Table of Contents

My notes on XDAQ/XMAS monitoring

This documents configuration of monitoring of a simple quantity (`EventCounter` or `StateName`) of the TTCci application.

- make sure /etc/slp.conf has the correct server address, at the time of writing, this was:

```
net.slp.DAAddresses = 10.176.29.112
```

- make sure the slp daemon is running (if not, start it with `sudo /etc/init.d/slp start`)
- in the application tag (e.g. for RCT):
  - changed the group from 'profile' to 'ttc' for the TTCci application.
  - add a tag publish="true"
  - add a tag service="TTCci_RCT"
- make sure you run the xplore application in the same xdaq process, copy e.g. /opt/xdaq/share/ttczone/profile/ttc-mi.profile as profile and use it.
- Check with

```
slptool -i $(hostname -f) findsrvs service:peer:http  | grep -i vmepcs2b16-01
```

that you see the corresponding applications published.

- The profile about also has a sensor application in it. Click on the sensor application and select the 'Infospaces' tab. You should see all items in infospaces, e.g. `StateName` or `EventCounter`.

- create the flash list: copy e.g. ttczone/flash/ttc-mi-monitoring.flash to ttczone/flash/ttc-rct-monitoring.flash . Change infospace=".." to infospace="xdaq-application:lid=40" .
- change the id attribute of the xmas:flash tag
- copy ttczone/sensor/ttc-mi-monitoring.sensor to ttczone/sensor/ttc-rct-monitoring.sensor . Edit the file by setting the location of the flash file to the location of ttczone/flash/ttc-rct-monitoring.flash . Don't forget to change also the part after the # sign with the id you have put in ttc-rct-monitoring.flash .
- In the profile mentioned above, the sensor application is configured to look for sensor files in ttczone/sensor (the autoConfSearchPath tag). Remove this and add the explicit location of the sensor file (like in ttczone/profile/general-sensor.profile). Make sure that there is a publish tag in this file !! (e.g. copy and adapt from ttczone/profile/general-sensor.profile).
- restart the xdaq process with the sensor.
- Go to the sensor application on ttc-trigger and check whether you see a line `urn:xdaq-flashlist:ttc_mi_monitoring` in the 'flashlists' tab.
- on ttc-mi (where the pulser lives), add the new flash list to ttczone/pulser/ttc.pulser and restart the pulser.
- go to the pulser application (at the time of writing it was here). Click on the Wait until the 'pulse events' tab, wait until the flashlist ttc_rct_monitoring/ttc has a non-zero time (i.e. is not 01-01-1970) by periodically reloading. You should then see the flashlist also appear in the 'statistics' and 'sensors' tab.

- make sure that the sensor publishes to the correct group (ttc at the time of writing) !
- In our setup, in order to propagate from the 'ws-eventing' for the group ttc to general, make sure you add the flashlist also in /ttczone/sensor/general-sensor.sensor !!! (otherwise, the flashlist does not appear on the las...)

- ALSO do not forget to add the flash list to the las !

- how to declare that the flash list should take the quantity from the ParameterQuery/infospace ? Put a sensor in the same xdaq process (?!)
- After adding a quantity to the flash list, restart the application containing the sensor and restart the general-las (on ttc-mi).

Modified/added files after I got this to work:

- added files on ttc-trigger (i.e. subsystem specific):

```
ttczone/flash/ttc-rct-monitoring.flash
ttczone/sensor/ttc-rct-monitoring.sensor
ttczone/profile/ttc-trigger.profile
```

- modified files (on ttc-trigger, i.e. subsystem specific):

```
ttczone/conf/ttc-rct-ttcci.conf
ttczone/conf/ttc-rct.xml
```

- modified files (on ttc-mi, i.e. general monitoring):

```
ttczone/sensor/general-las.las
ttczone/sensor/general-sensor.sensor
ttczone/pulser/ttc.pulser
```

SLP (srvloc) debugging

```
tshark -i eth0 -V -R srvloc udp
grep Predicate: /tmp/log
grep 'Internet Protocol, Src:' /tmp/log | cut -d\  -f4 | sort -u
```

Example SLP communication

on ttc-mi:

- xdaq.exe on port 5007 (xmas::store ?) sends "Service Request"

