UNIVERSAL DATA INTEGRATION LAYER (UDIL) - MULTIVENDOR COMPLIANT ADVANCED METERING INFRASTRUCTURE (AMI) IN PAKISTAN

SUSTAINABLE ENERGY FOR PAKISTAN (SEP) PROJECT

UDIL Rev. No: PITC/UDIL/3.2.0-22029012

Submission Date: September 21, 2022

Submitted by: Power Information Technology Company (PITC),
406, WAPDA House, Lahore
Ministry of Energy (MoE), Power Division, Govt. of Pakistan
Tel. +92(42)99202666 | Fax +92(42)99202047-48

www.pitc.com.pk

Note: This document stipulates the data structure to be implemented at the head-ends level for smooth communication between MDC and MDM Systems through Universal Data Integration Layer.

SUMMARY

Title:	Universal Data Integration Layer (UDIL)

	Name, Title	Signature, Date
Document Prepared By:	Dr. Muhammad Kashif Shahzad, CTO, PITC	20-DEC-2018
	Mr. Waqas Safdar, Technology Advisor, SEP	20-DEC-2018

	Name, Title	Signature, Date
Document Approved By:	Mr. Imtiaz Ahmad, CEO, PITC	UDIL No: PITC/UDIL/1.1.0-81021224 24-DEC-2018

Revision History	Review Date	Revision Date	Signature, Date
1	20-05-2019	24-05-2019	UDIL Rev. No: PITC/UDIL/1.1.1-91025042 24-MAY-2019
2	01-07-2019	11-07-2019	UDIL Rev. No: PITC/UDIL/1.1.2-91020711 11-JULY-2019
3	04-09-2019	18-09-2019	UDIL Rev. No: PITC/UDIL/1.1.3-91020981 18-SEPT-2019
4	22-01-2020	22-01-2020	UDIL Rev. No: PITC/UDIL/1.1.5-02020122 22-JAN-2020
5	13-11-2020	13-11-2020	UDIL Rev. No: PITC/UDIL/2.1.5-02021131 13-NOV-2020
6	01-06-2021	30-07-2021	UDIL Rev. No: PITC/UDIL/3.0.0-12027003 30-JUL-2021
7	01-11-2021	19-11-2021	UDIL Rev. No: PITC/UDIL/3.1.0-12021119 19-NOV-2021

8	10-08-2022	21-09-2022	UDIL Rev. No: PITC/UDIL/3.2.0-22029012 21-SEPT-2022

Template Distributed To:	Location:	Copy No:	Location:	Copy Number
	Master file	1		
	PITC Website	2		

Copy Number: 01

Revision History:	Date	Description	Author	Comments
1	21-09-2022	Following instruction is added in Section 3, page-15. "In other words, Data needs to be populated directly in UDIL tables from MDC upon meter connection. No Triggers, Events, Recurring Services or Cron Jobs etc. are allowed to populate data in UDIL tables after meter disconnection."	Dr. Kashif Shahzad, CTO Mr. Abu Sufyan, Dy. Manager (AMI)	This will ensure all data being updated in MDC exposed tables without delay.
2	21-09-2022	The following instruction is added in the section 3.1.7 "Device Communication History" on page-33 with reference to debug log message: "message_log" will display proper log against each type data type. Following fields are added in the respective table: Field: message_log Data Type: Text Description: Message Log showing communication insights using UDIL Codes, including timestamps. Sample Message Logs are shown below Field: bytes_sent Data Type: Int Message: Size of Total Data sent to Meter (in Bytes) Data upload from MDC to Meter Field: bytes_received Data Type: Int Message: Size of Total Data received from Meter (in Bytes) Data download from Meter to MDC Field: bytes_remaining Data Type: Int Message: Remaining bytes of data package in SIM for meter communication (Optional) Field: sim_id Data Type: Varchar Message: Inserted SIM ID in Meter	Dr. Kashif Shahzad, CTO Mr. Abu Sufyan, Dy. Manager (AMI)	This will help in providing clear picture of meter communication with SIM id, duration and data package consumed.

		Modem. Field: connection_duration Data Type: Int Message: Total Duration of Meter Connection with MDC in Seconds		
3	21-09-2022	Following instruction is added in the section 3.3.6 "Device Creation" Meter programmed in Mode-I/non-Keep-alive mode must support both Wakeup mechanisms as defined in DDS-110. Wakeup Password must be alpha-numeric. • Wakeup by SMS • Wakeup by Call	Dr. Kashif Shahzad, CTO Mr. Abu Sufyan, Dy. Manager (AMI)	This will be tested that wakeup call can be send through SMS or call as needed.
4.	21-09-2022	Following service is added in section 3.4.2 "On_Demand_Parameter_Read" • Type 'MDI' – Programmed MDI Reset Date & Time	Dr. Kashif Shahzad, CTO Mr. Abu Sufyan, Dy. Manager (AMI)	This will provide programmed MDI reset date and time.
5.	21-09-2022	Following field description is added in section 3.4.3 "Transaction Status" Field: indv_status Data Type: Int Description: '0' for In-Progress. '1' for complete. '2' for Failed.	Dr. Kashif Shahzad, CTO Mr. Abu Sufyan, Dy. Manager (AMI)	This will improve transaction status monitoring.
6.	21-09-2022	Following field description is added in section 3.1 "READ REQUESTS – TABULAR" Field: id Data Type: Bigint Description: Auto-increment (Not NULL)	Dr. Kashif Shahzad, CTO Mr. Abu Sufyan, Dy. Manager (AMI)	This will handle wrong db_datetime and time stamping issues.

CONTENTS

ABBREVIATIONS & ACRONYMS

AEB	Area Electricity Board (former name for a DISCO)
AEDB	Alternative Energy Development Board
AMI	Advanced Metering Infrastructure
AMR	Automated Meter Reading
AT&C	Aggregate Technical and Commercial
B2B	Business-to-Business
BI	Business Intelligence
BW	Business Warehouse
CE (Ops)	Chief Engineer Operations
CE (P&E)	Chief Engineer Planning and Engineering
CIS	Customer Information System
CIM	Common Interface Model
COP	Chief of Party
CP	Commercial Procedures
CSD	Customer Service Director
CT	Current Transformer
DBMS	Data Base Management System
DISCO	State-owned Electricity Distribution Company
D&S	Design & Standards
ETL	Extract, Transform and Load
FESCO	Faisalabad Electric Supply Company Limited
FY	Financial year
GEPCO	Gujranwala Electric Power Company Limited
GIS	Geographic Information System
GoP	Government of Pakistan
HE	Head End System
HESCO	Hyderabad Electric Supply Company Limited
HT	High Tension, or High Voltage Level
IESCO	Islamabad Electric Supply Company Limited
KWh	Kilo Watt Hour
KPIs	Key Performance Indicators
LDI	USAID's Load Data Improvement Project
LESCO	Lahore Electric Supply Company Limited
LT	Low Tension, or Low Voltage Level
MDC	Meter Data Collector
MDM	Meter Data Management
MEPCO	Multan Electric Power Company Limited
M&E	Monitoring and Evaluation
MoE	Ministry of Energy
MW	Megawatt
NEPRA	National Electric Power Regulatory Authority

NPCC	National Power Control Center
NTDC	National Transmission and Despatch Company
PEPCO	Pakistan Electric Power Company Limited
PESCO	Peshawar Electric Supply Company Limited
PDC	Power Distribution Control Center
PITC	Pakistan Information Technology Company
QESCO	Quetta Electric Supply Company Limited
RO	Revenue Officer
SDO	Sub Divisional Officer
SE	Superintending Engineer
SEP	USAID's Sustainable Energy for Pakistan Project
SEPCO	Sukkur Electric Power Company Limited
S&S	Standards & Specification Department under PEPCO (formerly D&S department)
SQL	Structured Query Language
TESCO	Trible Area Electric Supply Company Limited
TC	SEP Technical Component
UDIL	Universal Data Integration Layer
USAID	United States Agency for International Development
WAPDA	Water and Power Development Authority
XEN	Executive Engineer

1. EXECUTIVE SUMMARY

1.1 PROJECT OVERVIEW

1.1.1 BACKGROUND

As part of the USAID Power Distribution Program (PDP), smart meters were installed on feeders at the grid stations across all the DISCOs for recording load profiles and real-time monitoring of power dispatch. The project was further extended to provide support to three DISCOs (MEPCO, PESCO and HESCO) with smart meters installation at the customer level. Customers were selected based on certain parameters, such as high load profiles and agricultural context (tube-wells). The project had mainly three components; provision of smart meters, installation of headend applications, and development of a web-based portal for monitoring of the installed smart meters. Initially, the solution was designed to support integration of all types and makes of smart meters, but unfortunately due to lack of understanding and technical capacity, the Advanced Metering Infrastructure (AMI) project remained confined to a single vendor specific solution and ended up in a "vendor lock" situation. In turn, it prevented DISCOs from further expanding AMI solution both at the grid and consumer levels. The installed system does not provide support to the smart meters from multiple vendors and therefore there is an immediate need to find solution to provide a level playing field to all meter manufacturers enabling them to integrate their devices with the existing system.

As a part of the solution. Sustainable Energy for Pakistan Project (SEP) intended to standardize Universal Data Integration Layer (UDIL) structure and introduce a Meter Data Management (MDM) solution along with Business Intelligence (BI) tool to expose multiple vendor Head Ends (HEs) in common AMI landscape and provide structured information through analytics over dashboards enabling DISCOs to access near real-time information for prudent decision making. Such a vendor independent plug-and-play arrangement will help DISCOs to procure smart meters through competitive bidding leading to the most cost-effective AMI deployment.

1.1.2 SUSTAINABLE ENERGY FOR PAKISTAN PROJECT (SEP)

With significant increase in generation during the previous government, the gap between supply and demand has narrowed down significantly and sufficient installed base of generation is now available to meet the demand. Despite projected surplus generation, many risks and constraints to the continued development and sustainable supply of electricity remains. The most significant of all these is accumulation of circular debt at a rapid pace in view of increased supply of electricity in the distribution system with excessively high technical as well as commercial losses. Unless the DISCOs are disciplined to manage their businesses and improve financial performance, such effort is not self-sustaining. Cash starved DISCOs would not be able to invest in infrastructure which is necessary to support the transmission and distribution of increased supply of power, which in turn is necessary to maintain reliability of the grid.

High loss DISCOs in general, consume significant chunk of the country's total generation which puts excessive financial burden on the entire distribution sector and is an impediment to the economic growth of the country. DISCOs' business processes are heavily reliant on manual processing, supplemented by information technology components allowing for errors and potential manipulation of results. The limited capacity of its staff to embrace the latest technology and the inability to meet with the growing challenges, greatly compromise overall DISCOs' performance. In this case, outdated policies and procedures, poor governance and inefficient work practices are all detrimental to the GOP efforts to enhance value of all public-sector DISCOs for future privatization.

1.1.3 SEP SCOPE OF WORK

The objective of USAID's Sustainable Energy for Pakistan (SEP) Project is to help Pakistan attract private sector engagement in developing sustainable, clean energy future for the country through its various activities across all segments of the clean energy domain; generation, transmission, distribution, and governance.

