IDM UID

9KMNAD

VERSION CREATED ON / VERSION / STATUS

01 Feb 2013 / 2.1/ Signed

EXTERNAL REFERENCE

Report

CODAC PON Archiving System - Test Plan

Test Plan of CODAC PON Archiving System

Approval Process			
	Name	*Action*	*Affiliation*
Author	**Utzel N.**	**01-Feb-2013:signed**	**IO/DG/DIP/CHD/CSD/CDC**
CoAuthor			
Reviewers			
Approver			
Document Security: level 1 (IO unclassified) *RO: Di Maio Franck*			
Read Access	**LG: SOPRA extra, AD: ITER, AD: External Collaborators, AD: Division - Control System Division, AD: Section - CODAC, project administrator, RO**		

Change Log				
Title (Uid)	***Version***	***Latest Status***	***Issue Date***	***Description of Change***
CODAC PON Archiving System - Test Plan (9KMNAD_v2_1)	v2.1	Signed	01 Feb 2013	Demo SVN Unit
CODAC PON Archiving System - Test Plan (9KMNAD_v2_0)	v2.0	Signed	21 Jan 2013	Test of: - event driven archiving - archiving scan mode - archiving monitor with threshold mode - archived data types 6KSamples/sec performance tests.
CODAC PON Archiving System - Test Plan (9KMNAD_v1_2)	v1.2	Signed	06 Jun 2012	At CSS startup, the Welcome screen should be closed
CODAC PON Archiving System - Test Plan (9KMNAD_v1_1)	v1.1	Signed	06 Jun 2012	Additional information regarding the workspace dedicated to the tests.
CODAC PON Archiving System - Test Plan (9KMNAD_v1_0)	v1.0	Signed	05 Jun 2012	

ITER_D_9KMNAD

CODAC PON Archiving System

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

This document describes the tests that should be performed for CODAC PON Archiving System 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 BEAUTY - Best Ever Archive Utility, Yet - in order to be installed as part of CODAC Core System. These tests also compare the capabilities of BEAUTY against these described in Philosophy of ITER PON Archiving (ITER_D_B7N2B7).

Particular functions to be tested are the archive engine configuration via an XML file generated by SDD, the different archiving/sampling modes and archive graphical user interface (GUI) that plots historic and live data – i.e. the main components of the archive system as shown on Figure 1-I - BEAUTY Architecture, except for "Other tools for configuration & samples" not part of CODAC Core System.

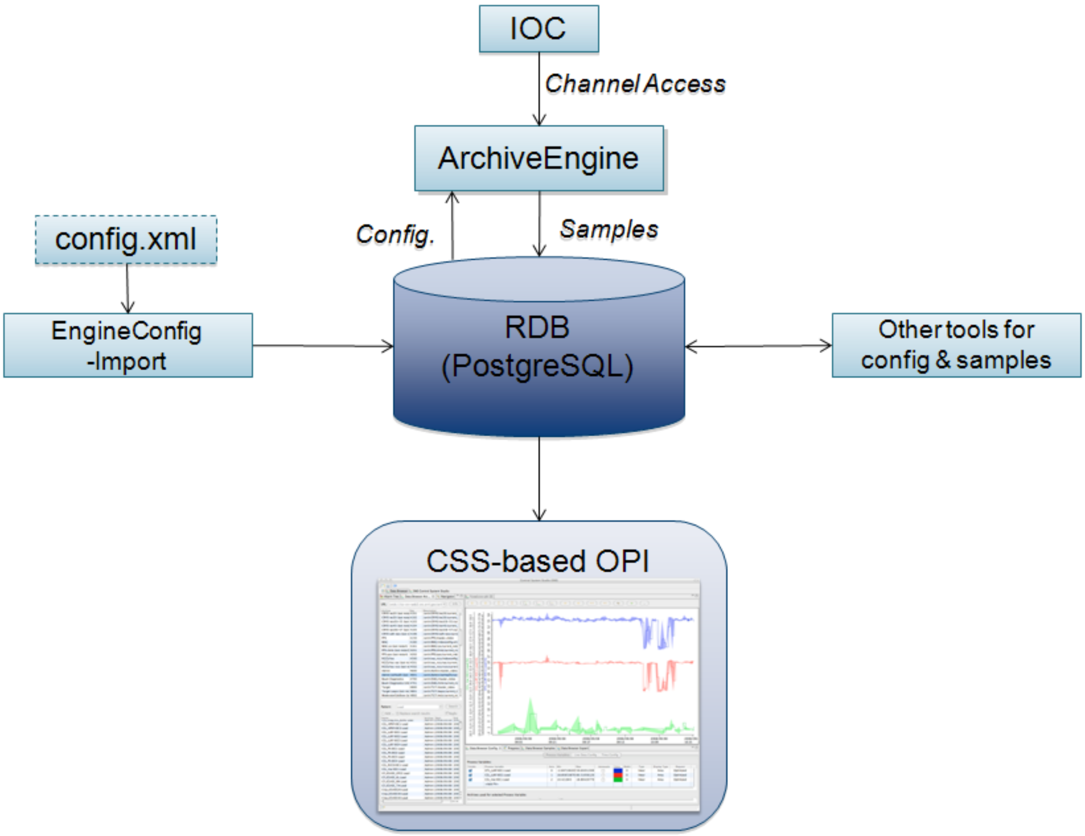


Figure 1-I - BEAUTY Architecture

1.2 Scope

The test items are:

- The operational version of BEAUTY,
- The data, including all the configuration data needed to run the archive 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 Archive Utility, Yet (BEAUTY) - is a distributed archive system consisting of:

- Archive Engine which takes samples from IOCs via Channel Access
- And stores them on a Relational Database with their original time stamp, alarm status/severity, process variable value and metadata (unit, limits…),
- CSS user interface for accessing to historic data samples in that storage.

1.4 References

[RD1] IEEE 829-2008 Standard for Software and System Test Documentation

[RD2] Bugzilla Manual for CODAC Core System Developers (33KAC4)

[RD3] STR-T – Software Test Report Template (6SBGVY)

[RD4] Philosophy of ITER PON Archiving (B7N2B7)

1.5 Definitions

BEAUTY	Best Ever Archive Utility, Yet
CSS	Control System Studio
EPICS	Experimental Physics and Industrial Control System
IOC	Input / Output Controller
PV	Process Variable
CA	Channel Access
RDB	Relational Database
GUI	Graphical User Interface
SVN	Apache Subversion
STP	Software Test Plan
STR	Software Test Report

2 Details of the Testing Process

2.1 Definition of test levels

The described component tests will focus on the desired features of CODAC PON Archiving System.

Following test levels are defined in this test plan to organize the testing activity.

Archive Configuration Component Test	**CFG**
Test of the different archiving modes import configuration	
Archive Engine Component Test	**ENG**
Test of archiving overrun and monitoring	
Archive Display Component Test	**DSP**
Test of archived data plot in CSS and Web Data Browser	
Archive Report Component Test	**RPT**
Test of Web Archive Reports	
Archive Engine System Performance Test	**PRF**
Test of at least 4K samples archived continuously in the database	

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 source 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 PON Archiving System includes the following products:

- org.csstudio.iter.archive.config.rdb.product with the list of features:
    - org.csstudio.iter.archive.config.rdb.app.feature
    - org.csstudio.iter.core.util.feature
    - org.eclipse.iter.feature
- org.csstudio.iter.archive.engine.product with the list of features:
    - org.csstudio.iter.archive.engine.app.feature
    - org.csstudio.iter.core.util.feature
    - org.eclipse.iter.feature

3.1.2 Features to be tested

The main CODAC PON Archiving System features to be tested are:

- Archive configuration import/export
- Archive Engine startup and monitoring
- PV changes notification and archiving
- Historical data plot

3.1.3 Features not to be tested

The Other tools for configuration and samples are not part of CODAC PON Archiving System 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 PON Archiving System.

3.2.2 Item pass/fail criteria

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

3.3 Test 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

<table>
<tr><td colspan="2"></td></tr>
<tr><td>CFG-01</td><td>

3.4.1 Archive Configuration Import

</td></tr>
<tr><td>Prerequisite</td><td>

In a Linux console, create a working directory, download and start a demo IOC:
1.$ mkdir test
2.$ cd test
3.$ svn co https://svnpub.iter.org/codac/iter/codac/dev/units/m-css/trunk/products/ITER/products/org.csstudio.iter.archive.engine.product/demo/m-TEST-BEAUTY

```
A    m-TEST-BEAUTY/.project
A    m-TEST-BEAUTY/doc
A    m-TEST-BEAUTY/doc/STP-CODAC_Engineering_Archiving_System.pdf
A    m-TEST-BEAUTY/src
A    …
A    m-TEST-BEAUTY/sdd.xml
A    m-TEST-BEAUTY/pom.xml
Checked out revision xxx.
```

4.$ cd m-TEST-BEAUTY

5.$ softIoc -s -d src/main/epics/TEST-BTY0App/Db/PSH0-TEST-BTY0.db

```
Starting iocInit

################################################################
## EPICS R3.14.12.3-rc1 $Date: Mon 2012-12-03 16:39:27 -0600$
## EPICS Base built Dec 10 2012
###############################################################
cas warning: Configured TCP port was unavailable.
cas warning: Using dynamically assigned TCP port 51323,
cas warning: but now two or more servers share the same UDP port.
cas warning: Depending on your IP kernel this server may not be
cas warning: reachable with UDP unicast (a host's IP in EPICS_CA_ADDR_LIST)
iocRun: All initialization complete
epics>
```

6. List the EPICS PVs defined in the database with the command dbl

```
epics> dbl
TEST-BTY0:AI1
TEST-BTY0:AI2
TEST-BTY0:AI3
TEST-BTY0:ARCHIVE
TEST-BTY0:BI
TEST-BTY0:RAMP1
TEST-BTY0:RAMP2
TEST-BTY0:RNDM-AI
TEST-BTY0:RNDM-BI
TEST-BTY0:RNDM-MBBI
TEST-BTY0:COMPRESS
TEST-BTY0:LONGIN
TEST-BTY0:MBBI
TEST-BTY0:STRING
TEST-BTY0:WAVEFORM
epics>
```

