQ_Vac:Pump2:Start_Cmd         Start
New : RFQ_Vac:Pump2:Start_Cmd         Stop
```

2. The 3 emails messages should have been received: 1 after the flush of ongoing automated actions as the queue length exceeds the threshold of 3 – this message lists 2 MINOR alarms, one for PUMP 1 pressure alarm and one when the pump stopped and has been sent before the delay of 30 seconds:



List of ALL Vacuum triggered alarms

codac-alarm-server@iter.org

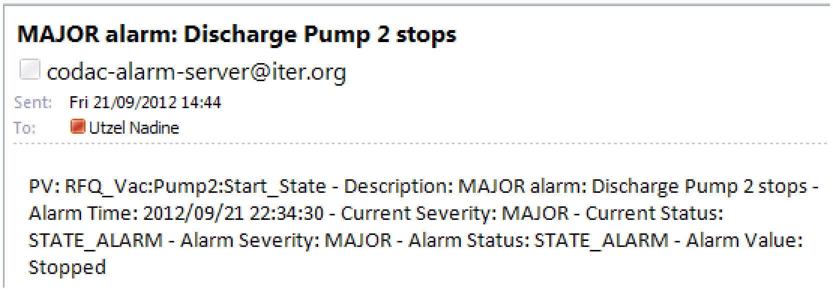
Sent: Tue 16/10/2012 10:45

To: Utzel Nadine

PV: RFQ_Vac:Pump1:Pressure - Description: MINOR alarm: Pump 1 fails - Alarm Time: 2012/10/16 10:38:41 - Current Severity: MINOR - Current Status: HIGH_ALARM - Alarm Severity: MAJOR - Alarm Status: HIHI_ALARM - Alarm Value: 9.00

PV: RFQ_Vac:Pump1:Start_State - Description: MINOR alarm: Discharge Pump 1 stops - Alarm Time: 2012/10/16 10:44:38 - Current Severity: MINOR - Current Status: STATE_ALARM - Alarm Severity: MINOR - Alarm Status: STATE_ALARM - Alarm Value: Stopped

Then 1 MAJOR alarm email regarding Pump 2 position:



MAJOR alarm: Discharge Pump 2 stops

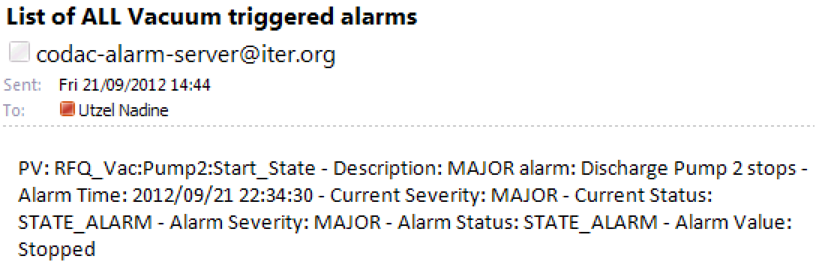
codac-alarm-server@iter.org

Sent: Fri 21/09/2012 14:44

To: Utzel Nadine

PV: RFQ_Vac:Pump2:Start_State - Description: MAJOR alarm: Discharge Pump 2 stops - Alarm Time: 2012/09/21 22:34:30 - Current Severity: MAJOR - Current Status: STATE_ALARM - Alarm Severity: MAJOR - Alarm Status: STATE_ALARM - Alarm Value: Stopped

Finally, after 30 seconds, an summary alarm email triggered by the previous notification:



List of ALL Vacuum triggered alarms

codac-alarm-server@iter.org

Sent: Fri 21/09/2012 14:44

To: Utzel Nadine

PV: RFQ_Vac:Pump2:Start_State - Description: MAJOR alarm: Discharge Pump 2 stops - Alarm Time: 2012/09/21 22:34:30 - Current Severity: MAJOR - Current Status: STATE_ALARM - Alarm Severity: MAJOR - Alarm Status: STATE_ALARM - Alarm Value: Stopped

3.4.11 LOG01 – LOG: Look for any SEVERE message

Prerequisite	1. None	
Test Cases	1. No SEVERE alert in the CSS log files	
Procedure	In a Linux console, check the log of CSS general services: 1. $ grep -r 'SEVERE' /var/opt/codac/css/ Now check the log of the services started manually for the demo applications: 2. $ grep -r 'SEVERE' ~/.css/	

Pass Criteria	1 - 2. No SEVERE messages except for: `~/.css/css/console.log:<date> SEVERE [Thread 1] org.csstudio.logging.PluginLogListener (logging) - Invalid preference page path: XML Syntax`	
To terminate the tests, stop the demo IOC, the demo alarm server and jms2rdb. Close css: 1. $ epics> exit 2. $ ps -ef|grep alarm-notifier|grep <user> 3. Kill the alarm-notifier processes using the command $ kill -9 <PID1> <PID2>… 4. $ ps -ef|grep alarm-server|grep <user> 5. Kill the alarm-server processes using the command $ kill -9 <PID1> <PID2>… 6. In CSS, close first OPI Probe by clicking on the tab cross, then use the menu File -> Exit		

3.5 Component Test Log

	3.5.1 CFG01 - Alarm Configuration Import	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.2 SRV01 - Alarm Server and Notifier Startup	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.3 DSP01 - Alarm Notification	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.4 DSP02 - Automated Action Manual Configuration	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.5 SRV02 - PV Alarm State Change Notification	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.6 SRV03 - Component Alarm State Change Notification	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	

Remarks		
	3.5.7 SRV04 - Automated Action Syntax	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.8 SRV05 - Automated Action Disabled	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.9 SRV06 - Alarm Server Configuration Reload	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