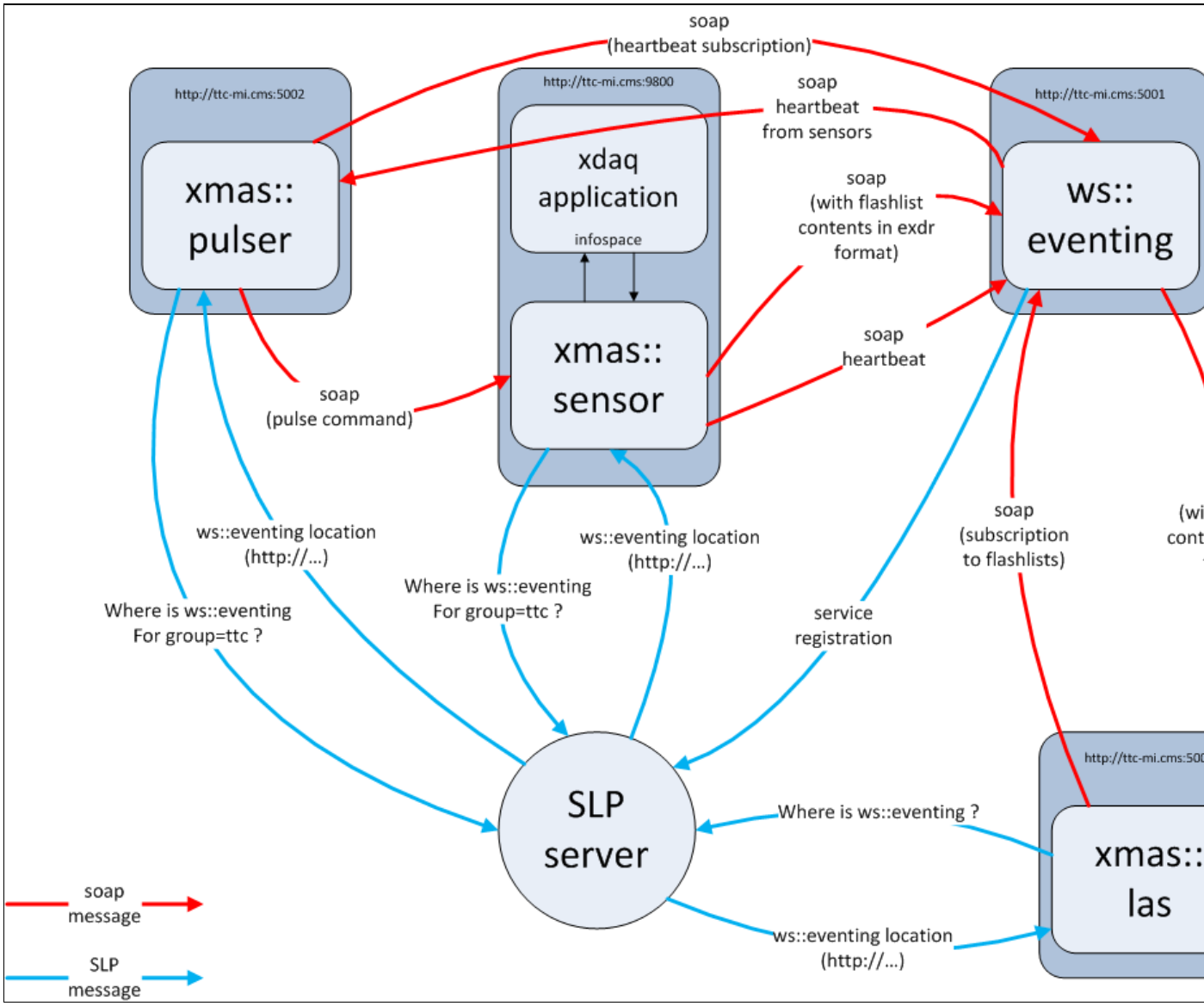
```
(&(|(zone=*ttczone*))(|(&(service=ws-eventing)(|(group=*database*)(group=*general*)))(&(se
```

gets back:Service Reply from srv-C2C03-20.cms (10.176.29.112):

```
URL: service:peer:http://vmepcs2b16-05.cms:5003/urn:xdaq-application:lid=400
```

Some insights from wiresharking and stracing

General picture (pdf):



Pulser actions

The pulser seems to send SOAP messages like the following to the sensor (not the cell !):

```
<soap-env:Envelope
  soap-env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header>
    <wsa:to xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://vmepcs2b16-05.cms
  </soap-env:Header>
  <soap-env:Body>
    <xmas:pulse xmlns:xmas="http://xdaq.web.cern.ch/xdaq/xsd/2006/xmas-10">
      <xmas:sample flashlist="urn:xdaq-flashlist:ttc_mi_monitoring" tag="ttc"/>
    </xmas:pulse>
  </soap-env:Body>
</soap-env:Envelope>
```

- Discovery of the ws::eventing: when stracing, it can be that you'll have to wait a minute or so until you see the corresponding request to be sent to the SLP daemon.