</td></tr>
<tr><td>Test Cases</td><td>1. Positive confirmation of the archive configuration loaded</td></tr>
<tr><td>Procedure</td><td>

In another Linux console or new Tab:

1.$ cd test/m-TEST-BEAUTY

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

<table>
<tr><td></td><td>2.$ archive-configtool -engine demo -description 'Demo Test Engine' -port 5812 -import -config src/main/beauty/TEST-BTY0-beauty.xml -replace_engine
Check that the Demo Archive Engine is configured
3. $ archive-configtool -list</td><td></td></tr>
<tr><td>Pass Criteria</td><td>2. The output of the command should be:
<pre>
archive-configtool -engine demo -port 5812 -import -config src/main/beauty/TEST-BTY0-beauty.xml -replace_engine
Importing   : src/main/beauty/TEST-BTY0-beauty.xml
Engine   : demo
Description   : Demo Test Engine
URL   : http://localhost:5812/main
Replace engine: true
Steal channels: false
2012-12-14 10:39:39.711 INFO [Thread 19]
org.apache.activemq.transport.failover.FailoverTransport (doReconnect) - Successfully
connected to tcp://localhost:61616
…
Import 'demo', Group 'TEST-BTY0'
Import 'demo', Group 'TEST-BTY0-SYSM'
</pre>
3. The output of the command should contain the following declaration:
<pre>
archive-configtool -list
2013-01-07 13:43:16.931 INFO [Thread 19]
org.apache.activemq.transport.failover.FailoverTransport (doReconnect) - Successfully
connected to tcp://localhost:61616
Engine 'demo' (Demo Test Engine) at http://localhost:5812/main [xx]
…
</pre></td><td></td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td>CFG-02</td><td><b>3.4.2 Archive Configuration Export</b></td><td></td></tr>
<tr><td>Prerequisite</td><td>1. Archive Configuration Imported successfully</td><td></td></tr>
<tr><td>Test Cases</td><td>1. Positive confirmation of the archive configuration export</td><td></td></tr>
<tr><td>Procedure</td><td>In the previous Linux console, export the Demo Archive Engine configuration:
1.$ archive-configtool -engine demo -export -config src/main/beauty/export-beauty.xml
Check the exported configuration:
3. $ gedit src/main/beauty/export-beauty.xml&
After the check, close gedit.</td><td></td></tr>
<tr><td>Pass Criteria</td><td>1. The output of the command should be:
<pre>
archive-configtool -engine demo -export -config src/main/beauty/export-beauty.xml
Exporting config for engine demo to src/main/beauty/export-beauty.xml
2013-01-07 13:50:18.700 INFO [Thread 19]
org.apache.activemq.transport.failover.FailoverTransport (doReconnect) - Successfully
connected to tcp://localhost:61616
</pre>
2. The xml configuration should be:
<pre>
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created by ArchiveConfigTool -engine demo -export
     2013/01/21 09:16:05.560116728
 -->
<engineconfig>
  <group>
    <name>TEST-BTY0</name>
</pre></td><td></td></tr>
</table>

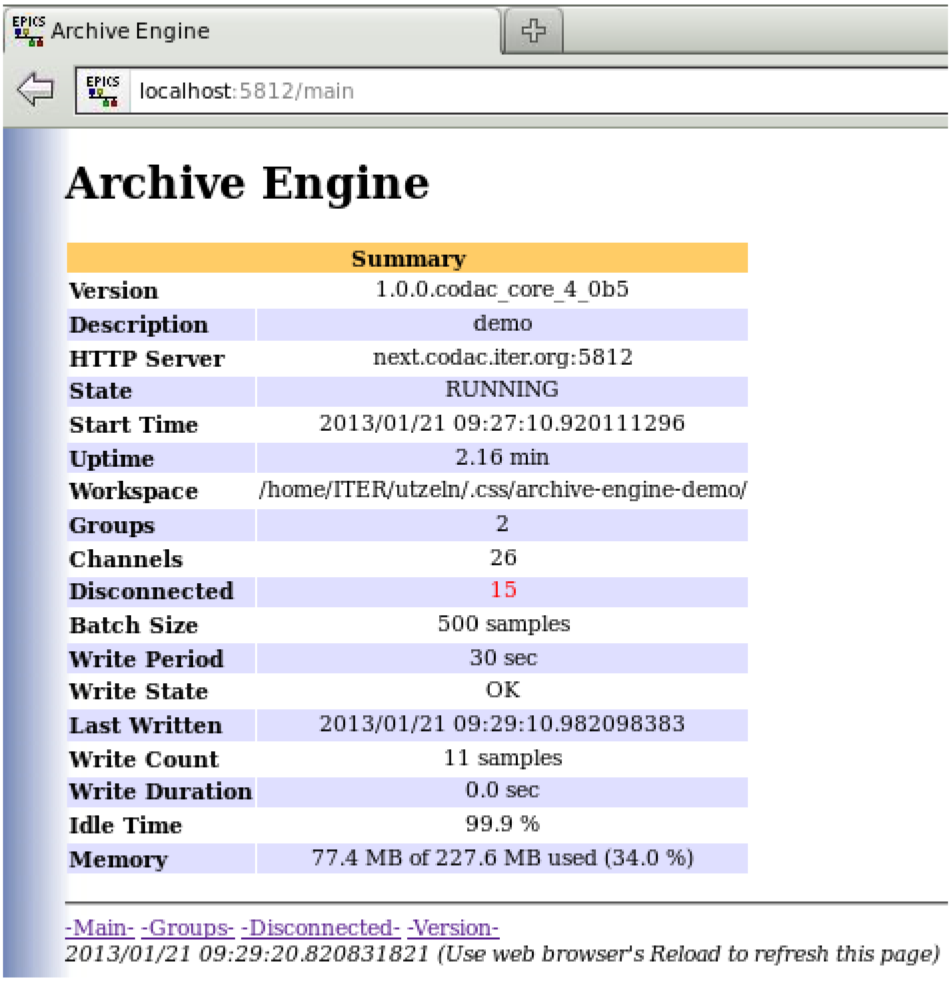
```
      <channel><name>TEST-BTY0:AI1</name><period>00:00:00.100</period><monitor/></channel>
      <channel><name>TEST-BTY0:AI2</name><period>00:00:00.100</period><monitor/></channel>
      <channel><name>TEST-BTY0:AI3</name><period>00:00:00.100</period><monitor/></channel>
      <channel><name>TEST-BTY0:ARCHIVE</name><period>00:00:01</period><monitor/></channel>
      <channel><name>TEST-BTY0:BI</name><period>00:00:01</period><monitor/></channel>
      <channel><name>TEST-BTY0:LONGIN</name><period>00:00:00.100</period><monitor/></channel>
      <channel><name>TEST-BTY0:MBBI</name><period>00:00:01</period><monitor/></channel>
      <channel><name>TEST-BTY0:RAMP1</name><period>00:00:02</period><monitor/></channel>
      <channel><name>TEST-BTY0:RAMP2</name><period>00:00:02</period><monitor/></channel>
      <channel><name>TEST-BTY0:STRING</name><period>00:05:00</period><scan/></channel>
      <channel><name>TEST-BTY0:WAVEFORM</name><period>00:01:00</period><scan/></channel>
  </group>
  <group>
    <name>TEST-BTY0-SYSM</name>
      <channel><name>TEST-BTY0-SYSM:H0-CLOAD</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0-FLOAD</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0-MLOAD</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0-NPRDROP</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0-NPSDROP</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0CORE-CPUUTL</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0CORE-DBDLC</name><period>00:05:00</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0CORE-FDUTL</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0CORE-MEMUTL</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0CORE-UPT</name><period>00:05:00</period><scan/></channel>
      <channel><name>TEST-BTY0-SYSM:H0SYSM-CPUUTL</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0SYSM-DBDLC</name><period>00:05:00</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0SYSM-FDUTL</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0SYSM-MEMUTL</name><period>00:00:05</period><monitor/></channel>
      <channel><name>TEST-BTY0-SYSM:H0SYSM-UPT</name><period>00:05:00</period><scan/></channel>
  </group>
</engineconfig>
```

ENG-01	**3.4.3 Archive Engine Startup**	
Prerequisite	1. Demo IOC running 2. Archive Configuration Imported successfully	
Test Cases	1. Positive confirmation of the demo Archive Engine started	
Procedure	In the previous Linux console, start the “demo” Archive Engine: 1.$ archive-engine -port 5812 -engine demo&	

Pass Criteria	1. The output of the command should be:
	`$ INFO [Thread 10] org.csstudio.archive.engine.Application (start) - Archive Engine <version>`
	`INFO [Thread 10] org.csstudio.archive.engine.server.EngineServer (<init>) - Engine HTTP Server on http://localhost:5812/main`
	`INFO [Thread 10] org.csstudio.archive.engine.Application (start) - Reading configuration 'demo'`
	`INFO [Thread 10] org.csstudio.archive.engine.Application (start) - Read configuration: 0 channels in 0.943 seconds`
	`2013-01-21 09:27:10.813 CONFIG [Thread 1] org.csstudio.utility.pvmanager.epics.Epics3DataSource (<clinit>) - Loading epics data source parameters: com.cosylab.epics.caj.CAJContext - 2`
	`2013-01-21 09:27:10.885 CONFIG [Thread 1] org.csstudio.utility.pvmanager.Activator (start) - Adding data source ca`
	`2013-01-21 09:27:10.885 CONFIG [Thread 1] org.csstudio.utility.pvmanager.Activator (start) - Adding data source sim`
	`2013-01-21 09:27:10.886 CONFIG [Thread 1] org.csstudio.utility.pvmanager.Activator (start) - Adding data source loc`
	`2013-01-21 09:27:10.886 CONFIG [Thread 1] org.csstudio.utility.pvmanager.Activator (start) - Adding data source epics`
	`2013-01-21 09:27:10.888 CONFIG [Thread 1] org.csstudio.utility.pvmanager.Activator (start) - Setting default data source to epics`
	`<… many Info messages …>`	