The SEP Project activities are categorized into following four (4) Technical Components:

Technical Component 1 (TC 1) – Create a credit-worthy business environment that attracts private sector investors in a competitive, transparent energy market accessible to all stakeholders

Technical Component 2 (TC 2) – Transform the transmission system operator – the National Transmission and Dispatch Company (NTDC) into an entity capable of managing and expanding the national grid while ensuring reliable, efficient, and stable transmission and dispatch services, and support the market operator and regulator (Central Power Purchasing Agency Limited (CPPA-G) and National Electric Power Regulatory Authority (NEPRA)) in transitioning to an open marketplace.

Technical Component 3 (TC 3) – Employ a systematic methodology for utility turnaround, developed and implemented for electricity distribution programs, delivering dramatic reduction in Aggregate Technical and Commercial (AT&C) losses and effectively enhancing the value of State-owned electricity distribution company (DISCOs) for privatization.

Technical Component 4 (TC 4) – Overcome specific barriers to clean energy (CE) investments through suitable policy, regulatory, or legislative amendments and transaction support services.

Based on SEP approved work plan for TC3, the Ministry of Energy (MOE) proposed to USAID to undertake list of tasks under the SEP project mainly aimed at reduction of AT&C losses and improving operational efficiency especially for the poor performing DISCOs.

Given the limited scope and resources for the commodity support under the USAID SEP project, it was envisaged to remain focused on selected priority tasks for providing technical assistance leading to improved financial and operational efficiency of selected DISCOs. SEP TC3 team together with USAID reviewed the list of tasks provided by the MOE to select the ones which are aligned with the scope and TOR of the USAID's SEP project. Successive consultation meetings were held with Pakistan Electric Power Company (PEPCO) and PESCO to develop a suitable technical assistance strategy.

Keeping in view the earlier selection of PESCO and MEPCO as turnaround DISCOs under the previous USAID Power Distribution Program (PDP), where considerable technical and commodity assistance was provided, detailed discussions were held with the senior management of both the DISCOs leading to the selection of the following short-listed tasks for immediate technical assistance:

- Preparation of plans for further upgrades and enhanced utilization of the Load Data Improvement Project (LDIP).
- Gap analysis and selected upgrading of installed AMR/AMI projects.
- Installation of Aerial Bundled Conductor (ABC) in congested areas.
- Specific gap analysis of ERP implementation.

1.2 AMI GAP ANALYSIS

Given an inefficient power distribution network infrastructure, lack of efficient system planning, high technical as well as commercial losses, financial mismanagement, and gaps in standards and specifications, DISCOs face an uphill task of meeting continuous regulator-mandated technical, operational and financial performance goals. In the absence of adequate technology updating, DISCOs are facing major issues in effectively operating and maintaining their respective distribution networks. Mainly the commercial operations of meter-to-cash are flawed by using outdated procedures and processes where they mainly rely on legacy metering technologies which result in the loss of cash collection owing to deficient billing systems.

One of the most significant interventions that was ever implemented in Pakistan was the commissioning of Advanced Metering Infrastructure (AMI). USAID under its Power Distribution Program (PDP) implemented the AMI System for PESCO, MEPCO and HESCO employing smart AMR meters, hardware / software for Meter Data Collection (MDC), end-user interface and commissioning of an integrated Advanced Metering Infrastructure (AMI). The objective of this initiative was to introduce a fully automated metering system which would help the distribution sector in reducing losses, enhance load management controls, provide access to real-time customer load profiles, improve billing quality and revenue resulting in reduced billing complaints, increase in operational efficiency, reduced operating costs and optimize the DISCOS metering and billing operations. The AMI head-end systems were installed at PITC Data Centre / NOC at WAPDA House Lahore and DISCOs field users were provided with remote access through secure private physical and logical VPN links. They were provided with access to the meter data repository through web-based applications to use the meter readings for billing and analysis's purpose whereas sub-divisions, divisions and revenue offices were connected over the IP cloud to perform day-to-day operational and other commercial activities. DISCOs acknowledged that the AMI rollout by USAID's PDP has resulted in tremendous savings on account of energy sale and improvement in the revenue collection in addition to other benefits such as technical energy loss reduction and improved system reliability. It also helped them with improved customer service, considerable reduction in billing complaints, increased operational efficiency, demand side load management, reduced operating costs and fully automated electricity metering and billing operation.

The biggest disadvantage with the existing AMI system is that it does not support integration of smart meters from multiple vendors and has created monopoly in favor of a single vendor. Given the benefits accrued over the years, DISCOs desired to expand AMR meter installations but could not accomplish it owing to difficulties in solicitations due to the vendor lock situation. Another major obstacle in expanding AMI activities is the absence of standards and specifications to promote standardized multivendor compliant AMI platform. National Transmission and Despatch Company (NTDC)'s design department who is custodian of the technical specifications, PITC which is responsible for extending IT services to DISCOs, AMR meter manufacturers and AMI integrators all should have thought through a workable solution to enable DISCOs expand their smart meter deployment but unfortunately, no concrete efforts were made to address the problem.

2. OPEN ARCHITECTURE-BASED AMI LANDSCAPE

Automating the metering process is a proven most efficient and cost-effective way to streamline distribution operations which combines interval data measurements with continuously available remote communication. It ensures timely billing, increased billing accuracy, flexible billing cycles, easier energy theft detection and creates customer energy profiles for targeting energy efficiency/demand response programs. With the rising demand, operational and environmental constraints, and aging infrastructures, nowadays utilities around the world are looking for more than just bill creation using the data. They use this information to improvise better and efficient customer services, improved utility asset management, timely outage management as well as effective system planning.

To mitigate the difficulty in integrating smart meters of various manufacturers on the existing landscape, SEP in collaboration with DISCOs, PITC, NTDC's S&S and Smart Meter Manufacturers initiated the design and development of an open architecture solution for establishing an integrated AMI platform offering equal opportunity to all vendors including international manufacturers. This document contains a detailed design of an MDM-based Multivendor Compliant AMI platform employing integrated middleware layer and standardized Universal Data Integration Layer (UDIL) to aggregate the data received from various brands of smart meters through individual MDCs/HEs.

2.1 UNIVERSAL DATA INTEGRATION LAYER (UDIL)

Presently the MDC systems are provided by a specific meter manufacturers and support recording of brand specific meter profiles only. To perform billing operations, DISCOs are required to switch to AMR manufacturer specific interface and fetch the billing determinants, transform the metering data to acceptable format and compile it before submitting it to the billing engine which in turn increases the chances of data corruption and loss. In other words, MDCs of each of the vendors work in silos and the data does not converge on a single platform which from the operations view point is highly inefficient. This warrants a standard Universal Data Integration Layer (UDIL) content and data structure that each of the vendor can expose at the MDC/HEs for data aggregation.

The primary purpose of developing a standardized UDIL is to facilitate DISCOs to integrate data from multi-vendor AMR meters with a common data warehouse in a standard format to:

- Enable integration of the multiple brand MDCs in the smart metering enterprise landscape.
- Consolidate metering data received from multiple MDCs using a common platform.
- Provide one portal to end-users for administering the smart meters irrespective of brands and makes.
- Integrate the MDM system with the billing system through universal data integration layer, for providing billing determinants and to retrieve payment information.
- Facilitate data analytics over a uniform e-system.
- Enable end-users to slice/dice data and generate charts/reports as per their requirements.
- End current single vendor-lock situation and open up the AMI landscape for all AMR meter manufacturers / suppliers.
- Enable DISCOs to deploy smart meters in a cost-effective manner with increased market access.

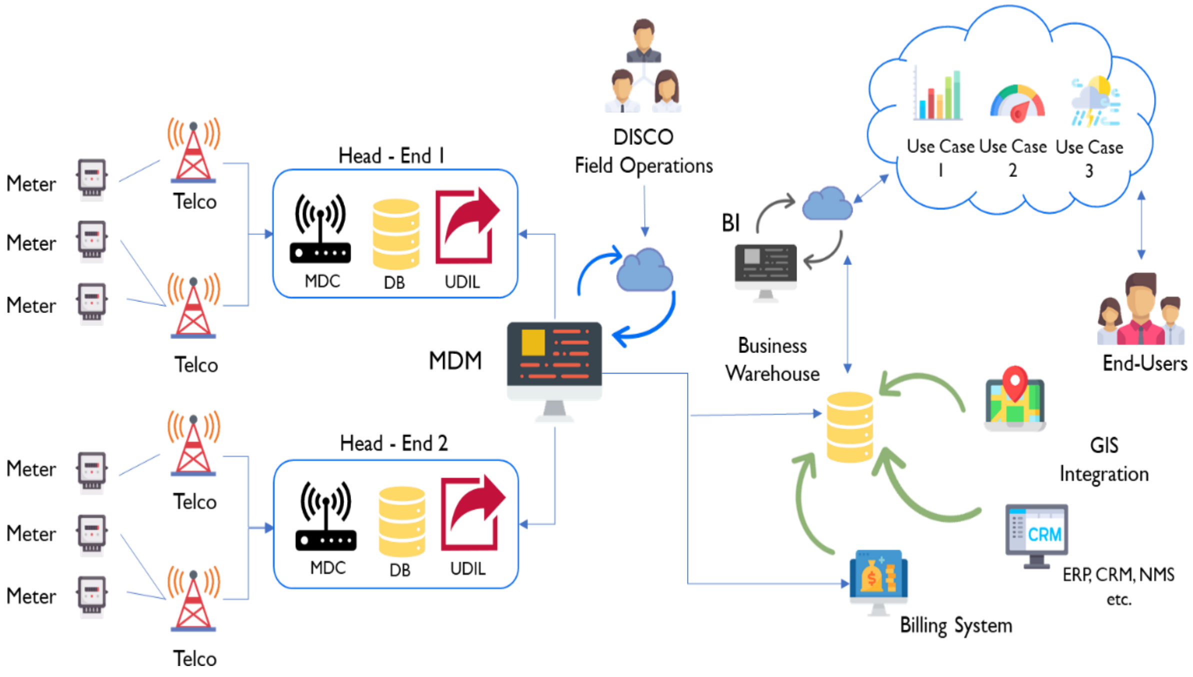
2.1.1 SYSTEM DESIGN ARCHITECTURE

To achieve multi-vendor compliance, a standardized universal data integration layer is required to aggregate the data streams from various MDCs on a common platform. The MDCs will collect

metering data directly from their respective AMR meters and push this data into the MDM systems by utilizing the Universal Data Integration layer (UDIL). Consequently, any MDM can easily bind all of the available MDCs metering data through UDIL and convert it into actionable information. From MDM, the billing determinants will flow to the billing system for consumer invoicing whereas the load data will aid the DISCOs to carry out prescriptive analysis and subsequent actions.

