




FNV Sync Log Extraction Utility

Working Version Status	DRAFT
Working Version	v1.0

Document Status	IN PROGRESS
Delivery Target	FNV2
Epic JIRA	 ANLYT-1114 - Support a USB logging tool to replace fastdevlog CLOSED
Tracking JIRA	<div> ANLYT-1115 - Implement a utility on SYNC which will poll SFTP looking for files dropped into a dropbox CLOSED</div> <div> ANLYT-1724 - Implement proof-of-concept usb launcher support for launching USB log capture tool CLOSED</div>
Completion	
Approved Version	
Owner	Tam, Stephen (S.)
Reviewer (s)	Niemi, Jonathan (J.) Martin, Daryl (D.) Legault, Brian (B.) Unknown User (ashenfie) Vording, Glenn (G.) Peng, Jing (J.) Unknown User (awooste1) Sartori, Chris (C.) Yohan, Dylan (D.) Hodgson, Matt (M.) Jodhka, Jaspreet (J.)
Approver (s)	
Approve Date	
Link to HLD	

1. Glossary

2. Overview

The Log Extraction utility is a component on the FNV SYNC module that will copy requested logs from an SFTP location, when available to a specified output path.

The main use is to provide log extraction similar to FastDevLog util available on SYNC3.2v2, where logs are extracted to USB.

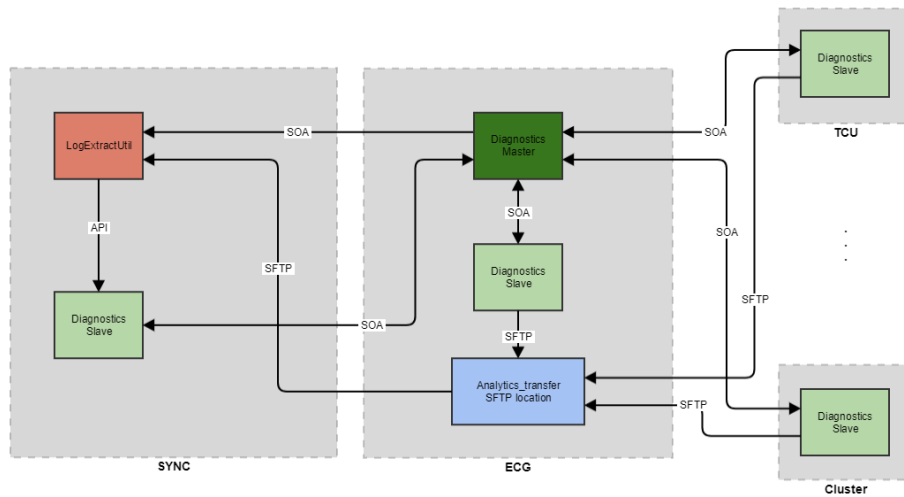
Existing Diagnostics process infrastructure will be used coordinate and capture logs for extraction to USB.

3. ECU Software Requirements

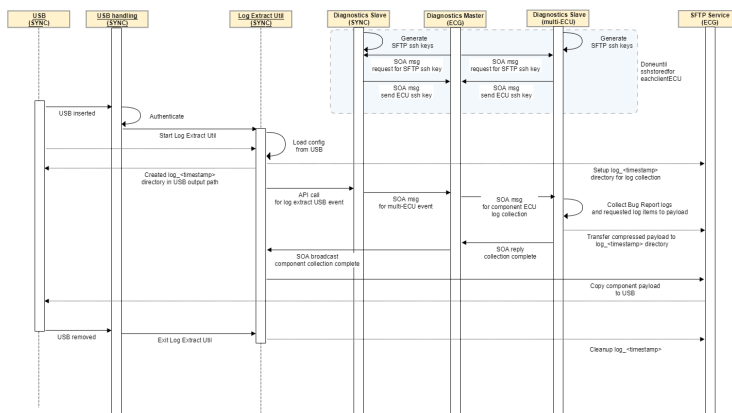
The Log Extraction Utility is meant to run on the SYNC.

- Access to a SOA broker in some way is critical since this is the only method of communication between the Log Extraction Utility and Diagnostics Master/Slaves running on other ECUs.
- Functional SOA API implementation in the SYNC build
- Functional [FNV File Transfer Service](#)

4. Components



LogExtractionDiagram



The log extraction to USB flow begins with the calling of the LogExtractUtil process.

The util tool then processes the config file to determine that log files and log folders to be captured for each ECU.

The util tool calls directly to the Diagnostics Slave to start the log capture and extraction to USB output path by utilizing the existing Diagnostics event multi-ECU logs capture architecture.

The SYNC Diagnostics slave then transmits all ecu log captures to the Diagnostics Master via SOA message to dispatch capture to the corresponding multi-ECU diagnostics slaves to process.

DIAGNOSTICS_EVENT_LOG_EXTRACT_USB has been added to the base event types to denote and store meta data status for the log extractions to USB.

quip_event_types.h

...

```
#define DIAGNOSTICS_EVENT_LOG_EXTRACT_USB 0x00000011
///< Show in Web Portal: No
///< Description: Denotes and event for log sextraction to USB
///< and is NOT transmitted OTA to server.
```

...