ENG-02	**3.4.4 Archive Engine Monitoring**	
Prerequisite	1. Demo IOC running	

<table>
<tr><td></td><td>2. Archive Configuration imported
3. Archive Engine started</td></tr>
<tr><td>Test Cases</td><td>1. Positive confirmation of the Archive Engine Monitoring</td></tr>
<tr><td>Procedure</td><td>In the previous Linux console, start the web interface of the demo Archive Engine:
1.$ firefox http://localhost:5812/main&</td></tr>
<tr><td>Pass Criteria</td><td>1. The demo Archive Engine web interface should display the Write State (OK), Write Count (0) and Last Written sample time:



Archive Engine

<table>
<tr><th colspan="2">Summary</th></tr>
<tr><td>Version</td><td>1.0.0.codac_core_4_0b5</td></tr>
<tr><td>Description</td><td>demo</td></tr>
<tr><td>HTTP Server</td><td>next.codac.iter.org:5812</td></tr>
<tr><td>State</td><td>RUNNING</td></tr>
<tr><td>Start Time</td><td>2013/01/21 09:27:10.920111296</td></tr>
<tr><td>Uptime</td><td>2.16 min</td></tr>
<tr><td>Workspace</td><td>/home/ITER/utzeln/.css/archive-engine-demo/</td></tr>
<tr><td>Groups</td><td>2</td></tr>
<tr><td>Channels</td><td>26</td></tr>
<tr><td>Disconnected</td><td>15</td></tr>
<tr><td>Batch Size</td><td>500 samples</td></tr>
<tr><td>Write Period</td><td>30 sec</td></tr>
<tr><td>Write State</td><td>OK</td></tr>
<tr><td>Last Written</td><td>2013/01/21 09:29:10.982098383</td></tr>
<tr><td>Write Count</td><td>11 samples</td></tr>
<tr><td>Write Duration</td><td>0.0 sec</td></tr>
<tr><td>Idle Time</td><td>99.9 %</td></tr>
<tr><td>Memory</td><td>77.4 MB of 227.6 MB used (34.0 %)</td></tr>
</table>

-Main- -Groups- -Disconnected- -Version-

2013/01/21 09:29:20.820831821 (Use web browser's Reload to refresh this page)</td></tr>
<tr><td>ENG-03</td><td>

3.4.5 Event Driven Archiving

</td></tr>
<tr><td>Prerequisite</td><td>1. Demo IOC running
2. Archive Configuration imported
3. Archive Engine started</td></tr>
<tr><td>Test Cases</td><td>1. Archiving is enabled according to a specific variable defined as the “enabling” variable - TEST-BTY0:ARCHIVE: whenever the value of this variable is ON, sampling and archiving of the whole group TEST-BTY0 will be enabled until the variable returns to OFF</td></tr>
<tr><td>Procedure</td><td>Using the previous web interface of the demo Archive Engine:</td></tr>
</table>

1. Click on the link Groups at the bottom of the screen and check the status of the group TEST-BTY0

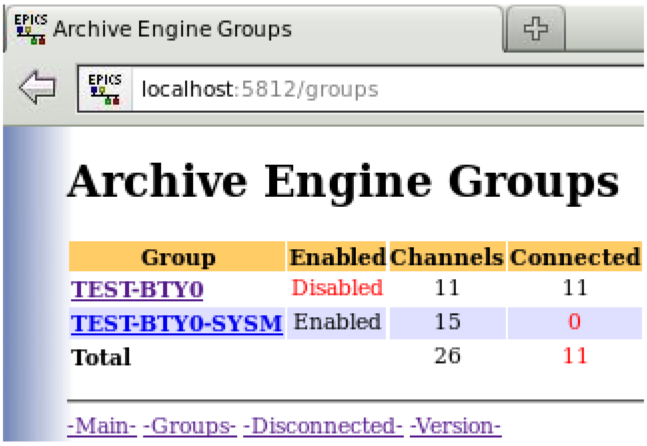
From the previous Linux console, change the Enabling PV value:

2.$ caput TEST-BTY0:ARCHIVE ON

3. Refresh the Archive Engine Groups Web page and check if the status of the group TEST-BTY0 has changed

Pass Criteria

1. The Archive Engine Groups page should inform that the group TEST-BTY0 is Disabled:



Archive Engine Groups

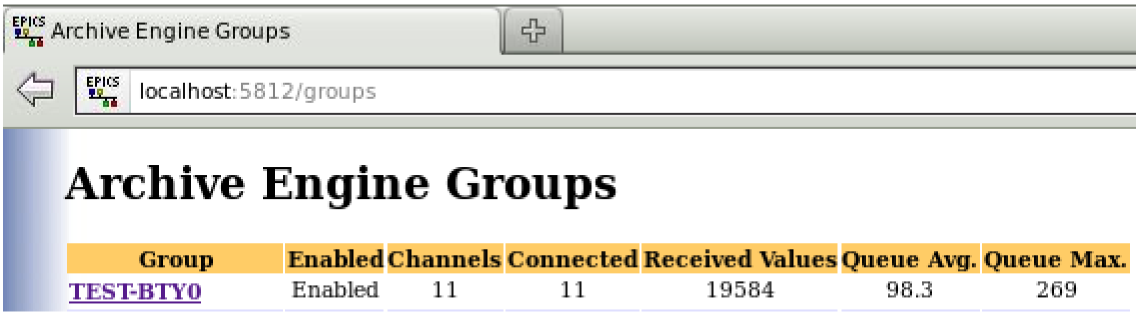
Group	Enabled	Channels	Connected
TEST-BTY0	Disabled	11	11
TEST-BTY0-SYSM	Enabled	15	0
Total		26	11

-Main- -Groups- -Disconnected- -Version-

2. The output of the caput command is:

```
Old : TEST-BTY0:ARCHIVE                      OFF
New : TEST-BTY0:ARCHIVE                      ON
```

3. The Archive Engine Groups page should inform that the group TEST-BTY0 is now Enabled and that some Samples have already been received:



Archive Engine Groups

Group	Enabled	Channels	Connected	Received Values	Queue Avg.	Queue Max.
TEST-BTY0	Enabled	11	11	19584	98.3	269

ENG-04

3.4.6 Archiving Monitor Mode

Prerequisite

1. Demo IOC running

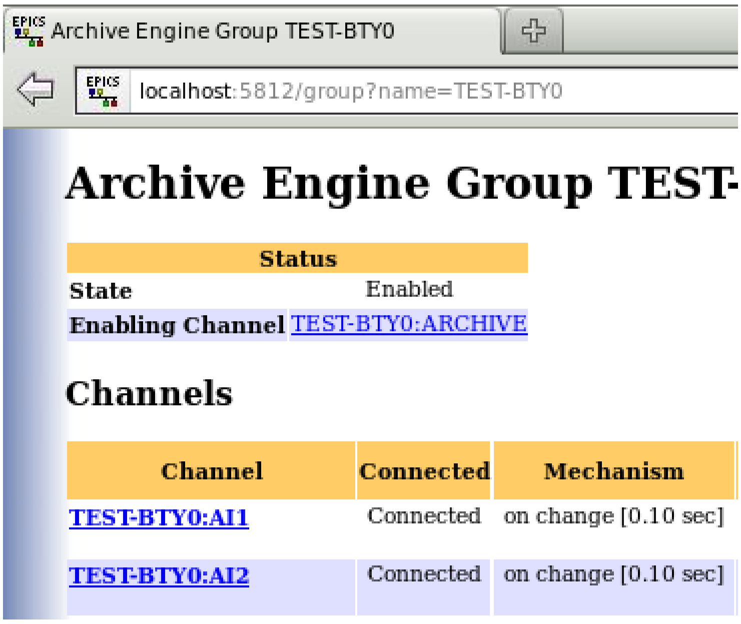
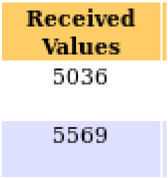
2. Archive Configuration imported

3. Archive Engine started and Group Enabled

Test Cases

1. Using the Archiving Monitor mode, each value received is stored:

- In this mode, the Archive Engine requests a Channel Access Monitor, i.e. it subscribes to changes and stores all the values sent out,
- The Channel Access server determines when values are sent to the Archive Engine client.

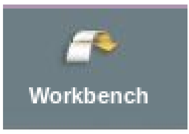
	TEST-BTY0:AI1 is an analogue value at 10Hz from 0 to 100C with an archive threshold ADEL of 5 – so only changes above 5C will be archived. On the other hand, TEST-BTY0:AI2 is also an analogue value at 10Hz from 0 to 100C but without ADEL defined – so all changes will be archived, at 10 values per second.
Procedure	Using the previous Archive Engine Groups Web page: 1. Click on the link TEST-BTY0 and check the Received Values for TEST-BTY0:AI1 and TEST-BTY0:AI2
Pass Criteria	1. The Archive Engine Group TEST-BTY0 page should show that the 2 analogue signals are archived on change at estimated rate of 10Hz:  Archive Engine Group TEST-BTY0 localhost:5812/group?name=TEST-BTY0 **Archive Engine Group TEST-** **Status** **State** Enabled **Enabling Channel** TEST-BTY0:ARCHIVE **Channels** **Channel** / **Connected** / **Mechanism** TEST-BTY0:AI1 / Connected / on change [0.10 sec] TEST-BTY0:AI2 / Connected / on change [0.10 sec] 2. And it should show that more values are archived for the PV without ADEL threshold: TEST-BTY0:AI1 samples < TEST-BTY0:AI2 samples.  **Received Values** 5036 5569
DSP-01	**3.4.7 Historical Data Plot**
Prerequisite	1. Demo IOC running 2. Archive Configuration imported 3. Archive Engine started
Test Cases	1. Positive confirmation of historical data retrieval
Procedure	In the previous Linux console, start the Operator Interface to plot historical data: 1.$ css& 2. Select a Workspace by browsing and selecting the working directory test. Click on OK to

validate the workspace

Check the Welcome pages and online Help:

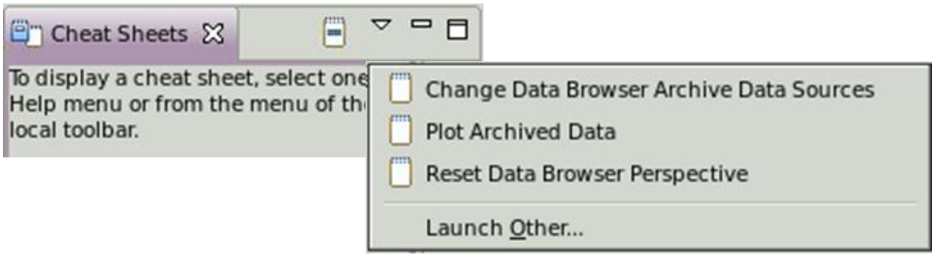
3. From the Welcome to “CSS for ITER!” Page, click on “First Steps”. A short description of CSS Data Browser should be given. Click then on the link “Data Browser” in this “First Steps” page, just before the short description. The Online Help is displayed.