The UDIL will have the capacity to perform various rules on data packets received from multiple MDCs and facilitate its integration with the common data warehouse and billing systems. The central data warehouse will serve as the reporting server to MDM without any burden in collecting packets from multiple MDCs. The typical system architecture will be as follows:

AMI Enterprise Landscape



2.1.2 GENERAL SCOPE

SEP in consultation with all the stakeholders (PEPCO, NEPRA, PITC, DISCOs, AMR meter manufacturers, AMI vendors and NTDC's Standards & Specification Department) has designed and developed standardized specifications for Universal Data Integration Layer (UDIL) database structure, tables, commands and controls. An MDM system will be installed in the enterprise landscape to consolidate meter data, process commands and services enabling the reading of all headend exposed tables as per standard UDIL specifications thus creating an integrated AMI landscape employing multi-vendor headend systems.

The salient system functionalities will be to:

- Communicate with the multiple MDCs on defined intervals/schedule for exchange of data packet including meter profiles, grid profiles, alerts, commands and to store the acquired data in the intermediate database.
- Provide one single interface to end-users for sending commands to meters via MDC and recording of the activity and response in the intermediate database.
- Provide dashboard to various stakeholders for access to instant information on key parameters and generate technical reports.

- Facilitate integration with the billing system, flat file-based or automatically feeding billing determinants for customer invoicing.
- Fetch transactional data specific to smart meter from billing system and store it in the intermediate database for energy accounting and vigilance.
- Introduce web-based customizable dashboard for users to convert data into structured information and develop appropriate analytics.

2.1.3 BUSINESS PROCESS SUPPORT

The MDM-based system will support the business and operational processes in line with the prevalent WAPDA Commercial Procedures (CP) such as:

- CP2: New Connection
- CP3: Temporary Disconnection
- CP4: Permanent Disconnection
- CP5: Reconnection
- CP6: Meter Reading
- CP7: Meter Change
- CP8: New Connection (Revenue Office)
- CP9: Meter Change (Billing Control Section)
- CP10: Billing
- CP13: Temporary Disconnection (Revenue Office)
- CP14: Permanent Disconnection (Revenue Office)
- CP15: Reconnection (Revenue Office)

2.1.4 FUNCTIONAL REQUIREMENTS

The functional requirements of the system are primarily aimed at providing an interface to DISCOs for managing the meter installation and operations through a universal data integration layer. This will connect with MDM system to perform operations through relevant meters MDCs. Therefore, the user will not be concerned on how the back-end communication takes place for multiple brand of meters and will be exposed to a single interface to execute routine functions.

The MDM system will generate billing determinants for export to the billing systems for invoicing functions. Transactional data from the billing systems will be mirrored into the MDM system so that users could compare technical information with the commercial codes.

Smart meter manufacturers / vendors will be required to expose key information from their respective MDCs on standard data structures as defined in this document by fetching data packets in a unified manner. The MDM will employ the functionality of reading multiple MDCs regardless of communication mediums such as web services, data tables, flat file etc.

3. UDIL DATA STRUCTURE

To establish communication with MDM systems, MDCs of each of the meter manufacturer have to follow the standard data structures. This includes field name, field type and field description in transact SQL. The structures are categorized in this section below. Meter vendors have to provide scheduled reads via database tables and commands via API/Web services. In case of web services, the field names will be classified as service parameters. As a thumb rule, length of varchar/string is 50 chars and decimal refer to 3 digits precision as per relevant design standard. No decimal value should exceed 3 digits precision. Composite Key 🔑 are represented with light blue background color and symbol.

It is pertinent to mention here that no restriction has been imposed on the smart meter manufacturers to follow exclusive data transfer options described in this document. Further, all the write requests from MDM systems, through universal data integration layer, needs to be generated in asynchronous mode i.e. jobs will be executed on meter whenever, it makes connection with MDC. The db_datetime field is used for database indexing purpose (preferred binary indexing). "db_datetime" field must contain actual time in which data is inserted. Any partial or in-complete data any of provided tables will be marked as wrong or discrepancy and it may lead to meter inspection.

Data Update is not allowed in any tables described in "section 3.1" except Meter Visuals table.

It is worth mentioning that all web services should be on REST Protocol.

Only Database tables are allowed neither views nor tables derived from Database Views are allowed.

Similarly, upon connection of meter with MDC all type of required data must be available in underlying UDIL tables before meter disconnects from MDC. In other words, Data needs to be populated directly in UDIL tables from MDC upon meter connection. No Triggers, Events, Recurring Services or Cron Jobs etc. are allowed to populate data in UDIL tables after meter disconnection.

MDC must be capable to handle all connections/meters/devices concurrently against procured quantity of a particular tender from DISCO for which UDIL compliance testing request is submitted.

MDC must synchronize meter's date time on daily basis.

3.1 READ REQUESTS – TABULAR

The tabular requests are based on fetching information directly from database tables exposed by MDC for the MDM system via universal data integration layer.

MDM will update "is_synced" column after successful read operation. MDC can purge data from these tables where "is_synced" flag is 1 (if required). MDC will always insert "is_synced" with 0 value.

3.1.1 INSTANTANEOUS DATA

These are electrical parameters recorded by MDC. The request is made by end-user on ad hoc basis and includes basic grid and meter profiles.

This table will contain two types of data

- All On-Demand Instantaneous read of all type of devices having any communication type
- Instantaneous data packet of that instant for every non-keepalive (Mode-1) device/meter upon every connection.
- Saved data if programmed using Meter Data Sampling. By default, meters will be programmed for storing instantaneous data twice a day.

The required list is as follow:

TABLE ID: *INSTANTANEOUS_DATA*

Table Field	Data Type	Description
current_tariff_register	Int	Current/Active Tariff Register
signal_strength	Decimal	Signal Strength in dBm
msn	Bigint	Meter Serial Number
frequency	Decimal	Frequency in Hertz
🔑 global_device_id	Varchar	Unique ID for each device by MDM
🔑 meter_datetime	Datetime	Meter Date & Time
current_phase_a	Decimal	Current Phase A
current_phase_b	Decimal	Current Phase B
current_phase_c	Decimal	Current Phase C
voltage_phase_a	Decimal	Voltage Phase A
voltage_phase_b	Decimal	Voltage Phase B
voltage_phase_c	Decimal	Voltage Phase C
aggregate_active_pwr_pos	Decimal	Aggregate Active Power Import
aggregate_active_pwr_neg	Decimal	Aggregate Active Power Export
aggregate_active_pwr_abs	Decimal	Aggregate Active Power Absolute
aggregate_reactive_pwr_pos	Decimal	Aggregate Reactive Power Import
aggregate_reactive_pwr_neg	Decimal	Aggregate Reactive Power Export
aggregate_reactive_pwr_abs	Decimal	Aggregate Reactive Power Absolute as per Pakistan standard
average_pf	Decimal	Average Power Factor
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database
is_synced	Int	0 By default 1 if data is safe to delete from MDC
🔑 id	Bigint	Auto-increment (Not NULL) Unique-ID

3.1.2 BILLING DATA

These registers are based on tariff types programmed in meter for recording of billing determinants at defined intervals. The packet of information includes energy, maximum demand for each of the register defined as per the tariff requirements.

Billing data will be saved inside device/meter at 12:00 A.M daily and communicated upon every communication i.e. if communication establishes after data is saved inside meter.

Following columns are mandatory

*_*_neg_* fields are required for bi-directional meters.

* However, if meter is recording *_*_neg_* fields in case of uni-directional then actual *_*_neg_* fields are required

*_*_pos_*, *_*_abs_* fields are required for all types of meters

TABLE ID: *BILLING_DATA*

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
active_energy_pos_t1	Decimal	T1 Active kWh (Import)
active_energy_pos_t2	Decimal	T2 Active kWh (Import)
active_energy_pos_t3	Decimal	T3 Active kWh (Import)
active_energy_pos_t4	Decimal	T4 Active kWh (Import)
active_energy_pos_tl	Decimal	Total Active kWh (Import)
active_energy_neg_t1	Decimal	T1 Active kWh (Export)
active_energy_neg_t2	Decimal	T2 Active kWh (Export)
active_energy_neg_t3	Decimal	T3 Active kWh (Export)
active_energy_neg_t4	Decimal	T4 Active kWh (Export)
active_energy_neg_tl	Decimal	Total Active kWh (Export)
active_energy_abs_t1	Decimal	T1 Absolute Active kWh
active_energy_abs_t2	Decimal	T2 Absolute Active kWh
active_energy_abs_t3	Decimal	T3 Absolute Active kWh
active_energy_abs_t4	Decimal	T4 Absolute Active kWh
active_energy_abs_tl	Decimal	Total Absolute Active kWh
reactive_energy_pos_t1	Decimal	T1 Reactive kVARh (Import)
reactive_energy_pos_t2	Decimal	T2 Reactive kVARh (Import)
reactive_energy_pos_t3	Decimal	T3 Reactive kVARh (Import)
reactive_energy_pos_t4	Decimal	T4 Reactive kVARh (Import)
reactive_energy_pos_tl	Decimal	Total Reactive kVARh (Import)
reactive_energy_neg_t1	Decimal	T1 Reactive kVARh (Export)
reactive_energy_neg_t2	Decimal	T2 Reactive kVARh (Export)
reactive_energy_neg_t3	Decimal	T3 Reactive kVARh (Export)
reactive_energy_neg_t4	Decimal	T4 Reactive kVARh (Export)
reactive_energy_neg_tl	Decimal	Total Reactive kVARh (Export)
reactive_energy_abs_t1	Decimal	T1 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t2	Decimal	T2 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t3	Decimal	T3 Absolute Reactive kVARh as per Pakistan standard

Table Field	Data Type	Description
reactive_energy_abs_t4	Decimal	T4 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_tl	Decimal	Total Absolute Reactive kVARh as per Pakistan standard
active_mdi_pos_t1	Decimal	T1 Active MDI kW (Import)
active_mdi_pos_t2	Decimal	T2 Active MDI kW (Import)
active_mdi_pos_t3	Decimal	T3 Active MDI kW (Import)
active_mdi_pos_t4	Decimal	T4 Active MDI kW (Import)
active_mdi_pos_tl	Decimal	Total Active MDI kW (Import)
active_mdi_neg_t1	Decimal	T1 Active MDI kW (Export)
active_mdi_neg_t2	Decimal	T2 Active MDI kW (Export)
active_mdi_neg_t3	Decimal	T3 Active MDI kW (Export)
active_mdi_neg_t4	Decimal	T4 Active MDI kW (Export)
active_mdi_neg_tl	Decimal	Total Active MDI kW (Export)
active_mdi_abs_t1	Decimal	T1 Absolute Active MDI kW
active_mdi_abs_t2	Decimal	T2 Absolute Active MDI kW
active_mdi_abs_t3	Decimal	T3 Absolute Active MDI kW
active_mdi_abs_t4	Decimal	T4 Absolute Active MDI kW
active_mdi_abs_tl	Decimal	Total Absolute Active MDI kW
cumulative_mdi_pos_t1	Decimal	T1 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t2	Decimal	T2 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t3	Decimal	T3 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t4	Decimal	T4 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_tl	Decimal	Total Cumulative Active MDI kW (Import)
cumulative_mdi_neg_t1	Decimal	T1 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t2	Decimal	T2 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t3	Decimal	T3 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t4	Decimal	T4 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_tl	Decimal	Total Cumulative Active MDI kW (Export)
cumulative_mdi_abs_t1	Decimal	T1 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t2	Decimal	T2 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t3	Decimal	T3 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t4	Decimal	T4 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_tl	Decimal	Total Absolute Cumulative Active MDI kW
mdc_read_datetime	Datetime	Reading Date & Time of MDC

Table Field	Data Type	Description
db_datetime	Datetime	Record Entry Date & Time in Database
is_synced	Int	0 By default 1 if data is safe to delete from MDC
🔑 id	Bigint	Auto-increment (Not NULL) Unique-ID

3.1.3 MONTHLY BILLING DATA

The information maintained in this table is a summarized view of cumulative information as maintained in previous table. An additional counter maintains as to how many times MDI register have been reset for audit purpose.