- The pulser seems to talk to the ws::eventing (after discovering it via SLP):

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/" soap-env:encodingSt
```

```
  <soap-env:Header>
    <wsa:Action xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://schemas.xmlso
    <wsa:ReplyTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://vmepcs2b16-0
    <wsa:To xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://vmepcs2b16-05.cms
  </soap-env:Header>
  <soap-env:Body>
    <wse:Subscribe xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
      <wse:EndTo>
        <wsa:Address xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://vmepcs2b
      </wse:EndTo>
      <wse:Delivery Mode="http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push">
        <wse:NotifyTo>
          <wsa:Address xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://vmepcs
        </wse:NotifyTo>
      </wse:Delivery>
      <wse:Expires>PT2M</wse:Expires>
      <wse:Filter Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">//xmas:heartbeat</wse:Fi
    </wse:Subscribe>
  </soap-env:Body>
</soap-env:Envelope>
```

Sensor actions

- Starting up an application with a sensor e.g. produces the following SLP requests:
  - URL: service:peer:http://vmepcs2b16-01.cms:2000/urn:xdaq-application:lid=11

    ,

```
  (class=xmas::sensor::Application),
  (className=xmas::sensor::Application),
  (context=http://vmepcs2b16-01.cms:2000),
  (group=ttc),(hasInstance=false),
  (icon=/xmas/sensor/images/Sensor.png),
  (icon16x16=/xmas/sensor/images/Sensor_16x16.ico),
  (id=11),
  (instance=0),
  (network=local),
  (publish=true),
  (service=sensor),
  (uuid=d8ad07fb-ecb5-4804-8a0f-e9b40ed6ec5a),
  (zone=ttczone),
  (domain=xdaq)
```

which seems to be the publication of the corresponding sensor application to the SLP daemon.

- The sensor discovers the ws-eventing using SLP (seen in strace),

see also the sensor's log:

```
 INFO  ttczone.cms.vmepcS2B16-01.p:2000.xmas::sensor::Application.lid(11)
```

- The sensor seems to send heartbeat SOAP messages to the ws-eventing:

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/" soap-env:encodingSt
  <soap-env:Header/>
  <soap-env:Body>
    <xmas:heartbeat xmlns:xmas="http://xdaq.web.cern.ch/xdaq/xsd/2006/xmas-10" xmas:url="http://v
      <xmas:item name="class" value="xmas::sensor::Application"/>
      <xmas:item name="className" value="xmas::sensor::Application"/>
      <xmas:item name="context" value="http://vmepcs2b16-05.cms:9800"/>
      <xmas:item name="group" value="ttc"/>
      <xmas:item name="hasInstance" value="false"/>
      <xmas:item name="icon" value="/xmas/sensor/images/Sensor.png"/>
```

```
      <xmas:item name="icon16x16" value="/xmas/sensor/images/Sensor_16x16.ico"/>
      <xmas:item name="id" value="11"/>
      <xmas:item name="instance" value="0"/>
      <xmas:item name="network" value="local"/>
      <xmas:item name="publish" value="true"/>
      <xmas:item name="service" value="sensor"/>
      <xmas:item name="uuid" value="1203f813-c56c-4f95-941b-55aba9ebc317"/>
    </xmas:heartbeat>
  </soap-env:Body>
</soap-env:Envelope>
```

- The sensor seems to send the flashlist contents to the ws-eventing (with a multi part message where one part is the following soap message and another part is the flashlist content in exdr format):

```
<soap-env:Envelope
  soap-env:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/">
  <soap-env:Header>
    <wsa:Action xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">urn:xdaq-application
    <wsa:To xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">http://vmepcs2b16-05.cms
  </soap-env:Header>
  <soap-env:Body>
    <xmas:report xmlns:xmas="http://xdaq.web.cern.ch/xdaq/xsd/2006/xmas-10" xmlns:xmas-sensor="ht
      <xmas:sample flashlist="urn:xdaq-flashlist:ttc_rct_monitoring" originator="http://vmepcs2b1
    </xmas:report>
  </soap-env:Body>
</soap-env:Envelope>
```

LAS actions

- The las says such things as:

```
INFO  ttczone.cms.vmepcS2B16-05.p:5005.xmas::las::Application.instance(0)
```

which is the `ws-eventing` URL.