4. Close the Online Help windows and Close the Welcome screen by clicking on Workbench icon:



.

From the Help menu, select the option Cheat Sheets… (Shift+Alt+Q H) and from the new View Cheat Sheets, use the down arrow to select Plot Archived Data:


Cheat Sheets
Change Data Browser Archive Data Sources
Plot Archived Data
Reset Data Browser Perspective
Launch Other...

To display a cheat sheet, select one
Help menu or from the menu of th
local toolbar.

.

You can try to do the lesson with Pattern Search criteria “TEST-BTY0:*” and by plotting any test PVs. When finished, close the Cheat Sheets View.

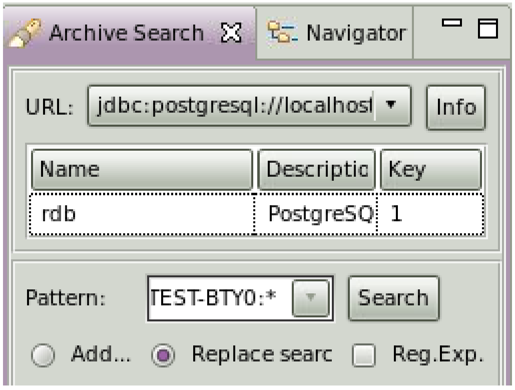
5. Open the Archive Perspective: Window -> Open Perspective -> Other… and select Data Browser

In CSS, open a new plot:

6. Click on button Open a new Data Browser plot in the menu bar

In CSS, from the Archive Search View:

7. Enter in the Pattern field TEST-BTY0:* and click on Enter key



URL: jdbc:postgresql://localhost

Name	Descriptio	Key
rdb	PostgreSQ	1

Pattern: TEST-BTY0:*

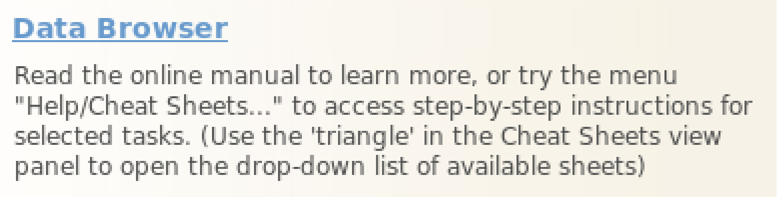
○ Add... ● Replace searc □ Reg.Exp.

8. Drag and drop the 2 ramp PVs from the searched PVs list to the plot - TEST-BTY0:RAMP1 and TEST-BTY0:RAMP2

9. Right-click on the plot and select the option Inspect Samples. In the view, select one of the 2 plotted PVs and check that the first samples have been retrieved from the database (RDB)

<table>
<tr>
<td>Pass Criteria</td>
<td>

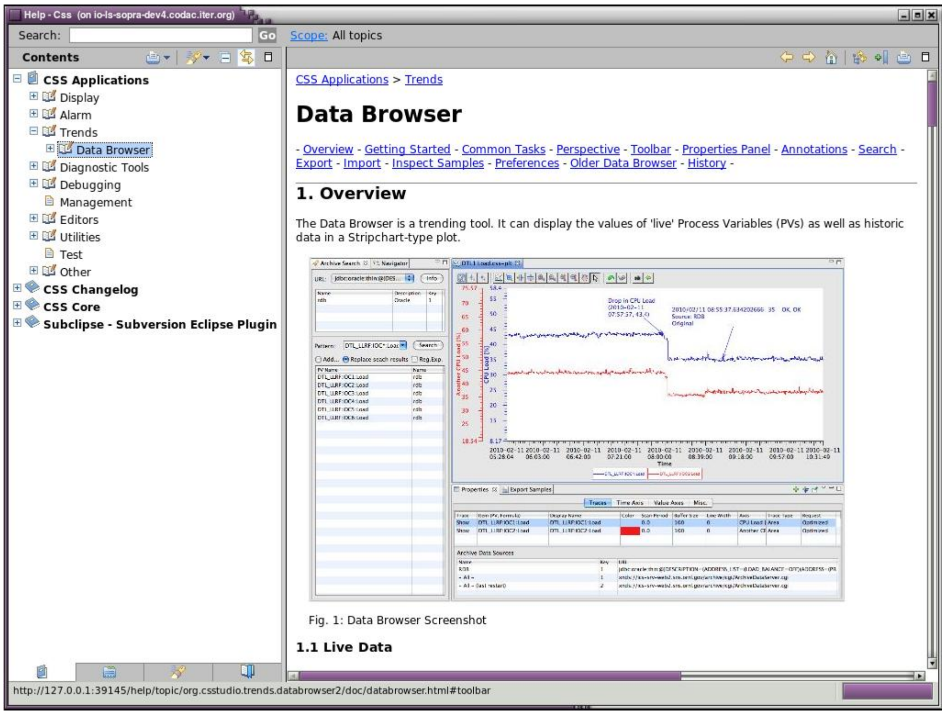
3. Welcome First Steps for CSS Data Browser should appear:



Data Browser

Read the online manual to learn more, or try the menu "Help/Cheat Sheets..." to access step-by-step instructions for selected tasks. (Use the 'triangle' in the Cheat Sheets view panel to open the drop-down list of available sheets)

After clicking on : Data Browser”, the Online Help is opened on the Data Browser topic:



CSS Applications > Trends

Data Browser

- Overview - Getting Started - Common Tasks - Perspective - Toolbar - Properties Panel - Annotations - Search - Export - Import - Inspect Samples - Preferences - Older Data Browser - History -

1. Overview

The Data Browser is a trending tool. It can display the values of 'live' Process Variables (PVs) as well as historic data in a Stripchart-type plot.

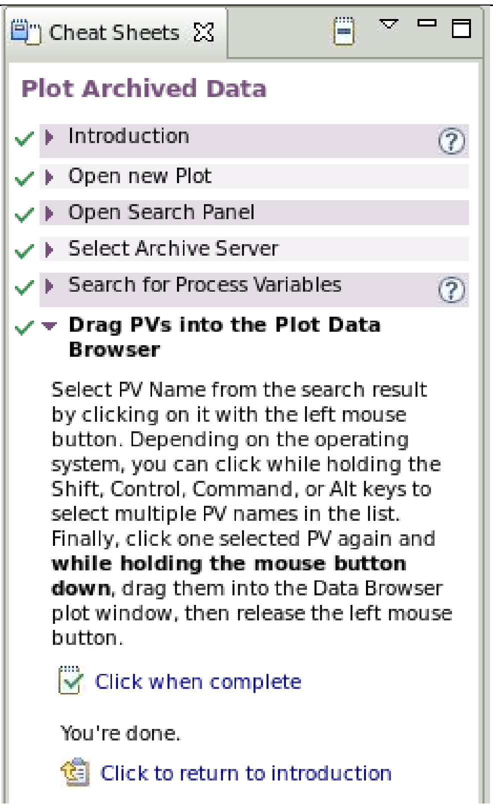
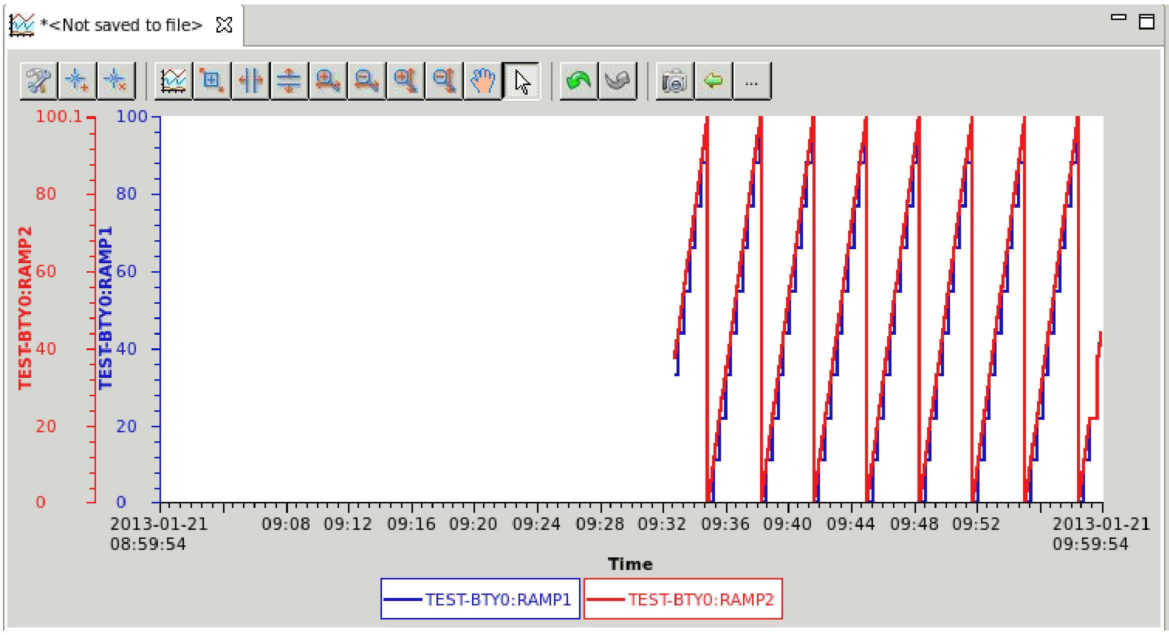
Fig. 1: Data Browser Screenshot