Monthly Billing data will be saved inside device/meter at programmed MDI Reset date and will be available in monthly billing table upon every communication i.e. if communication establishes after data is saved inside meter.

Following columns are mandatory

*_*_neg_* fields are required for bi-directional meters

* However, if meter is recording *_*_neg_* fields in case of uni-directional then actual *_*_neg_* fields are required

*_*_pos_*, *_*_abs_* fields are required for all types of meters

TABLE ID: *MONTHLY_BILLING_DATA*

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
active_energy_pos_t1	Decimal	T1 Active kWh (Import)
active_energy_pos_t2	Decimal	T2 Active kWh (Import)
active_energy_pos_t3	Decimal	T3 Active kWh (Import)
active_energy_pos_t4	Decimal	T4 Active kWh (Import)
active_energy_pos_tl	Decimal	Total Active kWh (Import)
active_energy_neg_t1	Decimal	T1 Active kWh (Export)
active_energy_neg_t2	Decimal	T2 Active kWh (Export)
active_energy_neg_t3	Decimal	T3 Active kWh (Export)
active_energy_neg_t4	Decimal	T4 Active kWh (Export)
active_energy_neg_tl	Decimal	Total Active kWh (Export)
active_energy_abs_t1	Decimal	T1 Absolute Active kWh
active_energy_abs_t2	Decimal	T2 Absolute Active kWh
active_energy_abs_t3	Decimal	T3 Absolute Active kWh
active_energy_abs_t4	Decimal	T4 Absolute Active kWh
active_energy_abs_tl	Decimal	Total Absolute Active kWh
reactive_energy_pos_t1	Decimal	T1 Reactive kVARh (Import)

Table Field	Data Type	Description
reactive_energy_pos_t2	Decimal	T2 Reactive kVARh (Import)
reactive_energy_pos_t3	Decimal	T3 Reactive kVARh (Import)
reactive_energy_pos_t4	Decimal	T4 Reactive kVARh (Import)
reactive_energy_pos_tl	Decimal	Total Reactive kVARh (Import)
reactive_energy_neg_t1	Decimal	T1 Reactive kVARh (Export)
reactive_energy_neg_t2	Decimal	T2 Reactive kVARh (Export)
reactive_energy_neg_t3	Decimal	T3 Reactive kVARh (Export)
reactive_energy_neg_t4	Decimal	T4 Reactive kVARh (Export)
reactive_energy_neg_tl	Decimal	Total Reactive kVARh (Export)
reactive_energy_abs_t1	Decimal	T1 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t2	Decimal	T2 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t3	Decimal	T3 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t4	Decimal	T4 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_tl	Decimal	Total Absolute Reactive kVARh as per Pakistan standard
active_mdi_pos_t1	Decimal	T1 Active MDI kW (Import)
active_mdi_pos_t2	Decimal	T2 Active MDI kW (Import)
active_mdi_pos_t3	Decimal	T3 Active MDI kW (Import)
active_mdi_pos_t4	Decimal	T4 Active MDI kW (Import)
active_mdi_pos_tl	Decimal	Total Active MDI kW (Import)
active_mdi_neg_t1	Decimal	T1 Active MDI kW (Export)
active_mdi_neg_t2	Decimal	T2 Active MDI kW (Export)
active_mdi_neg_t3	Decimal	T3 Active MDI kW (Export)
active_mdi_neg_t4	Decimal	T4 Active MDI kW (Export)
active_mdi_neg_tl	Decimal	Total Active MDI kW (Export)
active_mdi_abs_t1	Decimal	T1 Absolute Active MDI kW
active_mdi_abs_t2	Decimal	T2 Absolute Active MDI kW
active_mdi_abs_t3	Decimal	T3 Absolute Active MDI kW
active_mdi_abs_t4	Decimal	T4 Absolute Active MDI kW
active_mdi_abs_tl	Decimal	Total Absolute Active MDI kW
cumulative_mdi_pos_t1	Decimal	T1 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t2	Decimal	T2 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t3	Decimal	T3 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t4	Decimal	T4 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_tl	Decimal	Total Cumulative Active MDI kW (Import)
cumulative_mdi_neg_t1	Decimal	T1 Cumulative Active MDI kW (Export)

Table Field	Data Type	Description
cumulative_mdi_neg_t2	Decimal	T2 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t3	Decimal	T3 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t4	Decimal	T4 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_tl	Decimal	Total Cumulative Active MDI kW (Export)
cumulative_mdi_abs_t1	Decimal	T1 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t2	Decimal	T2 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t3	Decimal	T3 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t4	Decimal	T4 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_tl	Decimal	Total Absolute Cumulative Active MDI kW
Longitude	Decimal	Longitude of Device
Latitude	Decimal	Latitude of Device
picture_1	BLOB	Meter Screen Picture 1
picture_2	BLOB	Meter Screen Picture 2
reading_mode	Int	0 for Manual Reading, 1 for Automatic Reading
mdi_reset_datetime	Datetime	MDI Reset Date & Time
reset_count	Int	MDI Reset Count Number
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database
is_synced	Int	0 By default 1 if data is safe to delete from MDC
🔑 id	Bigint	Auto-increment (Not NULL) Unique-ID

3.1.4 LOAD PROFILE DATA

The load data records the complete meter and grid profiles at pre-defined intervals.

Following columns are mandatory

msn, global_device_id, meter_datetime, channel_id, interval, active_energy_pos_t1, active_energy_pos_t2, active_energy_pos_tl, reactive_energy_pos_t1, reactive_energy_pos_t2, reactive_energy_pos_tl, active_energy_neg_t1, active_energy_neg_t2, active_energy_neg_tl, reactive_energy_neg_t1, reactive_energy_neg_t2, reactive_energy_neg_tl, active_energy_abs_t1, active_energy_abs_t2, active_energy_abs_tl, reactive_energy_abs_t1, reactive_energy_abs_t2, reactive_energy_abs_tl, aggregate_active_pwr_pos, aggregate_reactive_pwr_pos, aggregate_active_pwr_neg, aggregate_reactive_pwr_neg, aggregate_active_pwr_abs, aggregate_reactive_pwr_abs, mdc_read_datetime, db_datetime

In above list of columns

*_*_neg_* fields are required for bi-directional meters

* However, if meter is recording *_*_neg_* fields in case of uni-directional mode then actual *_*_neg_* fields are required

*_*_pos_*, *_*_abs_* fields are required for all types of meters.

Load Profiles will be saved after every 30 minutes at 00 and 30 minute of every hour.

TABLE ID: ***LOAD_PROFILE_DATA***

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
frequency	Decimal	Frequency in Hertz
channel_id	Int	Load Profile Channel ID where applicable
interval	Int	Interval of Profile
active_energy_pos_t1	Decimal	T1 Active kWh (Import)
active_energy_pos_t2	Decimal	T2 Active kWh (Import)
active_energy_pos_t3	Decimal	T3 Active kWh (Import)
active_energy_pos_t4	Decimal	T4 Active kWh (Import)
active_energy_pos_tl	Decimal	Total Active kWh (Import)
active_energy_neg_t1	Decimal	T1 Active kWh (Export)
active_energy_neg_t2	Decimal	T2 Active kWh (Export)
active_energy_neg_t3	Decimal	T3 Active kWh (Export)
active_energy_neg_t4	Decimal	T4 Active kWh (Export)
active_energy_neg_tl	Decimal	Total Active kWh (Export)
active_energy_abs_t1	Decimal	T1 Absolute Active kWh
active_energy_abs_t2	Decimal	T2 Absolute Active kWh
active_energy_abs_t3	Decimal	T3 Absolute Active kWh
active_energy_abs_t4	Decimal	T4 Absolute Active kWh
active_energy_abs_tl	Decimal	Total Absolute Active kWh
reactive_energy_pos_t1	Decimal	T1 Reactive kVARh (Import)
reactive_energy_pos_t2	Decimal	T2 Reactive kVARh (Import)
reactive_energy_pos_t3	Decimal	T3 Reactive kVARh (Import)
reactive_energy_pos_t4	Decimal	T4 Reactive kVARh (Import)
reactive_energy_pos_tl	Decimal	Total Reactive kVARh (Import)
reactive_energy_neg_t1	Decimal	T1 Reactive kVARh (Export)
reactive_energy_neg_t2	Decimal	T2 Reactive kVARh (Export)
reactive_energy_neg_t3	Decimal	T3 Reactive kVARh (Export)
reactive_energy_neg_t4	Decimal	T4 Reactive kVARh (Export)
reactive_energy_neg_tl	Decimal	Total Reactive kVARh (Export)
reactive_energy_abs_t1	Decimal	T1 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t2	Decimal	T2 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t3	Decimal	T3 Absolute Reactive kVARh as per Pakistan standard
reactive_energy_abs_t4	Decimal	T4 Absolute Reactive kVARh as per Pakistan standard

Table Field	Data Type	Description
reactive_energy_abs_tl	Decimal	Total Absolute Reactive kVARh as per Pakistan standard
active_mdi_pos_t1	Decimal	T1 Active MDI kW (Import)
active_mdi_pos_t2	Decimal	T2 Active MDI kW (Import)
active_mdi_pos_t3	Decimal	T3 Active MDI kW (Import)
active_mdi_pos_t4	Decimal	T4 Active MDI kW (Import)
active_mdi_pos_tl	Decimal	Total Active MDI kW (Import)
active_mdi_neg_t1	Decimal	T1 Active MDI kW (Export)
active_mdi_neg_t2	Decimal	T2 Active MDI kW (Export)
active_mdi_neg_t3	Decimal	T3 Active MDI kW (Export)
active_mdi_neg_t4	Decimal	T4 Active MDI kW (Export)
active_mdi_neg_tl	Decimal	Total Active MDI kW (Export)
active_mdi_abs_t1	Decimal	T1 Absolute Active MDI kW
active_mdi_abs_t2	Decimal	T2 Absolute Active MDI kW
active_mdi_abs_t3	Decimal	T3 Absolute Active MDI kW
active_mdi_abs_t4	Decimal	T4 Absolute Active MDI kW
active_mdi_abs_tl	Decimal	Total Absolute Active MDI kW
cumulative_mdi_pos_t1	Decimal	T1 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t2	Decimal	T2 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t3	Decimal	T3 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_t4	Decimal	T4 Cumulative Active MDI kW (Import)
cumulative_mdi_pos_tl	Decimal	Total Cumulative Active MDI kW (Import)
cumulative_mdi_neg_t1	Decimal	T1 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t2	Decimal	T2 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t3	Decimal	T3 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_t4	Decimal	T4 Cumulative Active MDI kW (Export)
cumulative_mdi_neg_tl	Decimal	Total Cumulative Active MDI kW (Export)
cumulative_mdi_abs_t1	Decimal	T1 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t2	Decimal	T2 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t3	Decimal	T3 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_t4	Decimal	T4 Absolute Cumulative Active MDI kW
cumulative_mdi_abs_tl	Decimal	Total Absolute Cumulative Active MDI kW
current_phase_a	Decimal	Current Phase A
current_phase_b	Decimal	Current Phase B
current_phase_c	Decimal	Current Phase C
voltage_phase_a	Decimal	Voltage Phase A
voltage_phase_b	Decimal	Voltage Phase B
voltage_phase_c	Decimal	Voltage Phase C
active_pwr_pos_phase_a	Decimal	Active Power Import Phase A

