

DCA

This page is under construction

- [Overview](#)
- [Architectural diagram](#)
- [Process Flow](#)
- [Message Format from Setup Box](#)
- [STB Script Design](#)
- [API Documentation](#)

Overview

Data Collection and Analysis (DCA) is responsible for collecting the data from various log files and send to server. The script `dca_utility.sh` is in place on the set-top box. Based on the DCM response it will create a cron job in the setup box.

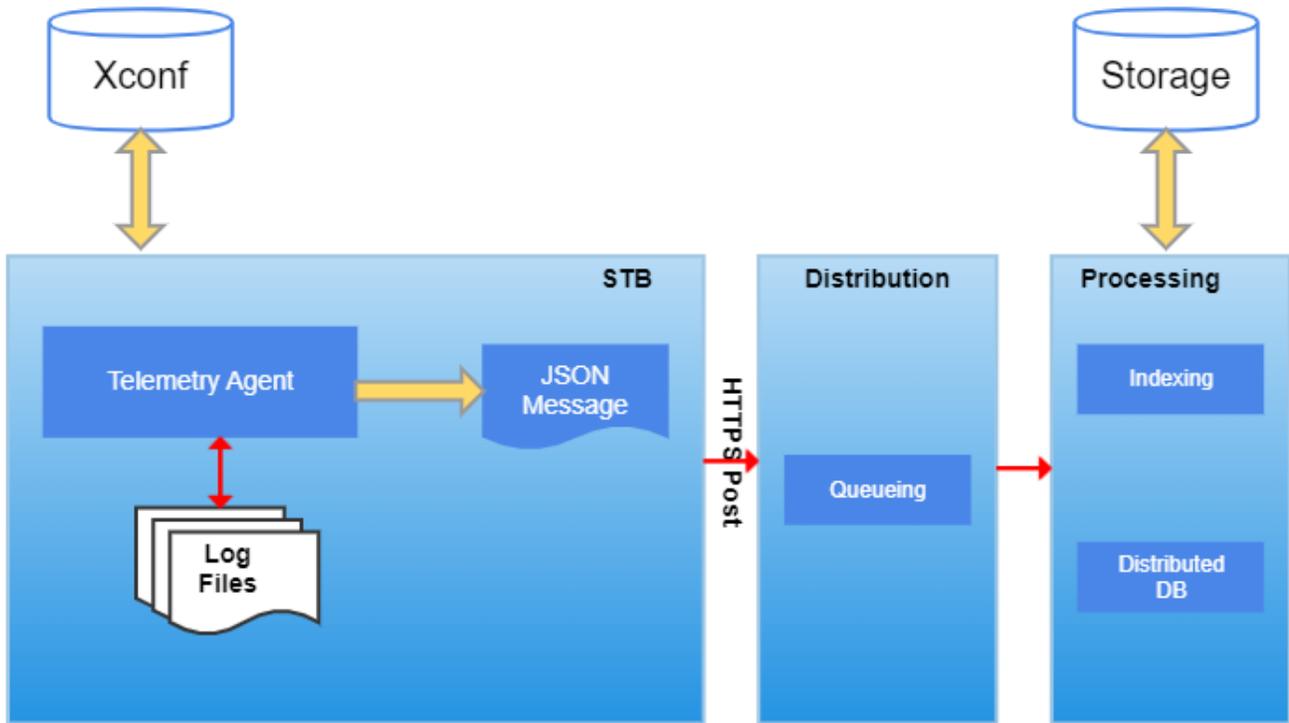
```
*/20 * * * * nice -n 20 sh /lib/rdk/dca_utility.sh 1
```

This cron job will send out the JSON message with the level one parameter values to the splunk server.

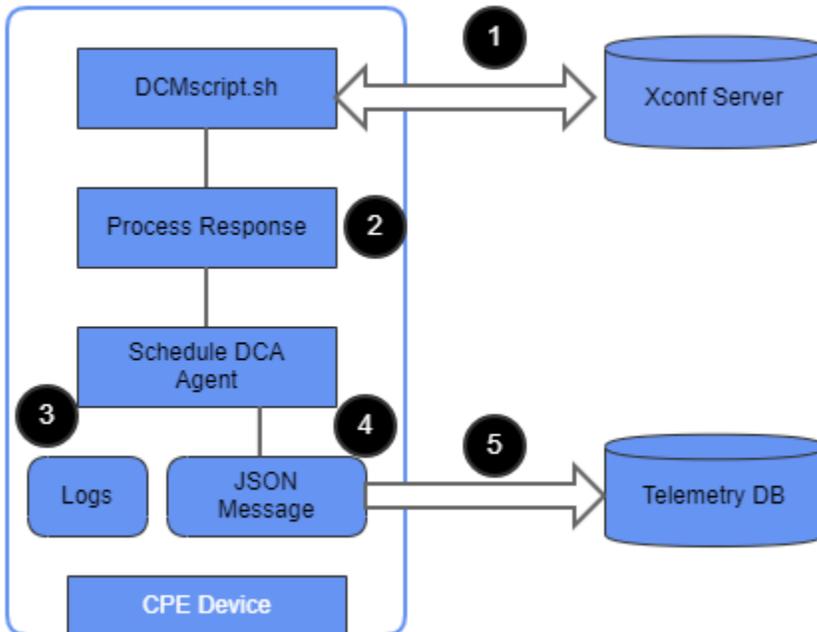
There are 3 level of logins message and upload to the server. Using S3 as the log back end for second and third level logging.