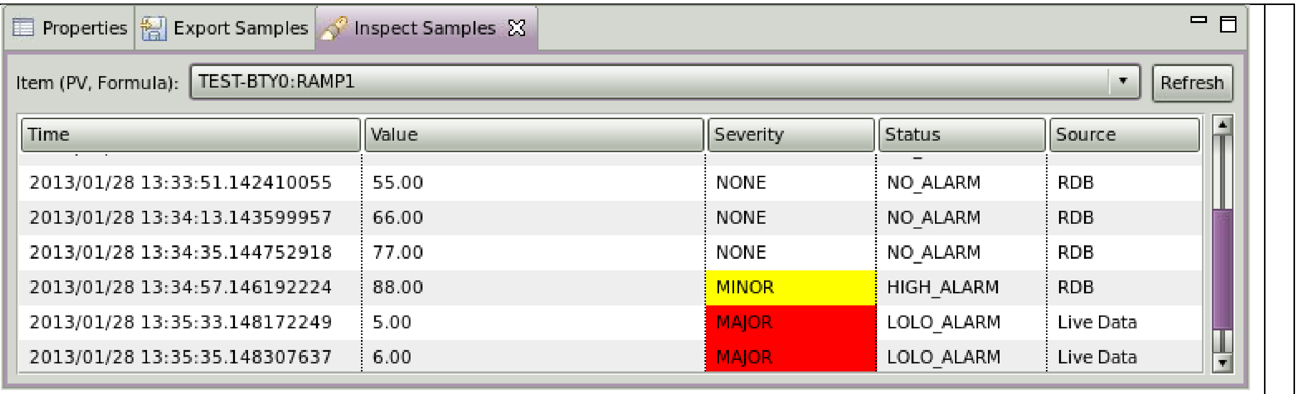
1.1 Live Data

http://127.0.0.1:39145/help/topic/org.csstudio.trends.databrowser2/doc/databrowser.html#toolbar

4. Once the lesson is finished, the Cheat Sheets View should look like:

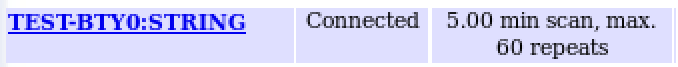
</td>
<td></td>
</tr>
</table>

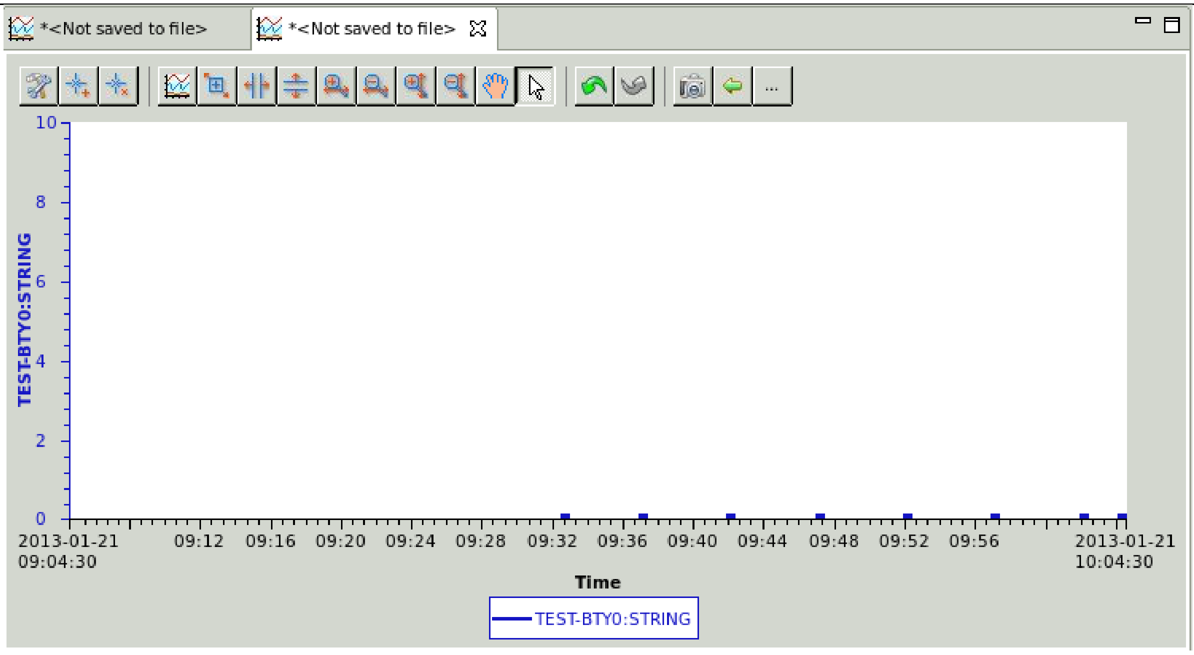
	 Cheat Sheets Plot Archived Data Introduction Open new Plot Open Search Panel Select Archive Server Search for Process Variables Drag PVs into the Plot Data Browser Select PV Name from the search result by clicking on it with the left mouse button. Depending on the operating system, you can click while holding the Shift, Control, Command, or Alt keys to select multiple PV names in the list. Finally, click one selected PV again and **while holding the mouse button down**, drag them into the Data Browser plot window, then release the left mouse button. You're done. 8. Historical data from -1hour should be plotted:  *<Not saved to file> 100.1 80 60 40 20 0 100 TEST-BTY0:RAMP2 TEST-BTY0:RAMP1 2013-01-21 08:59:54 09:08 09:12 09:16 09:20 09:24 09:28 09:32 09:36 09:40 09:44 09:48 09:52 2013-01-21 09:59:54 Time TEST-BTY0:RAMP1 TEST-BTY0:RAMP2 9. Source of the first samples is RDB:



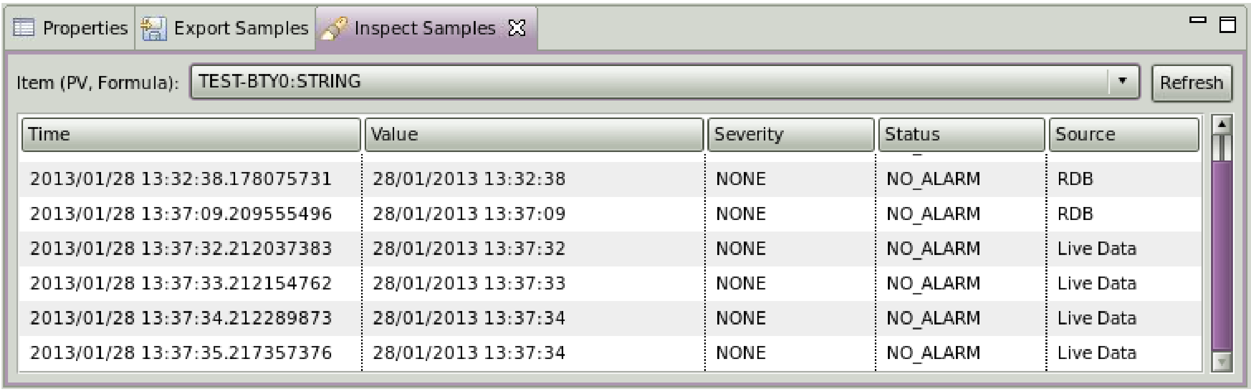
Item (PV, Formula): TEST-BTY0:RAMP1

Time	Value	Severity	Status	Source
2013/01/28 13:33:51.142410055	55.00	NONE	NO_ALARM	RDB
2013/01/28 13:34:13.143599957	66.00	NONE	NO_ALARM	RDB
2013/01/28 13:34:35.144752918	77.00	NONE	NO_ALARM	RDB
2013/01/28 13:34:57.146192224	88.00	MINOR	HIGH_ALARM	RDB
2013/01/28 13:35:33.148172249	5.00	MAJOR	LOLO_ALARM	Live Data
2013/01/28 13:35:35.148307637	6.00	MAJOR	LOLO_ALARM	Live Data

ENG-05	**3.4.8 Archiving Scan Mode**	
Prerequisite	1. Demo IOC running 2. Archive Configuration imported 3. Archive Engine started	
Test Cases	1. Using the Archiving Scan mode, the Archive Engine receives each update from the data source but only writes the most recent ones at periodic times. TEST-BTY0:STRING acquires the IOC date & time every second but is archived using the scan mode every 5 minutes.	
Procedure	First, from the Archive Engine Group TEST-BTY0 monitoring web page: 1. Check the archiving Mechanism defined for TEST-BTY0:STRING In CSS, open a new plot: 2. Click on button Open a new Data Browser plot in the menu bar In CSS, from the Archive Search View: 3. Drag and drop TEST-BTY0:STRING from the PVs list to the plot and check that there is a point every 5 minutes in the historic data 4. Right-click on the plot and select the option Inspect Samples. In the view, select TEST-BTY0:STRING and check that the date of the first samples retrieved from the database (RDB) with a sample every 5 minutes. Then check that the live data are samples every second.	
Pass Criteria	1. the IOC date & time is archived every 5 minutes  TEST-BTY0:STRING | Connected | 5.00 min scan, max. 60 repeats 3. Points every 5 minutes for historic data and continuous points for live data:	



4. Historic samples from RDB every 5 minutes and live data every second:

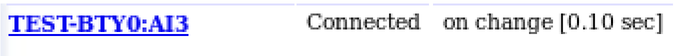
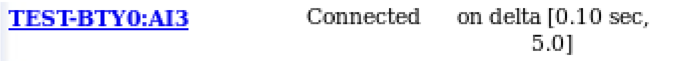


