****

****

*ICD and Design Document*

*Server*

**Mobile CEM Labs LLC**

**4006 Belt Line Road,   
Suite 115 Addison,   
Texas, 75001  
United States**

**TABLE OF CONTENTS**

**Contents -------------------------------------------------------------------------------------------Page No.**

[1 Introduction 4](#_Toc346996541)

[1.1 Scope 4](#_Toc346996542)

[1.2 Audience 4](#_Toc346996543)

[1.3 How This Document is Organized 4](#_Toc346996544)

[2 Architecture Overview 6](#_Toc346996545)

[3 Interface Definition 8](#_Toc346996547)

[3.1 Introduction 8](#_Toc346996549)

[3.2 File Format 8](#_Toc346996550)

[3.3 Message Information 8](#_Toc346996551)

[3.4 Message Detail Information 10](#_Toc346996552)

[4 Design Document 11](#_Toc346996553)

[4.1 Functional View 11](#_Toc346996555)

[4.2 Execution View 14](#_Toc346996556)

[4.3 Implementation Details 14](#_Toc346996557)

This page intentionally left blank

***Chapter***

*1*

Introduction

1. Scope

This document scope is to describe the interface with UE application that collect the data from diagnostics and upload to server. The document also detail out the Design for the Windows Server.

1. Audience
2. How This Document is Organized

| # | Chapter | Contents |
| --- | --- | --- |

Table 1: Chapter Overview

The acronyms and abbreviations used in this document are listed below

| Acronym | Explanation |
| --- | --- |

Table 2: Acronyms and Abbreviations

***Chapter***

*2*

Architecture Overview





The Windows Server Application (known as KPIGenService) executes as service which periodically checks if any files has been upload on the predefined folder. This service is being architect in such a way that it can easily be

1. Adapted for different for different type of file to be parse
   1. File upload from UE
   2. UETRACE or CALL TRACE
2. Adaptability for adding multiple type of decoder module
   1. ASN Decoder
   2. Qualcomm Log Decoder
   3. UETRACE Decoder

Also, Windows Server application runs in concepts of Main Thread and Worker Thread Concept where main thread assigned the task to Work thread. The number of threads is configurable entities that would be depending upon underline target hardware.

Each work thread, parse the UE specific files and calculate the KPI from the decoded information and insert the information into the database plus generate files into .cvs format.

***Chapter***

*3*

Interface Definition

1. 1. Introduction

The interface between the UE and Window Server Application (known as KPIGenService Application) is file based .The KPIGenService application is an post processing engine that parse the files and decode the encoded buffers and update the file into the database.

* 1. File Format

After capturing data, UE store the logging messages in the file having format as ($IMEI\_$TESTID\_$NUM) where, $IMEI is the mobile equipment identifier, $TestID is the identification for particular test. $NUM is sequence number of file. For Eg. 123456781234567\_TEST\_1.

The message format in the file is given below:

* 1. Message Information

|  |  |  |  |
| --- | --- | --- | --- |
| **Network** | **Message** | **ID** | **Description** |
| **GSM** | GRR\_CELL\_RESELECTION\_MEASURMENT | 1000 | This provides computed C1, C2, C31, and C32 parameters for serving and neighboring cells prior to cell reselection. |
| GSM\_RR\_STATE\_MESSAGE | 1001 | This provides the current state of the RR layer, including sub-state information. |
| GSM\_RR\_CELLINFORM | 1002 | This provides system information of the serving cell. |
| GSM\_RR\_CHANNEL\_CONFIGURE | 1003 | This provides information about the channel configuration. |
| GSM\_RR\_SACCH\_REPORT | 1004 | This provides the results of measurements done by the mobile on serving and neighboring cells. |
| GSM\_RR\_CELL\_OPTIONS | 1005 | This provides information about cell options. |
| GSM\_TIMING\_ADVANCE | 1006 | This provides timing advance information on network-assigned and continuous timing advance values. |
| GRR\_PACKET\_SYSTEM\_INFORMATION | 1007 | This packet is generated each time Packet System Information 2 instances are received. |
| **WCDMA** | WCDMA\_CELL\_RESELECTION\_PACKET | 2000 | This provides information about serving and neighboring cells. |
| WCDMA\_PHYSICAL\_CHANNELS\_UPLINKS | 2001 | This log record holds the description of current active transport channel parameters generated upon new UL physical channel set up. |
| WCDMA\_ACTIVE\_SET | 2002 | This holds the parameters of the active cell. |
| WCDMA\_NEIGHBOR\_SET | 2003 | This record is generated when the RRC receives a neighbor set update message from the UTRAN. |
| WCDMA\_CELL\_ID | 2004 | This holds the log record that displays the various system information relevant to UE operation. |
| WCDMA\_RRC\_SIGNALING\_MESSAGE | 2005 | Log all the signaling messages. |
| WCDMA\_RRC\_STATES | 2006 | This holds the displays on the RRC and the current state of the RRC. |
| HS\_DSCH\_HARQ\_STA\_LOG\_PKT | 2007 | This packet will help measure the performance of HS-PDSCH. |
| MAC\_HS\_CONFIG\_LOG\_PKT | 2008 | The log packet provides detailed MAC configuration information. |
| MAC\_HS\_HEADER\_LOG\_PKT | 2009 | Packet can be generated at every 500 ms interval in the worst-case. |
| MAC\_HS\_STATUS\_LOG\_PKT | 2010 | This packet is generated by MAC-hs across all existing MAC-hs queues. |
| MAC\_HS\_RESET\_LOG\_PKT | 2011 | This packet provides number of logical channels mapped to HS transport channel. |
| HS\_DECODE\_STATUS\_LOG\_PKT\_VER\_2 | 2012 | This packet is intended to provide the status of one HS sub frame. |
| UMTS\_NAS\_SIGNALING | 2013 | Determine the direction of UMTS (OTA) messages. |

* 1. Message Detail Information

For getting detail information of all messages please refer below attached “decode.h” file.



***Chapter***

*4*

Design Document

1. 1. Functional View



**Window Service Module**

This module implements the Microsoft Window Service and provides the functionality to register with Window Operating system as service. This provides functionality to start and stop the service. During the startup, this module creates the Worker thread and corresponding synchronization object for the same.

**Master Module**

The master module periodically watches the incoming folder, if any files are available for parse. It validate if all files for specific test case are uploaded using the file naming convention. It moves the file to the processing folder and invokes the Plug Engine and select the worker thread which is available for the processing. This module runs in context of Window Service.

**Plugin Engine**

The Plugin Engine module is core of the system which runs under the worker thread. It provides API for registration for the FileParse Module as well as Decoder Module. This module also maintains the test context and provides context information to Decoder Module. Its an basic framework which invoke the individual FileParse Module function based on the file extension type and provide its output function. The FileParse output is divided into two categories

1. Processed Data: - This is information which already being processed and does not require any further decoding. This Plug Engine shall update this information into the test context.
2. Unprocessed Data: - This information is still the encoded data and required further decoding. The unprocessed data is being passed to respective decoder based on decoded information. The context information are being passed to respective decoder.

**File Parser Module**

This module open the file and apply the parsing the parameters and update the processed data into the context and unprocessed data back to Plugin Engine.

**Decoder Module**

This module implements functionality for Qualcomm API decoder, ASN Decode and CSN Decode. This decoder is implements in thread safe mode (no global data) whereas the multiple instance of decoder can be invoked. The decoder module takes the memory buffer and performs decode and update the test context. On completion of decode, this module invoke the functionality for File Writter Module and DB Wrapper Module.

**File Writer Module**

This module read the UE context and open the file in append mode and write the data into the comma separate format per test case.

**DB Wrapper Module**

This module manages connection with Database and Insert the row into the database.

* 1. Execution View



* 1. Implementation Details

This shall be updated latter