- **First Level**
 - Frequency - start with 5 minutes and then expand.
 - If log upload frequency is 5 minutes, that is separate from the log collection frequency, which will be based on various parameters.
 - Parameters could be counters.
 - Parameters may not be available in logs but may need real-time probing using snmp/tr-69 or other methods.
 - We will need collect records based on frequency at which collection needs to occur.
 - Log some events back to the cloud. Could be to S3 or another destination.
 - Records that it upload should be records that could be fed into Splunk without having to do much processing, for example, a name/value pair or another that is easy to expose on Splunk.
 - There could be multiple records since logging period.
 - Data collection frequency and upload frequency can be configured differently.
 - The level will be enabled for every device.
- **Second Level**
 - Medium number of logs. Full logs will need to be pushed.
 - Second and Third level are similar.
- **Third Level**
 - Similar to current method of full log uploads.

Architectural diagram



Process Flow



- DCMscript.sh communicates with Xconf server and fetches the predefined markers
- Using the markers, DCM Script will prepare a sorted map file for the log lookup, creates a CRON job for DCA Agent.
- CRON job retrieves data from the device using DCA which filters search patterns by looking into logs.
- From the retrieved data, it will create a JSON formatted message.
- JSON format data is uploaded to server

Message Format from Setup Box

Sample Setup box JSON Message

```
{ "searchResult": [{"Total Space available": "30"}, {"Heartbeat": "61"}, {"Success tune": "280"}, {"Failure tune": "0"}, {"mac": "14:D4:FE:65:A9:81"}, {"Version": "PX001AN_VBN_master_HYB_021622172015_PBTsd_RTT"}]}
```

Sample Message from XCONF DCM Response

```
{ "urn:settings:GroupName": "test", "urn:settings:CheckOnReboot": true, "urn:settings:CheckSchedule:cron": "47
9 * *
*", "urn:settings:CheckSchedule:DurationMinutes": 10, "urn:settings:CheckSchedule:StartDate": "2014-01-01
01:01:01", "urn:settings:CheckSchedule:EndDate": "2017-11-01
01:01:01", "urn:settings:LogUploadSettings:Message": null, "urn:settings:LogUploadSettings:Name": "test", "urn:
settings:LogUploadSettings:NumberOfDays": 0, "urn:settings:LogUploadSettings:UploadRepositoryName": "Amazon", "urn:
settings:LogUploadSettings:RepositoryURL": "http://logs.xcal.tv", "urn:settings:LogUploadSettings:UploadOnReboot":
true, "urn:settings:LogUploadSettings:upload": true, "urn:settings:LogUploadSettings:LogFiles": [{"id": "2609da75-
59ed-494f-96a0-369328be6a9c", "name": "logfile1", "deleteOnUpload": true}, {"id": "6cfb1b02-4147-4980-8e3c-
26754d14329e", "name": "logfile2", "deleteOnUpload": true}], "urn:settings:LogUploadSettings:LogFiles:StartDate": "", "
urn:settings:LogUploadSettings:LogFiles:EndDate": "", "urn:settings:LogUploadSettings:UploadSchedule:cron": "6
2 * *
*", "urn:settings:LogUploadSettings:UploadSchedule:levelone:cron": null, "urn:settings:LogUploadSettings:
UploadSchedule:leveltwo:cron": null, "urn:settings:LogUploadSettings:UploadSchedule:levelthree:cron": "3", "urn:
settings:LogUploadSettings:UploadSchedule:DurationMinutes": 20, "urn:settings:LogUploadSettings:UploadSchedule:
StartDate": "2014-01-01
01:01:01", "urn:settings:LogUploadSettings:UploadSchedule:EndDate": "2016-01-01
01:01:01", "urn:settings:VODSettings:Name": null, "urn:settings:VODSettings:LocationsURL": null, "urn:settings:
VODSettings:SRMIPList": null}
```

STB Script Design

Script will create temporary files split from the logs based on the seek value (Which keeps track of the point till we have looked in the logs). Once the strings are looked in the temporary file the script deletes the temporary file from the setup box. All the string/SNMP queries that setup box has to monitor is derived from the DCM response.

API Documentation

To know more about SoC/Application level APIs details use in RDK, refer the link [DCA API Documentation](#)