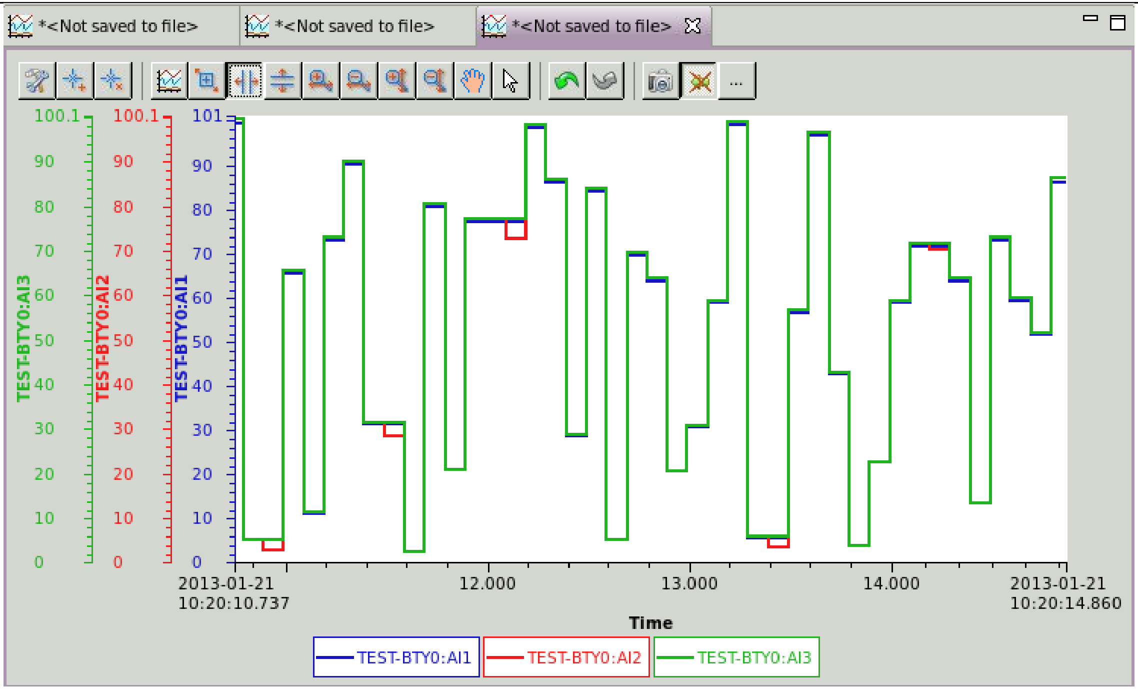
Item (PV, Formula): TEST-BTY0:STRING

Time	Value	Severity	Status	Source
2013/01/28 13:32:38.178075731	28/01/2013 13:32:38	NONE	NO_ALARM	RDB
2013/01/28 13:37:09.209555496	28/01/2013 13:37:09	NONE	NO_ALARM	RDB
2013/01/28 13:37:32.212037383	28/01/2013 13:37:32	NONE	NO_ALARM	Live Data
2013/01/28 13:37:33.212154762	28/01/2013 13:37:33	NONE	NO_ALARM	Live Data
2013/01/28 13:37:34.212289873	28/01/2013 13:37:34	NONE	NO_ALARM	Live Data
2013/01/28 13:37:35.217357376	28/01/2013 13:37:34	NONE	NO_ALARM	Live Data

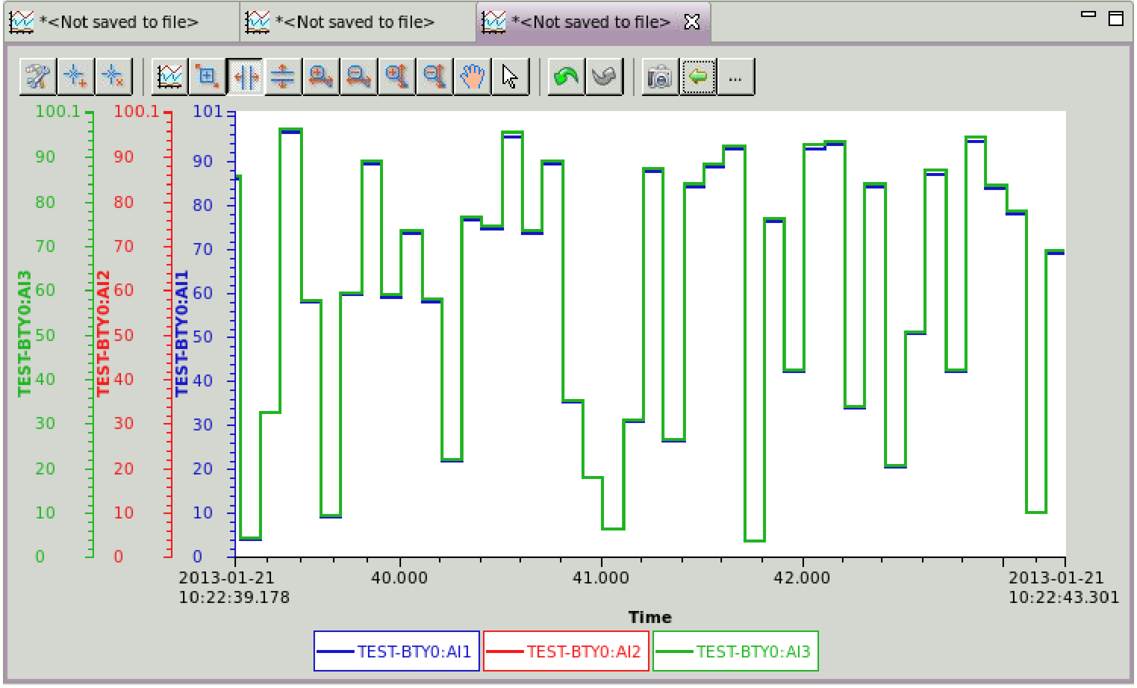
ENG-06	**3.4.9 Archiving Monitor with Threshold Mode**
Prerequisite	1. Demo IOC running 2. Archive Configuration imported 3. Archive Engine started
Test Cases	1. Using the Archive Monitor with Threshold mode, each value received is compared by the Archive Engine to the previous one and stored only if the change is greater than the dead band: • The Archive Engine performs the dead band checks in place of the Channel Access server, As this mode is not yet integrated in SDD, it is proposed to setup it manually for test purposes for the temperature TEST-BTY0:AI3 with a threshold of 5C.
Procedure	First, from the Archive Engine Group TEST-BTY0 monitoring web page: 1. Check the current archiving Mechanism defined with SDD for TEST-BTY0:AI3 Change the configuration to include the test of Archiving Monitor with Threshold Mode, import

	the new configuration and restart the Archive Engine:
	2. $ gedit src/main/beauty/TEST-BTY0-beauty.xml&
	3. in the Editor retrieve the PV TEST-BTY0:AI3 configuration and add a threshold of 5 to the monitor mode:
	`<channel >`
	`<name >TEST-BTY0:AI3</name>`
	`<period >0.1</period>`
	`<monitor >5</monitor>`
	`</channel>`
	Save the new xml configuration file. Then from the Linux console, import the updated configuration:
	4. $archive-configtool -engine demo -description 'Demo Test Engine' -port 5812 -import -config src/main/beauty/TEST-BTY0-beauty.xml -replace_engine
	5. From the Web Interface, enter http://localhost:5812/restart to restart the Archive Engine with the new configuration. Then select again the main page http://localhost:5812/main and check Groups -> TEST-BTY0 or click directly on the following link http://localhost:5812/group?name=TEST-BTY0 . Check the new archiving mode of the temperature AI3.
	In CSS, open a new plot:
	6. Click on button Open a new Data Browser plot in the menu bar
	In CSS, from the Archive Search View:
	7. Drag and drop TEST-BTY0:AI1, AI2 and AI3 from the PVs list to the plot and check that there is a point every 5 minutes in the historic data
	8. Right-click on the plot and select the option Open Properties Panel. In the Time Axis tab, change the Start time to -1m and enter to validate the new time scale. If needed, zoom horizontally the beginning of the plot to check if samples from RDB are the same for AI1 and AI3, the first temperature has the threshold of 5C defined on the IOC and the latter one has the threshold controlled by the archive engine.	

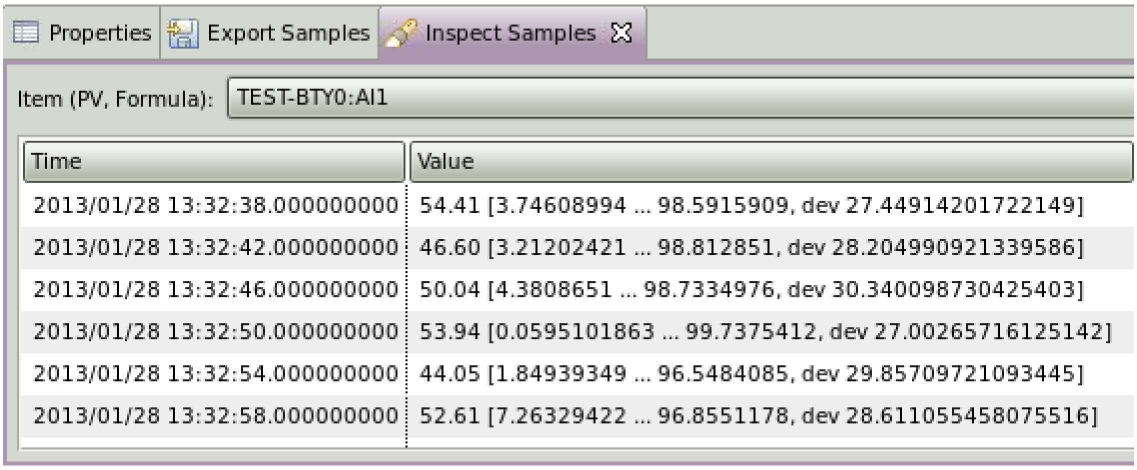
Pass Criteria	1. The temperature is monitored at 10Hz:  TEST-BTY0:AI3 Connected on change [0.10 sec] 5. The new archiving mode for the temperature AI3 should be:  TEST-BTY0:AI3 Connected on delta [0.10 sec, 5.0] 8. AI1 with the IOC threshold ADEL of 5C and AI3 with the archive engine threshold of 5C should be the same and slightly different from AI2 which has no threshold:	



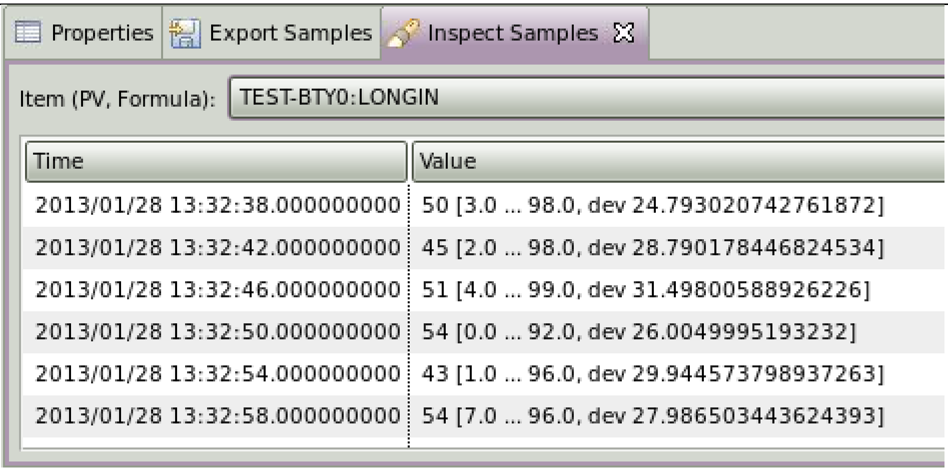
The live data of the 3 temperatures are identical as the data browser monitor directly the PV values without any threshold:



ENG-07	**3.4.10 Archived Data Types**
Prerequisite	1. Demo IOC running 2. Archive Configuration imported 3. Archive Engine started
Test Cases	1. Using CSS, plot of different archived data types are plotted such as double, integer, string or waveform.