Table Field	Data Type	Description
active_pwr_pos_phase_b	Decimal	Active Power Import Phase B
active_pwr_pos_phase_c	Decimal	Active Power Import Phase C
aggregate_active_pwr_pos	Decimal	Aggregate Active Power Import
active_pwr_neg_phase_a	Decimal	Active Power Export Phase A
active_pwr_neg_phase_b	Decimal	Active Power Export Phase B
active_pwr_neg_phase_c	Decimal	Active Power Export Phase C
aggregate_active_pwr_neg	Decimal	Aggregate Active Power Export
aggregate_active_pwr_abs	Decimal	Aggregate Active Power Absolute
reactive_pwr_pos_phase_a	Decimal	Reactive Power Import Phase A
reactive_pwr_pos_phase_b	Decimal	Reactive Power Import Phase B
reactive_pwr_pos_phase_c	Decimal	Reactive Power Import Phase C
aggregate_reactive_pwr_pos	Decimal	Aggregate Reactive Power Import
reactive_pwr_neg_phase_a	Decimal	Reactive Power Export Phase A
reactive_pwr_neg_phase_b	Decimal	Reactive Power Export Phase B
reactive_pwr_neg_phase_c	Decimal	Reactive Power Export Phase C
aggregate_reactive_pwr_neg	Decimal	Aggregate Reactive Power Export
aggregate_reactive_pwr_abs	Decimal	Aggregate Reactive Power Absolute as per Pakistan standard
average_pf	Decimal	Average Power Factor
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database
is_synced	Int	0 By default 1 if data is safe to delete from MDC
🔑 id	Bigint	Auto-increment (Not NULL) Unique-ID

3.1.5 EVENTS

Events data structure is for recording of the alerts and alarms generated by the meter.

TABLE ID: EVENTS

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑event_datetime	Datetime	Meter Date & Time when event occurred
event_code	Int	Event Code
event_counter	Bigint	Optional Event Counter
event_description	Varchar	Event Name/Description
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database

is_synced	Int	0 By default 1 if data is safe to delete from MDC
🔑 id	Bigint	Auto-increment (Not NULL) Unique-ID

Below are required events with details

Event Code	Event Name	Description	Class
101	MDI Reset	Monthly MDI RESET	All devices
102	Parameterization	Update Parameter of Meter	All devices
111	Power Fail Start	Power Supply Down	All devices
112	Power Fail End	Power Supply Up	All devices
113	Phase Failure	Any Phase Supply Down	All devices
114	Over Voltage	Voltage more than Limit	All devices
115	Under Voltage	Voltage less than Limit	All devices
116	Demand Overload	Connected load is more than programmed value	All devices
117	Reverse Energy	Reverse direction of current	3P, APMS
118	Reverse Polarity	Phase and Neutral interchange	3P, APMS
121	CT Bypass	Current not flowing through CT Start	3P, APMS
122	CT Bypass End	Current not flowing through CT End	3P, APMS
131	Phase Failure L1 Start	L1 Phase Supply Down	3p, APMS
132	Phase Failure L1 End	L1 Phase Supply Up	3p, APMS
133	Phase Failure L2 Start	L2 Phase Supply Down	3p, APMS
134	Phase Failure L2 End	L2 Phase Supply Up	3p, APMS
135	Phase Failure L3 Start	L3 Phase Supply Down	3p, APMS
136	Phase Failure L3 End	L3 Phase Supply Up	3p, APMS
137	Phase failure Trip	Trip on Any Phase Supply Down	APMS
138	Over Voltage L1 Start	L1 Voltage more than Limit	APMS
139	Over Voltage L1 End	L1 Voltage within Limit	APMS
140	Over Voltage L2 Start	L2 Voltage more than Limit	APMS
141	Over Voltage L2 End	L2 Voltage within Limit	APMS
142	Over Voltage L3 Start	L3 Voltage more than Limit	APMS
143	Over Voltage L3 End	L3 Voltage within Limit	APMS
144	Over Voltage Trip	Trip on Voltage more than Limit	APMS
145	Under Voltage L1 Start	L1 Voltage less than Limit	APMS
146	Under Voltage L1 End	L1 Voltage within Limit	APMS
147	Under Voltage L2 Start	L2 Voltage less than Limit	APMS

Event Code	Event Name	Description	Class
148	Under Voltage L2 End	L2 Voltage within Limit	APMS
149	Under Voltage L3 Start	L3 Voltage less than Limit	APMS
150	Under Voltage L3 End	L3 Voltage within Limit	APMS
151	Under Voltage Trip	Trip on Voltage less than Limit	APMS
152	Phase Over Current L1 Start	L1 Current more than Limit	APMS
153	Phase Over Current L1 End	L1 Current within Limit	APMS
154	Phase Over Current L2 Start	L2 Current more than Limit	APMS
155	Phase Over Current L2 End	L2 Current within Limit	APMS
156	Phase Over Current L3 Start	L3 Current more than Limit	APMS
157	Phase Over Current L3 End	L3 Current within Limit	APMS
158	Phase Over Current Trip	Trip on Current more than Limit	APMS
159	Demand Overload End	Connected load is less than programmed value	All Devices
160	Overload Start	Connected load is more than Overload Limit	APMS
161	Overload End	Connected load is within Overload Limit	APMS
162	Over Load Trip	Trip on Load more than Overload Limit	APMS
163	Voltage Unbalance Start	Difference in Voltage of any two phases more than programmed limit.	APMS
164	Voltage Unbalance End	Difference in Voltage of any two phases is within limit.	APMS
165	Current Unbalance Start	Difference in current of any two phases more than programmed limit.	APMS
166	Current Unbalance End	Difference in current of any two phases is within limit.	APMS
167	High Apparent Power Start	Value of Apparent Power more than Limit	APMS
168	High Apparent Power End	Value of Apparent Power is within Limit	APMS
169	Remote Switch ON	Relay Connected Remotely	APMS
170	Remote Switch OFF	Relay Dis-connected Remotely	APMS
171	Automatic Switch ON	Relay Automatically Connected	APMS
172	Automatic Switch OFF	Relay Automatically Dis-connected	APMS
173	Manual Switch ON	Relay Manually Connected	APMS
174	Manual Switch OFF	Relay Manually Dis-connected	APMS
175	Residual Current Alarm Start	Value of Residual Current more than Limit	APMS
176	Residual Current Alarm End	Value of Residual Current is within Limit	APMS
177	Residual Current Trip	Trip on Value of Residual Current more than Limit	APMS
179	Short Circuit Trip	Trip on Short Circuit Occur	APMS
180	MCCB Trip	Main Breaker Trip	APMS

Event Code	Event Name	Description	Class
187	Reverse Energy L1 Start	Reverse direction of current flowing through L1	3p
188	Reverse Energy L1 End	Forward direction of current flowing through L1	3p
189	Reverse Energy L2 Start	Reverse direction of current flowing through L2	3p
190	Reverse Energy L2 End	Forward direction of current flowing through L2	3p
191	Reverse Energy L3 Start	Reverse direction of current flowing through L3	3p
192	Reverse Energy L3 End	Forward direction of current flowing through L3	3p
193	Reverse Polarity L1 Start	Phase L1 and Neutral interchange	3p
194	Reverse Polarity L1 End	Phase L1 and Neutral in accurate terminal	3p
195	Reverse Polarity L2 Start	Phase L2 and Neutral interchange	3p
196	Reverse Polarity L2 End	Phase L2 and Neutral in accurate terminal	3p
197	Reverse Polarity L3 Start	Phase L3 and Neutral interchange	3p
198	Reverse Polarity L3 End	Phase L3 and Neutral in accurate terminal	3p
201	Time Synchronization	Meter Datetime adjusted	All Devices
202	Contactor ON	Meter's Relay Connect	All Devices with Relay / Breaker
203	Contactor OFF	Meter's Relay Dis-Connect	All Devices with Relay / Breaker
206	Door Open	BOX Door Open	All Devices
207	Battery Low	Meter's Battery Low	All Devices
208	Memory Failure	Meter's Memory got failed	All Devices
301	Optical Port Login	Optical Port connected with Meter	All Devices
303	Sanction Load Control Programmed	Sanction Load Limit update in Meter	3p, 1p
304	Load Shedding Schedule Programmed	Load Shedding Schedule updated in Meter	All Devices
305	IP & Port Programmed	IP & Port updated in Meter	All Devices
306	Time of Use Programmed	Time of Use information updated in Meter	3P
307	Wakeup SIM Programmed	Wakeup SIM Numbers Updated in Meter	All Devices
308	Over Voltage Function Programmed	Over Voltage Limit Updated in Meter	APMS
309	Under Voltage Function Programmed	Under Voltage Limit Updated in Meter	APMS
310	Over Current Function Programmed	Over Current Limit Updated in Meter	APMS
311	Over Load Function Programmed	Under Current Limit Updated in Meter	APMS
312	Phase Failure Function Programmed	Phase Failure threshold value Updated in Meter	APMS
313	Voltage Unbalance Function Programmed	Voltage Unbalance threshold percentage Updated in Meter	APMS
314	Current Unbalance Function Programmed	Current Unbalance threshold percentage Updated in Meter	APMS

Event Code	Event Name	Description	Class
315	High Apparent Power Function Programmed	High Apparent Power threshold value Updated in Meter	APMS
316	Over Voltage Function Cancelled	Over Voltage Function Cancelled	APMS
317	Under Voltage Function Cancelled	Under Voltage Function Cancelled	APMS
318	Over Current Function Cancelled	Over Current Function Cancelled	APMS
319	Over Load Function Cancelled	Over Load Function Cancelled	APMS
320	Phase Failure Function Cancelled	Phase Failure Function Cancelled	APMS
321	Voltage Unbalance Function Cancelled	Voltage Unbalance Function Cancelled	APMS
322	Current Unbalance Function Cancelled	Current Unbalance Function Cancelled	APMS
323	High Apparent Power Function Cancelled	High Apparent Power Function Cancelled	APMS
324	Sanction Load Control Cancelled	Sanction Load Control Cancelled	1P, 3P
325	Load Shedding Schedule Cancelled	Load Shedding Schedule Cancelled	All Devices
326	Time of use Programmed Cancelled	Time of use Programmed Cancelled	3P

Events to be configured as ‘Major Alarms’ are following and they will be reported immediately upon their Occurrence irrespective of communication interval.