- The las seems to discover the ws-eventing via SLP

and then sends a soap subscription request to the ws-eventing:

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/" soap-env:encodingSt
  <soap-env:Header>
    <wsa:Action xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
      http://schemas.xmlsoap.org/ws/2004/08/eventing/Subscribe
    </wsa:Action>
    <wsa:ReplyTo xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
       http://vmepcs2b16-05.cms:5003/urn:xdaq-application:lid=400
    </wsa:ReplyTo>
    <wsa:To xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
       http://vmepcs2b16-05.cms:5003/urn:xdaq-application:lid=400
    </wsa:To>
  </soap-env:Header>
  <soap-env:Body>
    <wse:Subscribe xmlns:wse="http://schemas.xmlsoap.org/ws/2004/08/eventing">
      <wse:EndTo>
        <wsa:Address xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
           http://vmepcs2b16-05.cms:5005/urn:xdaq-application:lid=10
         </wsa:Address>
      </wse:EndTo>
      <wse:Delivery Mode="http://schemas.xmlsoap.org/ws/2004/08/eventing/DeliveryModes/Push">
        <wse:NotifyTo>
          <wsa:Address xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">
            http://vmepcs2b16-05.cms:5005/urn:xdaq-application:lid=10
```

```
          </wsa:Address>
        </wse:NotifyTo>
      </wse:Delivery>
      <wse:Expires>PT2M</wse:Expires>
      <wse:Filter Dialect="http://www.w3.org/TR/1999/REC-xpath-19991116">
          //xmas:sample[((@flashlist='urn:xdaq-flashlist:ttc_mi_monitoring')and(contains(@tag,'tt
      </wse:Filter>
    </wse:Subscribe>
  </soap-env:Body>
</soap-env:Envelope>
```

where it seems to know which flashlists it should look at by reading the file `/opt/xdaq/share/ttczone/sensor/general-las.las`.

ws::eventing actions

- The ws-eventing seems to push the flashlist contents to the las via a multi-part SOAP message (with the flashlist contents in exdr format in the second part), the first part is e.g.:

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/" soap-env:encodingSt
  <soap-env:Header>
    <wsa:Action xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">urn:xdaq-application
    <wsa:To xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">vmepcs2b16-05.cms:5005</
  </soap-env:Header>
  <soap-env:Body>
    <xmas:report xmlns:xmas="http://xdaq.web.cern.ch/xdaq/xsd/2006/xmas-10" xmlns:xmas-sensor="ht
      <xmas:sample flashlist="urn:xdaq-flashlist:ttc_mi_monitoring" originator="http://vmepcs2b16
    </xmas:report>
  </soap-env:Body>
</soap-env:Envelope>
```

- It also seems to 'forward' heartbeat messages received from the sensors to the pulser:

```
<soap-env:Envelope xmlns:soap-env="http://schemas.xmlsoap.org/soap/envelope/" soap-env:encodingSt
  <soap-env:Header>
    <wsa:Action xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">urn:xdaq-application
    <wsa:To xmlns:wsa="http://schemas.xmlsoap.org/ws/2004/08/addressing">vmepcs2b16-05.cms:5002</
  </soap-env:Header>
  <soap-env:Body>
    <xmas:heartbeat xmlns:xmas="http://xdaq.web.cern.ch/xdaq/xsd/2006/xmas-10" xmas:url="http://v
      <xmas:item name="class" value="xmas::sensor::Application"/>
      <xmas:item name="className" value="xmas::sensor::Application"/>
      <xmas:item name="context" value="http://vmepcs2b16-05.cms:9800"/>
      <xmas:item name="group" value="ttc"/>
      <xmas:item name="hasInstance" value="false"/>
      <xmas:item name="icon" value="/xmas/sensor/images/Sensor.png"/>
      <xmas:item name="icon16x16" value="/xmas/sensor/images/Sensor_16x16.ico"/>
      <xmas:item name="id" value="11"/>
      <xmas:item name="instance" value="0"/>
      <xmas:item name="network" value="local"/>
      <xmas:item name="publish" value="true"/>
      <xmas:item name="service" value="sensor"/>
      <xmas:item name="uuid" value="ce1d0d3f-edf0-4be5-88cf-5862e65fb0dc"/>
    </xmas:heartbeat>
  </soap-env:Body>
</soap-env:Envelope>
```

Links

- TS Alarms Monitoring Hands on Workshop
- Functions understood in flashlists seem to be interpreted in xmas/sensor/src/common/FlashListMonitor.cc. At the time of writing, this seems to know the

following functions:

systime()
context()
uuid()
session()
zone()
hostname()

It also looks like that environment variables can be exported to flash lists when the variable name is prepended with dollar ($) sign.

-- AndreHolzner - 20 May 2009

This topic: Main > AndreHolznerXdaqMonitoring
Topic revision: r11 - 2009-08-20 - AndreHolzner