<table>
<tr><td>Procedure</td><td>In CSS, open a new plot:
1. Click on button Open a new Data Browser plot in the menu bar
2. In CSS, from the Archive Search View with Pattern “TEST-BTY0:*”, drag and drop TEST-BTY0:AI1 in the plot area. Right-click on the plot and select Inspect Samples. Select the Item TEST-BTY0:AI1 and check that the archived data type is double.
3. Open a new Data Browser plot in the menu bar
4. In CSS, from the Archive Search View, drag and drop TEST-BTY0:LONGIN in the plot area. From the Inspect Samples View, select the Item TEST-BTY0:LONGIN and check that the archived data type is integer.
5. Open a new Data Browser plot in the menu bar
6. In CSS, from the Archive Search View, drag and drop TEST-BTY0:MBBI in the plot area. From the Inspect Samples View, select the Item TEST-BTY0:MBBI and check that the archived data type is an enumerated string.
7. Open a new Data Browser plot in the menu bar
8. In CSS, from the Archive Search View, drag and drop TEST-BTY0:STRING in the plot area. From the Inspect Samples View, select the Item TEST-BTY0:SRTING and check that the archived data type is a string that contains the current date and time.
9. Open a new Data Browser plot in the menu bar
10. In CSS, from the Archive Search View, drag and drop TEST-BTY0:WAVEFORM in the plot area. From the Inspect Samples View, select the Item TEST-BTY0: WAVEFORM and check that the archived data type is an array. Then make a right-click in the plot and select the option Inspect Waveforms. Select the Item TEST-BTY0: WAVEFORM and check the waveform. You can use the horizontal scrollbar to display the next 50 elements of the array.</td><td></td></tr>
<tr><td>Pass Criteria</td><td>2. The archived value type should be double:


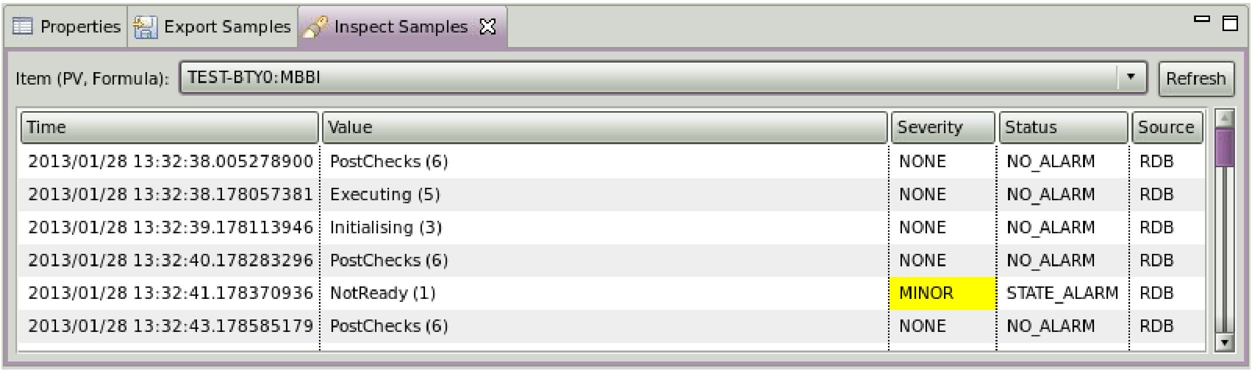
Item (PV, Formula): TEST-BTY0:AI1
<table>
<tr><th>Time</th><th>Value</th></tr>
<tr><td>2013/01/28 13:32:38.000000000</td><td>54.41 [3.74608994 ... 98.5915909, dev 27.44914201722149]</td></tr>
<tr><td>2013/01/28 13:32:42.000000000</td><td>46.60 [3.21202421 ... 98.812851, dev 28.204990921339586]</td></tr>
<tr><td>2013/01/28 13:32:46.000000000</td><td>50.04 [4.3808651 ... 98.7334976, dev 30.340098730425403]</td></tr>
<tr><td>2013/01/28 13:32:50.000000000</td><td>53.94 [0.0595101863 ... 99.7375412, dev 27.00265716125142]</td></tr>
<tr><td>2013/01/28 13:32:54.000000000</td><td>44.05 [1.84939349 ... 96.5484085, dev 29.85709721093445]</td></tr>
<tr><td>2013/01/28 13:32:58.000000000</td><td>52.61 [7.26329422 ... 96.8551178, dev 28.611055458075516]</td></tr>
</table>
4. The archived value type should be integer:</td><td></td></tr>
</table>



Item (PV, Formula): TEST-BTY0:LONGIN

Time	Value
2013/01/28 13:32:38.000000000	50 [3.0 ... 98.0, dev 24.793020742761872]
2013/01/28 13:32:42.000000000	45 [2.0 ... 98.0, dev 28.790178446824534]
2013/01/28 13:32:46.000000000	51 [4.0 ... 99.0, dev 31.49800588926226]
2013/01/28 13:32:50.000000000	54 [0.0 ... 92.0, dev 26.0049995193232]
2013/01/28 13:32:54.000000000	43 [1.0 ... 96.0, dev 29.944573798937263]
2013/01/28 13:32:58.000000000	54 [7.0 ... 96.0, dev 27.986503443624393]

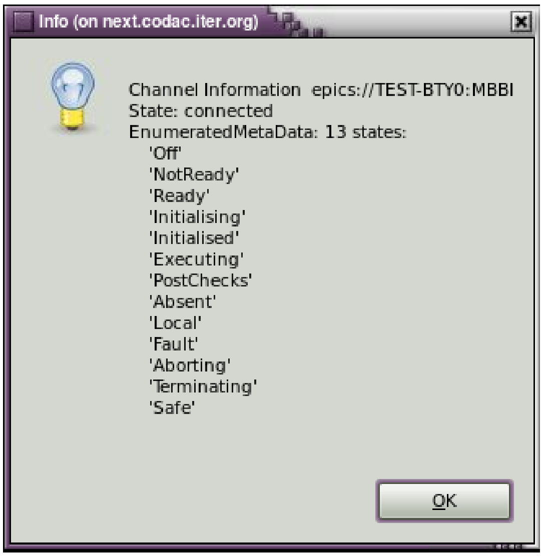
6. The archived value type should be a string:



Item (PV, Formula): TEST-BTY0:MBBI

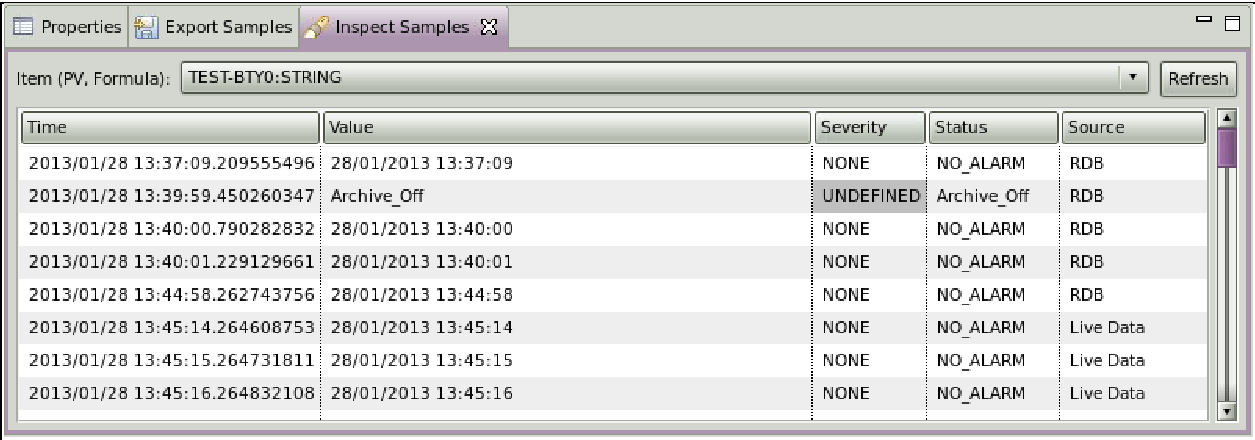
Time	Value	Severity	Status	Source
2013/01/28 13:32:38.005278900	PostChecks (6)	NONE	NO_ALARM	RDB
2013/01/28 13:32:38.178057381	Executing (5)	NONE	NO_ALARM	RDB
2013/01/28 13:32:39.178113946	Initialising (3)	NONE	NO_ALARM	RDB
2013/01/28 13:32:40.178283296	PostChecks (6)	NONE	NO_ALARM	RDB
2013/01/28 13:32:41.178370936	NotReady (1)	MINOR	STATE_ALARM	RDB
2013/01/28 13:32:43.178585179	PostChecks (6)	NONE	NO_ALARM	RDB

From the following enumeration:



Channel Information epics://TEST-BTY0:MBBI
State: connected
EnumeratedMetaData: 13 states:
'Off'
'NotReady'
'Ready'
'Initialising'
'Initialised'
'Executing'
'PostChecks'
'Absent'
'Local'
'Fault'
'Aborting'
'Terminating'
'Safe'