Normal Meters
Battery low (207), Contactor ON (202), Contactor Off (203), Door Open (206), Reverse energy (117)

Bi-Directional Meters
Battery low (207), Contactor ON (202), Contactor Off (203), Door Open (206)

APMS
Battery low (207), Contactor ON (202), Contactor Off (203), Door Open (206), Phase failure Trip (137), Over Voltage Trip (144), Under Voltage Trip (151), Phase Over Current Trip (158), Over Load Trip (162), Residual Current Trip (177), Short Circuit Trip (179), MCCB Trip (180)

3.1.6 METER VISUALS

The visual record presents the latest information about the meter. This information gets replaced with the next latest update received from the meter. MDC will intelligently update changed columns in this table.

TABLE ID: ***METER_VISUALS***

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑 global_device_id	Varchar	Unique ID for each device by MDM
last_command	Varchar	Last Command Sent to Meter
last_command_datetime	Datetime	Date & Time of Last Command
last_command_resp	Varchar	Last Command Response
last_command_resp_datetime	Datetime	Response Date & Time of Last Command
meter_datetime	Datetime	Meter Date & Time
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry/Update Date & Time in Database
active_energy_pos_tl	Decimal	Value of Positive Absolute kWh
active_energy_pos_tl_datetime	Datetime	Latest Reporting Date & Time for kWh
active_energy_neg_tl	Decimal	Value of Negative Absolute kWh
active_energy_neg_tl_datetime	Datetime	Latest Reporting Date & Time for kWh
reactive_energy_pos_tl	Decimal	Value of Positive Absolute kVARh
reactive_energy_pos_tl_datetime	Datetime	Latest Reporting Date & Time for kVARh
reactive_energy_neg_tl	Decimal	Value of Negative Absolute kVARh
reactive_energy_neg_tl_datetime	Datetime	Latest Reporting Date & Time for kVARh
aggregate_active_energy_abs_tl	Decimal	Total absolute Active kVARh
aggregate_active_energy_abs_tl_datetime	Datetime	Latest Reporting Date & Time for Total absolute Active kVARh
aggregate_active_pwr_pos	Decimal	Aggregate Active Power Import
aggregate_active_pwr_pos_datetime	Datetime	Latest Reporting Date & Time for positive kW
aggregate_active_pwr_neg	Decimal	Aggregate Active Power Export
aggregate_active_pwr_neg_datetime	Datetime	Latest Reporting Date & Time for negative kW
aggregate_reactive_pwr_pos	Decimal	Aggregate Reactive Power Import
aggregate_reactive_pwr_pos_datetime	Datetime	Latest Reporting Date & Time for positive kVAR
aggregate_reactive_pwr_neg	Decimal	Aggregate Reactive Power Export
aggregate_reactive_pwr_neg_datetime	Datetime	Latest Reporting Date & Time for negative kVAR
Aggregate_reactive_energy_abs_tl	Decimal	Total Reactive kVARh as per Pakistan standard
aggregate_reactive_energy_abs_tl_datetime	Datetime	Latest Reporting Date & Time for Reactive kVArh
current_phase_a	Decimal	Current Phase A

Table Field	Data Type	Description
current_phase_b	Decimal	Current Phase B
current_phase_c	Decimal	Current Phase C
voltage_phase_a	Decimal	Voltage Phase A
voltage_phase_b	Decimal	Voltage Phase B
voltage_phase_c	Decimal	Voltage Phase C
frequency	Decimal	Frequency in Hertz
average_pf	Decimal	Average Power Factor
last_contactor_on_datetime	Datetime	Last Contactor 'On Date & Time'
last_contactor_off_datetime	Datetime	Last Contactor 'Off Date & Time'
last_communication_datetime	Datetime	Last Date & Time of MDC on which device communicated
last_signal_strength	Decimal	Latest Signal Strength in dBm
power_status	Int	'0' for power off (power failure start event) '1' for power on (power failure end event)
power_status_datetime	Datetime	Date Time at which power failure start or end event occurred
auxr_datetime	Datetime	Date Time at which meter relay investigated
auxr_status	Int	Current Contactor status. 1 for 'On', 0 for 'Off'
dvtm_datetime	Datetime	Date Time at which meter clock investigated
dvtm_meter_clock	Datetime	Actual Meter Date Time
sanc_datetime	Datetime	Date Time at which meter sanctioned load investigated
sanc_load_limit	Varchar	Threshold Limit for kW
sanc_maximum_retries	Varchar	Maximum Retries in number
sanc_retry_interval	Varchar	Time Interval for retry in seconds
sanc_threshold_duration	Varchar	Duration to accept threshold crossing limit in seconds
sanc_retry_clear_interval	Varchar	Time after retries count is cleared in seconds
lsch_datetime	Datetime	Date Time at which meter sanctioned load investigated
lsch_start_datetime	Datetime	Date Time for starting load shedding schedule
lsch_end_datetime	Datetime	Date Time till load shedding schedule lasts
lsch_load_shedding_slabs	Text	JSON/XML Array Containing action time & relay operate values for multiple time slabs with relay status. Relay Operate value will be 0 for Disconnect & 1 for Connect

Table Field	Data Type	Description
tiou_datetime	Datetime	Date Time at which Time of use investigated
tiou_day_profile	Text	Associative Array containing multiple list of day profiles having name and tariff slabs array containing starting time (hh:mm) of tariffs in ascending order
tiou_week_profile	Text	Associative Array containing multiple list of week profiles having name and weekly day profiles array containing day profiles in ascending order of week starting from Monday to Sunday.
tiou_season_profile	Text	Associative Array containing multiple list of season profiles having name mapped with week profile name and start date (dd-MM)
tiou_holiday_profile	Text	Associative Array containing multiple list of holiday profiles having name mapped with execution date (dd-MM) and day profile name to be implemented
tiou_activation_datetime	Datetime	Date Time for the activation of condition
ippo_datetime	Datetime	Date Time at which IP & Port investigated
ippo_primary_ip_address	Varchar	New Primary IP Address
ippo_secondary_ip_address	Varchar	New Secondary IP Address
ippo_primary_port	Varchar	New Primary Port Address
ippo_secondary_port	Varchar	New Secondary Port Address
mdsm_datetime	Datetime	Date Time at meter data sampling investigated
mdsm_activation_datetime	Datetime	Date Time for the activation of condition
mdsm_data_type	Varchar	Valid data type values will be 'INST' for Instantaneous Data 'BILL' for Billing Profiles 'LPRO' for Load Profile
mdsm_sampling_interval	Varchar	Sampling Interval in Minutes ranges from 1 – 59
mdsm_sampling_initial_time	Varchar	Start Time Slot ranges from 0 to 59 e.g. If Sampling interval is set to 15 and Sampling Initial Time is 0 then data will be stored in meter at following time interval 00:00, 15:00, 30:00, 45:00 of every hour on daily basis

Table Field	Data Type	Description
oppo_datetime	Datetime	Date Time at which Optical Port investigated
oppo_optical_port_on_datetime	Datetime	Date & Time to enable Optical Port
oppo_optical_port_off_datetime	Datetime	Date & Time to disable Optical Port
wsim_datetime	Datetime	Date Time at which wakeup SIM number investigated
wsim_wakeup_number_1	Varchar	SIM NUMBER1 for wakeup
wsim_wakeup_number_2	Varchar	SIM NUMBER2 for wakeup
wsim_wakeup_number_3	Varchar	SIM NUMBER3 for wakeup
mtst_datetime	Datetime	Date Time at which meter status investigated
mtst_meter_activation_status	Varchar	1 for Active, 0 for Inactive Meters
dmdt_datetime	Datetime	Date Time at which device metadata investigated
dmdt_communication_mode	Varchar	'1' for GPRS/3G/4G '2' for RF '3' for PLC '4' for Ethernet '5' for other
dmdt_bidirectional_device	Varchar	'0' for False '1' for True
dmdt_communication_type	Varchar	'1' for Mode-I/Non-Keep-alive '2' for Mode-II/Keep-alive
dmdt_communication_interval	Varchar	Communication Interval in Minutes if Communication Type value is SET for '1' Otherwise 0
dmdt_initial_communication_time	Varchar	Initial Communication Time for meter in 24 Hrs in "HH:mm:ss" Format
dmdt_phase	Varchar	'1' for Single '2' for other '3' for Three-phase
dmdt_meter_type	Varchar	'1' for Normal '2' for Whole Current '3' for CTO '4' for CTPT '5' for other
mdi_reset_date	Int	Date of Every Month Ranges from 1 to 28
mdi_reset_time	Varchar	Time at which MDI Reset Occurs
msim_id	Varchar	SIM ID of SIM that is used by meter for communication

3.1.7 DEVICE COMMUNICATION HISTORY

Device Communication history data structure will be used for displaying all communications attempts of device along with status of success or failure and complete debug log message.

"message_log" will display proper log against each type data type. Sample format is given below

Communication history will be maintained for every non-keep alive (Mode-I) device/meter upon every connection and same will be maintained for every keep alive (Mode-II) upon every data packet share with MDC.

TABLE ID: *DEVICE_COMMUNICATION_HISTORY*

Table Field	Data Type	Description
msn	Bigint	Meter Serial Number
🔑global_device_id	Varchar	Unique ID for each device by MDM
🔑meter_datetime	Datetime	Meter Date & Time
communication_status	Int	1 for 'Successful', 0 for 'Failure'
message_log	Text	Message Log showing communication insights using *UDIL Codes, including timestamps. Sample Message Logs are shown below
longitude	Decimal	Longitude of Device
latitude	Decimal	Latitude of Device
bytes_sent	Int	Size of Total Data sent to Meter (in Bytes) Data upload from MDC to Meter
bytes_received	Int	Size of Total Data received from Meter (in Bytes) Data download from Meter to MDC
bytes_remaining	Int	Remaining bytes of data package in SIM for meter communication (Optional)
sim_id	Varchar	Inserted SIM ID in Meter Modem
connection_duration	Int	Total Duration in Seconds for which Connection between MDC & Meter lasted.
mdc_read_datetime	Datetime	Reading Date & Time of MDC
db_datetime	Datetime	Record Entry Date & Time in Database
is_synced	Int	0 By default 1 if data is safe to delete from MDC
🔑 id	Bigint	Auto-increment (Not NULL) Unique-ID

*UDIL Codes To keep log messages short yet meaningful use UDIL codes such as for 'Load Profile' use 'LPRO' likewise for 'Monthly Billing' use 'MBIL' etc.