4.1. Config File

Example log_extract_config.json

```
{
  "extractList": {
    "component": [
      {
        "ecu": "SYNC",
        "logItem": [
          { "type": "file", "path": "/tmp/ss_sys_diag.log" },
          { "type": "folder", "path": "/fs/mp/sql/" },
          { "type": "folder", "path": "/fs/rwdata/" },
          { "type": "folder", "path": "/fs/tmpfs/MM/" },
          { "type": "folder", "path": "/tmp/VCA_LOGS/" },
          { "type": "file", "path": "/fs/tmpfs/touch_screen_log.txt" },
          { "type": "file", "path": "/fs/tmpfs"
/VS_DIAG_SERVICE_CONFIG_SHARED_MEM" },
          { "type": "folder", "path": "/fs/tmpfs/BT/" },
          { "type": "file", "path": "/tmp/errorlog.txt" },
          { "type": "file", "path": "/tmp/bt_cmn.log" },
          { "type": "file", "path": "/tmp/osapi.log" },
          { "type": "file", "path": "/tmp/TeleNav.log" },
          { "type": "file", "path": "/tmp/usb_launcher_otg.txt" },
          { "type": "file", "path": "/tmp/hwid" },
          { "type": "file", "path": "/fs/tmpfs/ns_perf.log" },
          { "type": "file", "path": "/fs/tmpfs/ns_perf.log.1" },
          { "type": "file", "path": "/fs/tmpfs/btpbk" },
          { "type": "file", "path": "/fs/tmpfs/btpbk_tmp" },
          { "type": "file", "path": "/fs/tmpfs/btsms" },
          { "type": "file", "path": "/fs/tmpfs/btdevicedata.txt" },
          { "type": "file", "path": "/fs/tmpfs/btdevicelist.txt" },
          { "type": "file", "path": "/fs/tmpfs/Traffic.db" },
          { "type": "file", "path": "/fs/tmpfs/FuelPrices.db" },
          { "type": "file", "path": "/fs/tmpfs/MovieListings.db" },
          { "type": "file", "path": "/fs/tmpfs/RdsTraffic.db" },
          { "type": "file", "path": "/fs/tmpfs/Traffic.db" },
          { "type": "file", "path": "/fs/tmpfs/OnlineTraffic.db" },
          { "type": "folder", "path": "/fs/tmpfs/sirius_ds_ota/" },
          { "type": "file", "path": "/fs/mp/etc/system/config/DefaultWCU.cfg" },
          { "type": "file", "path": "/fs/mp/etc/system/config/syncgen3_sys.cfg" },
          { "type": "file", "path": "/fs/mp/etc/system/config/Decoded_ODL.xml" }
        ]
      },
      {
        "ecu": "ECG",
        "logItem": [
          ...
        ]
      },
      {
        "ecu": "TCU",
        "logItem": [
          ...
        ]
      },
      ...
      {
        "ecu": "Cluster",
        "logItem": [
          ...
        ]
      }
    ]
  }
}
```

5. Interfaces of the SW Units

5.1. System Interfaces (outside SYNC ECUs)

Log Extraction Utility will use:

- SOA as its primary interface to Diagnostics Master on ECG
- SFTP to the Analytics_transfer SFTP location on ECG

5.1.1.1. SFTP SSH public key

Master will check to see whether it has an ssh public key for all ECUs saved locally.

For each ECU that does not have a ssh public key, it will request for one from the slave component.

Name	Direction	Purpose	Message Data protobuf
SFTP_PUB_KEY_REQUIRED_REQ	Master broadcast	Broadcast from Master notify which ECUs are required to send their SSH public key required for SFTP access.	<pre>message SftpPubKeyRequiredBroadcast { repeated string ecu = 1; }</pre>

Name	Direction	Purpose	Message Data protobuf
SFTP_PUB_KEY_AVAILABLE_REQ	Slave Master	Request providing the SFTP SSH public key from the ECU. Master will save the public key and add the key to auth for SFTP.	<pre>message SftpPubKeyAvailableReq { string ecu = 1; bytes publicKey = 2; }</pre>
SFTP_PUB_KEY_AVAILABLE_RSP	Master Slave(s)	Ack a SFTP_PUB_KEY_AVAILABLE_REQ	See "Generic Response"

5.1.1.2. 5.1.1.2. Logs payload read to download from SFTP

Master will broadcast when SFTP_LOGS_READY to gather logs.

Name	Direction	Purpose	Message Data protobuf
SFTP_PAYLOAD_READY	Master broadcast	Broadcast from Master to notify when all requested log collection from all ECUs are completed and uploaded to the SFTP location, ready to be downloaded.	<pre>message SftpPayload { string ecu = 1; repeated string payload = 2; } message SftpLogsReady { repeated SftpPayload payloadList = 1; }</pre>