	3.5.10SRV07 - Flurry of Alarms	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		

4 SOFTWARE TEST PLAN CHECKLIST

For Assessment of:	
Agency Name	
Project Name	
Document Name	
Date	

Criteria	Yes / No / NA
DOCUMENT STANDARDS COMPLIANCE	
1 Have standards/guidelines been identified to define the work product?	
2 Does the work product format conform to the specified standard/guideline (Template)?	
3 Has the project submitted any request for deviations or waivers to the defined work product?	
4 Have the following areas been addressed completely:	
4a Approval authority?	
4b Revision approval?	
4c Revision control?	
TECHNICAL REFERENCE	
5 Is there evidence that the work product was reviewed by all stakeholders?	
6 Have acceptance criteria been established for the work product?	
7 Does the work product have a clearly defined purpose and scope?	
8 Are references to policies, directives, procedures, standards, and terminology provided?	
9 Does the work product identify any and all constraints/limitations?	
S/W TEST PLAN CONTENTS	
10 Does the S/W Test Plan address the following required information:	
10a Test levels?	
10b Test types (e.g., unit testing, software integration testing, systems integration testing, end-to-end testing, acceptance testing, regression testing)?	
10c Test classes?	
10d General test conditions?	
10e Test progression?	
10f Data recording, reduction, and analysis?	
10g Test coverage (breadth and depth) or other methods for ensuring sufficiency of testing?	
10h Planned tests, including items and their identifiers?	

Criteria	Yes / No / NA
10i Test schedules, Requirements traceability (or verification matrix)?	
10j Qualification testing environment, site, personnel, and participating organizations?	
11 Does the S/W Test Plan identify the environmental exposure as well as requirements for comprehensive, functional, aliveness, end-to-end, and mission simulation testing?	
12 Does the S/W Test Plan provide a System Overview that describes the unique complexities of the system?	
13 Does the S/W Test Plan address user guide, operations / maintenance validation?	
16 Does the S/W Test Plan identify any elements that will not be tested according to the test plan (e.g., externally developed software)?	
17 Does the S/W Test Plan address software architecture in terms of which software components will be based on heritage and which will be mostly or entirely new developments?	
18 Does the S/W Test Plan identify any software reuse? If so, is the extent of reuse or the anticipated modification described?	
S/W TEST ENVIRONMENT	
19 Does the S/W Test Plan include a figure of each system test environment? If so, does it reflect the system hardware approach, simulators, and special development?	
20 Does the S/W Test Plan identify specific test hardware and simulators for each external interface?	
TEST TOOLS	
21 Does the S/W Test Plan address test execution tools?	
TEST PROBLEM REPORTING & CORRECTIVE ACTION	
22 Does the S/W Test Plan provide a description of the problem reporting system to be used by the test team to report problems and/or recommended changes cited during the test activities?	
TEST PROGRESS PLANNING & TRACKING	
23 Does the S/W Test Plan describe the routine test progress reporting approach?	
24 Does the S/W Test Plan describe the Build Test verification methodology? If so, does the description address build verification test level objectives, environment, roles & responsibilities, entry/exit criteria, general guidelines, build test planning, build test scenario development, build test procedure preparation & dry run, build test execution, reporting, and archiving?	