Sample for Meter Connection Timeout

message_log:

```
2022-08-02 00:12:25 Establishing Connection with Meter
2022-08-02 00:15:25 No Response. Timeout.
```

Sample for Authentication Failure

message_log:

```
2022-08-02 00:12:25 Establishing Connection with Meter
2022-08-02 00:12:30 Meter IP: 111.22.33.444:5555.
2022-08-02 00:12:32 Connection Established.
2022-08-02 00:12:32 Login
2022-08-02 00:12:32 Failed
```

Sample for Successful Communication

Comments are only for explanation and not to be included in log.

The number of Quantities is shown for reference, and it can vary depending on the type of Meter (Unidirectional or Bidirectional) and on other factors.

message_log:

```
2022-08-02 00:12:25 Establishing Connection with Meter
2022-08-02 00:12:30 Meter IP: 111.22.33.444:5555
2022-08-02 00:12:32 Connection Established.
2022-08-02 00:12:32 Login
2022-08-02 00:12:32 Success
2022-08-02 00:12:32 FW_VER       -- Getting Firmware Protocol Version
2022-08-02 00:12:33 DEMO_METER v1.2.3
2022-08-02 00:12:33 MSN?         -- Getting MSN
2022-08-02 00:12:34 12345678     -- MSN Received
2022-08-02 00:12:34 TIME?        -- Getting Meter Time
2022-08-02 00:12:35 2022-08-02 00:12:35
2022-08-02 00:12:35 INST?        -- Requesting Instantaneous Data
2022-08-02 00:12:40 OK, 17       -- 17 of 17 Quantities received
2022-08-02 00:12:40 BILL?        -- Requesting Billing Data
2022-08-02 00:12:45 OK, 26       -- 26 of 26 Quantities Received
2022-08-02 00:12:45 MBIL?        -- Requesting Monthly Billing Data
2022-08-02 00:13:00 OK, 28       -- 28 of 28 Quantities Received
2022-08-02 00:13:00 LPRO?        -- Requesting Load Profile Data
2022-08-02 00:13:12 OK, 54       -- 54 of 54 Quantities Received
2022-08-02 00:13:15 EVNT?        -- Requesting Events
2022-08-02 00:13:00 OK, 3        -- 3 of 3 Quantities Received
2022-08-02 00:13:00 DC           -- Disconnected
```

Sample With only Load Profile Failed

Comments are only for explanation and not to be included in log.

The number of Quantities is shown for reference, and it can vary depending on the type of Meter (Unidirectional or Bidirectional) and on other factors.

message_log:

```
2022-08-02 00:12:25 Establishing Connection with Meter
2022-08-02 00:12:30 Meter IP: 111.22.33.444:5555
2022-08-02 00:12:32 Connection Established.
2022-08-02 00:12:32 Login
2022-08-02 00:12:32 Success
2022-08-02 00:12:32 FW_VER       -- Getting Firmware Protocol Version
2022-08-02 00:12:33 DEMO_METER v1.2.3
2022-08-02 00:12:33 MSN?         -- Getting MSN
2022-08-02 00:12:34 12345678     -- MSN Received
2022-08-02 00:12:34 TIME?        -- Getting Meter Time
2022-08-02 00:12:35 2022-08-02 00:12:35
2022-08-02 00:12:35 INST?        -- Requesting Instantaneous Data
2022-08-02 00:12:40 OK, 17       -- 17 of 17 Quantities received
2022-08-02 00:12:40 BILL?        -- Requesting Billing Data
2022-08-02 00:12:45 OK, 26       -- 26 of 26 Quantities Received
2022-08-02 00:12:45 MBIL?        -- Requesting Monthly Billing Data
2022-08-02 00:12:50 FAIL, 20     -- 20 of 28 Quantities Received. Error
2022-08-02 00:12:50 MBIL?        -- Requesting Monthly Billing Data Try 2
2022-08-02 00:13:00 OK, 28       -- 28 of 28 Quantities Received
2022-08-02 00:13:00 LPRO?        -- Requesting Load Profile Data
2022-08-02 00:13:10 FAIL, 31     -- 31 of 54 Quantities Received. Error
2022-08-02 00:13:10 LPRO?        -- Requesting Load Profile Data Try 2
2022-08-02 00:13:12 FAIL, 13     -- 13 of 54 Quantities Received. Error
2022-08-02 00:13:12 LPRO?        -- Requesting Load Profile Data Try 3
2022-08-02 00:13:15 FAIL, 22     -- 22 of 54 Quantities Received. Error
2022-08-02 00:13:15 EVNT?        -- Requesting Events
2022-08-02 00:13:00 OK, 3        -- 3 of 3 Quantities Received
2022-08-02 00:13:00 DC           -- Disconnected
```

3.2 AUTHORIZATION PROTOCOL

The Application Programming Interface (API) requests, as part of the universal data integration layer will require authentication before any communication.

For secure communication between the MDCs and the MDM system, through universal data integration layer, there is a requirement of passing a unique key. The key shall be time bound for 30 minutes minimum and the following function will be passed on to the MDC for getting the unique identification key. All colored input parameters i.e. (username, password, code, transactionid, privatekey) will be supplied in Header of all web services

3.2.1 AUTHORIZATION SERVICE

SERVICE NAME: ***AUTHORIZATION_SERVICE***

Input Parameters

Function	Argument	Data Type	Description
Param	username	String	Username for Web Service
Param	password	String	Password for Web Service
Param	code	String	Special Code for Web Service

Return Values

Param	status	String	0 for Failed, 1 for Success
Param	privatekey	String	Encrypted Value for using APIs
Param	message	String	Optional Message

3.3 WRITE REQUESTS – COMMANDS

There are several commands that may be required to pass on to meters for performing certain functions based on need basis. MDM being the central authority should pass on these commands to various MDC on standard format through universal data integration layer. These commands and their parameters are defined below. MDCs are required to interpret these commands correctly and subsequently perform and acknowledge back to MDM system through universal data integration layer.

- All API/services will follow HTTPS protocol for secure and encrypted sharing of information.
- All API/services will follow standard HTTP codes for each response with proper input fields values & range validations
  - In case of valid request and response, service will send status =1 along with HTTP Status Code of 200.
  - Similarly, in case of Un-Authorize call, service will send status = 0 along with HTTP Status Code of 401 Unauthorized.
  - Similarly, in case of In-Valid request format, service will send Status = 0 along with HTTP Status Code of 400 Bad Request and error in message/remarks.
- All Input and Output Parameters are of string type, array or as defined in the data type column.
- All Datetime fields in Tables/APIs should follow "yyyy-mm-dd hh:mm:ss" format.
- For all indv_status returned with "0" value in response, MDC must provide proper reason in "remarks" variable.
- MDC must update execute write commands on all devices/meters whenever device/meter establishes communication and update command status_level in transaction status in **TRANSACTION_STATUS** table at section 3.4.3.

3.3.1 AUX RELAY OPERATIONS

The objective is to remotely disconnect/reconnect a meter.

SERVICE NAME: ***AUX_RELAY_OPERATIONS***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made
Param	relay_operate	String	0 for Disconnect, 1 for Connect
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.2 TIME SYNCHRONIZATION

The objective is to synchronize meter time with the MDC time.

SERVICE NAME: ***TIME_SYNCHRONIZATION***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message

Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn & indv_status & remarks for representing individual status o f command against each meter

3.3.3 SANCTIONED LOAD CONTROL

The load limiting service is to be used to disconnect the supply (with warnings) after exceeding the sanctioned load range.

SERVICE NAME: ***SANCTIONED_LOAD_CONTROL***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	load_limit	String	Threshold Limit for kW
Param	maximum_retries	String	Maximum Retries in number
Param	retry_interval	String	Time Interval for retry in seconds
Param	threshold_duration	String	Duration to accept threshold crossing limit in seconds
Param	retry_clear_interval	String	Time after retries count is cleared in seconds
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.4 LOAD SHEDDING SCHEDULING

The load management service is to be used to auto disconnect/reconnect the supply on defined intervals as set by the DISCO.

SERVICE NAME: ***LOAD_SHEDDING_SCHEDULING***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	start_datetime	String	Date Time for starting load shedding schedule
Param	end_datetime	String	Date Time till load shedding schedule lasts
Param(set)	load_shedding_slabs	JSON/XML array	JSON/XML Array Containing action_time & relay_operate values for multiple time slabs with relay status. relay_operate value will be 0 for Disconnect & 1 for Connect
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.5 UPDATE TIME OF USE

Service to be used for remotely conducting possible changes of time slots for peak and off-peak readings.

SERVICE NAME: ***UPDATE_TIME_OF_USE***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	day_profile	String	Associative Array containing multiple list of day profiles having name and tariff_slabs array containing starting time (hh:mm) of tariffs in ascending order
Param	week_profile	String	Associative Array containing multiple list of week profiles having name and weekly_day_profile array containing day profiles in ascending order of week starting from Monday to Sunday.
Param	season_profile	String	Associative Array containing multiple list of season profiles having name mapped with week_profile_name and start_date (dd-MM)
Param	holiday_profile	String	Associative Array containing multiple list of holiday profiles having name mapped with date (dd-MM) and day_profile_name to be implemented
Param	activation_datetime	String	Date Time for the activation of condition
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.6 DEVICE CREATION

Service to be used for first time creation of a device in the MDC system. This service will act as update parameters if device is already created:

Mode-I/non-Keep-alive & Mode-II/Keep-alive is mandatory for every device / meter as per DDS.

If the device is converted from Uni-directional to Bi-directional then behavior of meter should change according to Bi-directional parameters including display sequence.

Meter programmed in Mode-I/non-Keep-alive mode must support both Wakeup mechanisms as defined in DDS-110. Wakeup Password must be alpha-numeric.

- Wakeup by SMS
- Wakeup by Call

SERVICE NAME: ***DEVICE_CREATION***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	device_identity	String	Associative array containing "dsn" as Device Serial Number(s) / Meter Serial Number(s) along with "global_device_id" as unique identifier of device
Param(set)	device_type	String	1 for Meter 2 for DCU 3 for APMS 4 for Grid meter 5 for others
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made

Param	communication_mode	String	‘1’ for GPRS/3G/4G ‘2’ for RF ‘3’ for PLC ‘4’ for Ethernet ‘5’ for other
Param	bidirectional_device	String	‘0’ for False ‘1’ for True
Param	communication_type	String	‘1’ for Mode-I/Non-Keep-alive ‘2’ for Mode-II/Keep-alive
Param	initial_communication_time	String	Time in 24 Hrs in “HH:mm:ss” Format
Param	communication_interval	String	Communication Interval in Minutes if Meter in in Mode-1/Non-Keepalive Otherwise 0
Param	sim_number	String	SIM Number
Param	sim_id	String	SIM ID
Param	mdi_reset_date	Int	Date of Every Month Ranges from 1 to 28
Param	mdi_reset_time	String	Time at which MDI Reset Occurs
Param	phase	String	‘1’ for Single ‘2’ for other ‘3’ for Three-phase
Param	meter_type	String	‘1’ for Normal ‘2’ for Whole Current ‘3’ for CTO ‘4’ for CTPT ‘5’ for other
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.7 UPDATE IP PORT

Service to be used for changing the IP Address and port of meter.