8. The archived value type should be a string yyyy/mm/dd hh:mm:ss archived every 5 minutes:

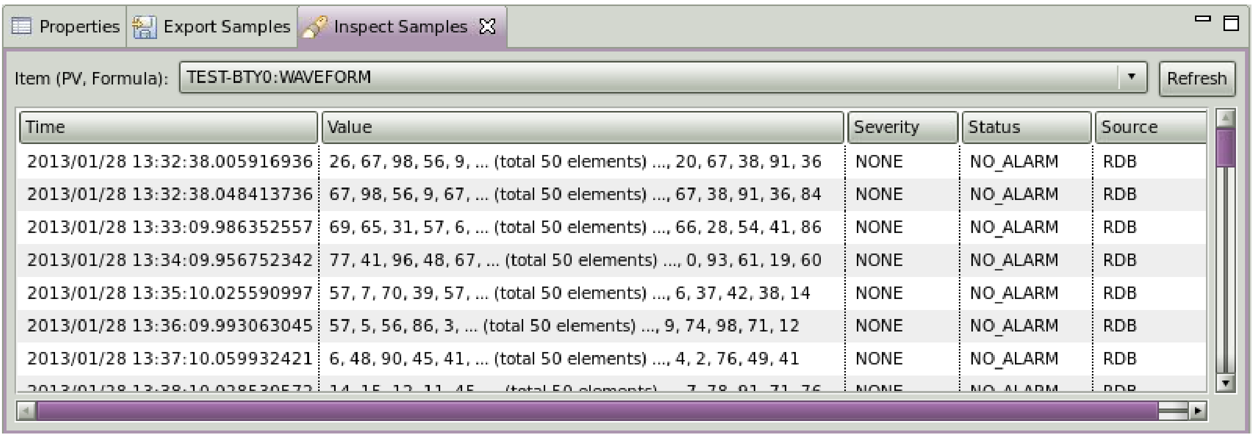


Item (PV, Formula): TEST-BTY0:STRING

Time	Value	Severity	Status	Source
2013/01/28 13:37:09.209555496	28/01/2013 13:37:09	NONE	NO_ALARM	RDB
2013/01/28 13:39:59.450260347	Archive_Off	UNDEFINED	Archive_Off	RDB
2013/01/28 13:40:00.790282832	28/01/2013 13:40:00	NONE	NO_ALARM	RDB
2013/01/28 13:40:01.229129661	28/01/2013 13:40:01	NONE	NO_ALARM	RDB
2013/01/28 13:44:58.262743756	28/01/2013 13:44:58	NONE	NO_ALARM	RDB
2013/01/28 13:45:14.264608753	28/01/2013 13:45:14	NONE	NO_ALARM	Live Data
2013/01/28 13:45:15.264731811	28/01/2013 13:45:15	NONE	NO_ALARM	Live Data
2013/01/28 13:45:16.264832108	28/01/2013 13:45:16	NONE	NO_ALARM	Live Data

Note: when the archive engine was restarted, a message "Archive_Off" was written in the database.

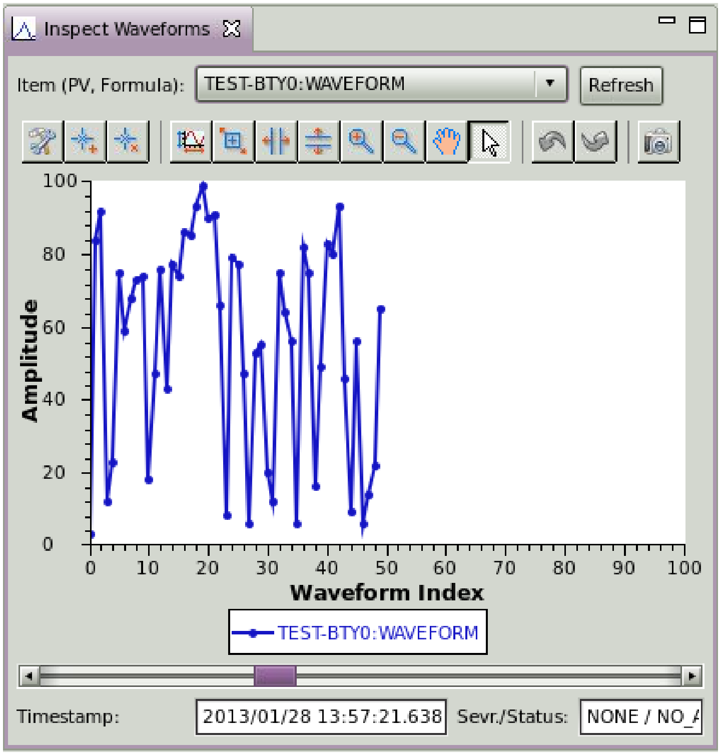
10. The archived data should be an array of maximum 50 elements:



Item (PV, Formula): TEST-BTY0:WAVEFORM

Time	Value	Severity	Status	Source
2013/01/28 13:32:38.005916936	26, 67, 98, 56, 9, ... (total 50 elements) ..., 20, 67, 38, 91, 36	NONE	NO_ALARM	RDB
2013/01/28 13:32:38.048413736	67, 98, 56, 9, 67, ... (total 50 elements) ..., 67, 38, 91, 36, 84	NONE	NO_ALARM	RDB
2013/01/28 13:33:09.986352557	69, 65, 31, 57, 6, ... (total 50 elements) ..., 66, 28, 54, 41, 86	NONE	NO_ALARM	RDB
2013/01/28 13:34:09.956752342	77, 41, 96, 48, 67, ... (total 50 elements) ..., 0, 93, 61, 19, 60	NONE	NO_ALARM	RDB
2013/01/28 13:35:10.025590997	57, 7, 70, 39, 57, ... (total 50 elements) ..., 6, 37, 42, 38, 14	NONE	NO_ALARM	RDB
2013/01/28 13:36:09.993063045	57, 5, 56, 86, 3, ... (total 50 elements) ..., 9, 74, 98, 71, 12	NONE	NO_ALARM	RDB
2013/01/28 13:37:10.059932421	6, 48, 90, 45, 41, ... (total 50 elements) ..., 4, 2, 76, 49, 41	NONE	NO_ALARM	RDB
2013/01/28 13:38:10.028530573	14, 15, 12, 11, 45, ... (total 50 elements) ..., 7, 78, 91, 71, 76	NONE	NO_ALARM	RDB

The waveform should look like that:



Item (PV, Formula): TEST-BTY0:WAVEFORM

Timestamp: 2013/01/28 13:57:21.638 Sevr./Status: NONE / NO_A

PRF-01

3.4.11 6KSamples/s archived

Prerequisite

An important disk space is required as 2 x 600 PVs at 10HZ are archived at 0.1sec by 2 archive engines.

1. Performance IOCs downloaded from SVN

From a new Linux console, start the EPICS IOC Performance Databases:

2. $ softIoc src/main/epics/SharedTemplateApp/Db/rndmIOC-all-start.cmd

From a new Linux console, Start the 2 Archives Engines:

3. $ src/main/beauty/rndmArchive-X-3K-start.cmd

4. $ src/main/beauty/rndmArchive-Y-3K-start.cmd

Test Cases

1. Using Archive Engine Monitoring web interface, check there is no overrun after some hours or days.

Procedure

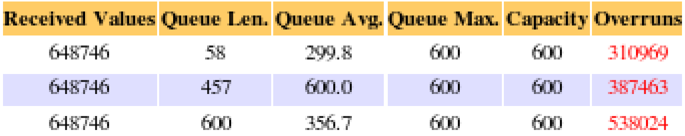
In a Linux console, open the web interface of the Archive Engine X:

1. $ firefox http://localhost:4912/main & and check the status of the archived PV by clicking on the links Groups -> Archive Engine X – IOC A and B

2. In the Web Browser, open a new tab and entre the URL of the second Archive Engine Y http://localhost:4914/main & and check the status of the archived PV by clicking on the links Groups -> Archive Engine Y – IOC C and D

Pass Criteria

1-2. On a virtual machine after “some” hours, there will be overruns:



Received Values	Queue Len.	Queue Avg.	Queue Max.	Capacity	Overruns
648746	58	299.8	600	600	310969
648746	457	600.0	600	600	387463
648746	600	356.7	600	600	538024

In order to know when the first overrun occurs, the log files should be checked:

```
$ cat ~/.css/archive-engine-Archive_Engine_X/console.log | grep 'overruns'
2013-01-10 17:09:43.708 WARNING [Thread 32]
org.csstudio.archive.engine.ThrottledLogger (log) - TEST-F001:rndmAx1: 216 overruns
$ cat ~/.css/archive-engine-Archive_Engine_Y/console.log | grep 'overruns'
2013-01-10 17:09:43.714 WARNING [Thread 30]
org.csstudio.archive.engine.ThrottledLogger (log) - TEST-F001:rndmCx1: 134 overruns
```

As the 2 Archive Engines X & Y were started at 14:06:

Start Time	2013/01/10 14:06:29

This means in this case that the first overruns occurs after 3 hours and so the rate of 6KSamples/sec was not achieved.

To terminate the tests, stop all the IOCs and the 3 demo Archive Engines. Close css and firefox:

1. $ epics> exit

From the archive engine web monitoring interface:

2-1. http://localhost:5812/stop

2-2. http://localhost:4912/stop

2-3. http://localhost:4914/stop

3. Close CSS using the menu File -> Exit. Do not save the plt configuration files.

4. $ archive-configtool -engine demo -delete_config

```
Deleted engine config 'demo'
```

5. Close firefox

3.5 Component Test Log

CFG-01	**3.5.1 Archive Configuration Import**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
CFG-02	**3.5.2 Archive Configuration Export**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-01	**3.5.3 Archive Engine Startup**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-02	**3.5.4 Archive Engine Monitoring**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-03	**3.5.5 Event Driven Archiving**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-04	**3.5.6 Archiving Monitor Mode**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		

DSP-01	**3.5.7 Historical Data Plot**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-05	**3.5.8 Archiving Scan Mode**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-06	**3.5.9 Archiving Monitor with Threshold Mode**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-06	**3.5.10 Archived Data Types**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		
ENG-06	**3.5.11 6KSamples/s archived**	[PASS / FAIL]
[Bug ID]	[Bug title to briefly describe the anomaly]	
Remarks		

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?	
10i Test schedules, Requirements traceability (or verification matrix)?	

Criteria	Yes / No / NA
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?	