SERVICE NAME: ***UPDATE_IP_PORT***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	primary_ip_address	String	New Primary IP Address
Param	secondary_ip_address	String	New Secondary IP Address
Param	primary_port	String	New Primary Port Address
Param	secondary_port	String	New Secondary Port Address
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.8 METER DATA SAMPLING

Service to be used for configuring sampling interval of data stored in meter.

SERVICE NAME: ***METER_DATA_SAMPLING***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made
Param	activation_datetime	String	Date Time for the activation of condition
Param	data_type	String	Valid data type values will be 'INST' for Instantaneous Data 'BILL' for Billing Profiles 'LPRO' for Load Profile
Param	sampling_interval	String	Sampling Interval in Minutes
Param	sampling_initial_time	String	Start Time Slot ranges from 0 to 59 e.g. If Sampling interval is set to 15 and Sampling Initial Time is 0 then data will be stored in meter at following time interval 00:00, 15:00, 30:00, 45:00 of every hour on daily basis
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.9 ACTIVATE METER OPTICAL PORT

This service is used to activate the Optical Port of the meter for the authorized user for a defined time limit. MDC will send activate/deactivate command to the meter to let the user access the optical port.

SERVICE NAME: ***ACTIVATE_METER_OPTICAL_PORT***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key
Param	optical_port_on_datetime	String	Date & Time to enable Optical Port
Param	optical_port_off_datetime	String	Date & Time to disable Optical Port

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.10 UPDATE WAKE UP SIM NUMBER

Service to be used for changing SIM numbers from which wakeup SMS or voice call is allowed.

SERVICE NAME: ***UPDATE_WAKE_UP_SIM_NUMBER***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made

Param	wakeup_number_1	String	11 digit SIM Number for wakeup e.g. 03*********
Param	wakeup_number_2	String	11 digit SIM Number for wakeup e.g. 03*********
Param	wakeup_number_3	String	11 digit SIM Number for wakeup e.g. 03*********
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.11 UPDATE METER STATUS

This service is used to change the status of meter to active or inactive. MDC will only entertain/read meter having meter activation status equal to 1. All meters with meter activation status equal to 0 will be considered stopped due to any work flow operation.

SERVICE NAME: ***UPDATE_METER_STATUS***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key
Param	meter_activation_status	String	1 for Active, 0 for Inactive Meters

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success

Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.12 UPDATE DEVICE META DATA

Service to be used for communication modes, type, interval and phase of meter.

SERVICE NAME: ***UPDATE_DEVICE_METADATA***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	communication_mode	String	'1' for GPRS/3G/4G '2' for RF '3' for PLC '4' for Ethernet '5' for other
Param	bidirectional_device	String	'0' for False '1' for True
Param	communication_type	String	'1' for Mode-I/non-Keep-alive '2' for Mode-II/Keep-alive
Param	initial_communication_time	String	Time in 24 Hrs in "HH:mm:ss" Format
Param	communication_interval	String	Communication Interval in Minutes if Communication Type value is SET for '1' Otherwise 0
Param	phase	String	'1' for Single '2' for other '3' for Three-phase
Param	meter_type	String	'1' for Normal '2' for Whole Current '3' for CTO '4' for CTPT '5' for other
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.13 UPDATE MDI RESET DATE

This service is used to update MDI RESET DATE of meter.

SERVICE NAME: ***UPDATE_MDI_RESET_DATE***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key
Param	mdi_reset_date	Int	Date of Every Month Ranges from 1 to 28
Param	mdi_reset_time	String	Time at which MDI Reset Occurs

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.14 APMS TRIPPING EVENTS

The purpose of this service is to set threshold limits for controlling tripping of APMS. Parameter named "type" will decide nature of tripping function.

- Type 'OVFC' – Over Voltage Function
- Type 'UVFC' – Under Voltage Function
- Type 'OCFC' – Over Current Function
- Type 'OLFC' – Over Load Function
- Type 'VUFC' – Voltage Unbalance Function
- Type 'PFFC' – Phase Failure Function
- Type 'CUFC' – Current Unbalance Function
- Type 'HAPF' – High Apparent Power Function

SERVICE NAME: ***APMS_TRIPPING_EVENTS***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	type	String	Value of tripping function to be invoked Valid types are given above
Param	critical_event_threshold_limit	String	Critical Event/Alarm threshold limiting value having following units Volt for OVFC, UVFC, PFFC Amp for OCFC, kW for OLFC, kVA for HAPF Percentage (%) for VUFC, CUFC
Param	critical_event_log_time	String	Monitoring time before recording event (seconds)
Param	tripping_event_threshold_limit	String	Tripping threshold limiting value having following units Volt for OVFC, UVFC, PFFC Amp for OCFC, OLFC kVA for HAPF Percentage (%) for VUFC, CUFC
Param	tripping_event_log_time	String	Monitoring time before initiating tripping (seconds)
Param	enable_tripping	String	'0' for Disable '1' for Enable
Param	request_datetime	String	Request Date & Time
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.3.15 PARAMETERIZATION_CANCELLATION

This service will be used to cancel programmed AMI operations. After cancelling, these programmed parameters device or meter will switch on factory default settings.
Valid Types for this service are

- Type 'SANC' – Programmed Sanctioned Load
- Type 'LSCH' – Programmed Load Shedding Schedule
- Type 'TIOU' – Programmed Time of Use
- Type 'OVFC' – Over Voltage Function
- Type 'UVFC' – Under Voltage Function
- Type 'OCFC' – Over Current Function
- Type 'OLFC' – Over Load Function
- Type 'VUFC' – Voltage Unbalance Function
- Type 'PFFC' – Phase Failure Function
- Type 'CUFC' – Current Unbalance Function
- Type 'HAPF' – High Apparent Power Function

Input Parameters

Function	Argument	Data Type	Description
Param	global_device_id	String	Unique ID for each device by MDM. Only one id allowed
Param	transactionid	String	Unique key for Request
Param	privatekey	String	Encrypted Key
Param	type	String	Transaction Type Applicable

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

3.4 ON-DEMAND REQUESTS – API BASED

The read requests, as defined above are based on fetching information from the MDC database at defined intervals. However, there can be a need to request the MDC to bring latest information from the meter on ad hoc basis. Note that the meters are not essentially kept on 'Always Alive' mode. Hence to achieve the requirement, MDM must be able to pass the information to MDC through the universal data integration layer to fetch scenario-specific information directly from the meter as defined in section 3.1. For all such data retrieval, ***On-demand synchronous requests*** shall be made to the meter via the MDC. If meter is online then it will give data immediately. The parameter is supplemented with a unique key named 'transactionid' generated by the universal data integration layer for reference purpose i.e. all the requests sent to multiple MDCs will be having a unique key assigned for differentiation. Apart from unique key, there will be a unique variable 'Type' (length 4 characters) passed through for identification of the service request. The unique key will be additionally supplemented with defined set of parameters designed to meet end-user criteria. These 'Types' are defined in Section 3.4.1 & 3.4.2.

For on demand parameter read, Response should be according to relevant columns defined in meter visuals. See examples in Section 4.16 for details.

3.4.1 ON_DEMAND_DATA_READ

Valid Types for this service are

- Type 'INST' – Instantaneous Data
- Type 'BILL' – Billing Profiles
- Type 'MBIL' – Monthly Billing Profile
- Type 'LPRO' – Load Profile
- Type 'EVNT' – Events

For Type 'INST' MDC will ignore value of "start_datetime" & "end_datetime" and will give actual instantaneous data

Input Parameters

Function	Argument	Data Type	Description
Param	global_device_id	String	Unique ID for each device by MDM. Only one id allowed
Param	transactionid	String	Unique key for each Request
Param	privatekey	String	Encrypted Key
Param	type	String	Transaction Type Applicable
Param	start_datetime	String	Start Date & Time of data to be fetched from meter.
Param	end_datetime	String	End Date & Time of data to be fetched from meter.

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of parameters as per data structure defined for 'INST', 'BILL', 'MBIL', 'LPRO', 'EVNT'

3.4.2 ON_DEMAND_PARAMETER_READ

Valid Types for this service are

- Type 'AUXR' – Auxiliary Relay Status
- Type 'DVTM' – Device Time
- Type 'SANC' – Programmed Sanctioned Load
- Type 'LSCH' –Programmed Load Shedding Schedule
- Type 'TIOU' – Programmed Time of Use
- Type 'IPPO' – Programmed IP & Port
- Type 'MDSM' – Programmed Meter Data Sampling
- Type 'OPPO' – Programmed Optical Port
- Type 'WSIM' – Programmed Wake-up SIM Number
- Type 'MSIM' – SIM-ID of SIM inside meter
- Type 'MTST' – Meter Status
- Type 'DMDT' – Device Meta Data
- Type 'MDI' – Programmed MDI Reset Date & Time
- Type 'OVFC' – Over Voltage Function
- Type 'UVFC' – Under Voltage Function
- Type 'OCFC' – Over Current Function
- Type 'OLFC' – Over Load Function
- Type 'VUFC' – Voltage Unbalance Function
- Type 'PFFC' – Phase Failure Function
- Type 'CUFC' – Current Unbalance Function
- Type 'HAPF' – High Apparent Power Function

Input Parameters

Function	Argument	Data Type	Description
Param	global_device_id	String	Unique ID for each device by MDM. Only one id allowed
Param	transactionid	String	Unique key for each Request
Param	privatekey	String	Encrypted Key
Param	type	String	Transaction Type Applicable

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of parameters as will populate meter visuals table.

3.4.3 TRANSACTION STATUS

Each command is required to be maintained in MDC with the latest stage indicator 'msn' and 'global_device_id' wise. This will help the MDM to get to know the status of command it has sent to the meter through the universal data integration layer and for all msn separated in rows. For standard, the command status numbers (status_level) are as follow:

- 0 – refers to waiting for processing.
- 1 – refers to commencing command processing.
- 2 – refers to communication request sent to meter.
- 3 – refers to communication established with meter.
- 4 – refers to command sent to meter.
- 5 – refers to command executed by meter.

MDC is required to store command status for monitoring purposes. After initiating any transaction MDM can query transaction status using 'Input Parameters' (as defined below) to read output response of write requests.

Input Parameters

Function	Argument	Data Type	Description
Param(set)	transactionid	String	Transaction ID(s) from MDM
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of response in format as described in TRANSACTION_STATUS below

TRANSACTION_STATUS

Table Field	Data Type	Description
🔑transactionid	Varchar(Max)	Transaction ID of command from MDM
msn	BigInt	Meter Serial Number
🔑 global_device_id	Varchar(Max)	Unique ID for each device by MDM
type	Varchar(Max)	Type of Request
command_receiving_datetime	DateTime	Date & Time of command received by MDC
status_level	Int	Status level range from 0 – 5. Latest value of status will be stored here. MDC will continuously update this column according to status.
status_1_datetime	DateTime	Date & Time when command processing started by MDC
status_2_datetime	DateTime	Date & Time when request was forwarded to meter from MDC
status_3_datetime	DateTime	Date & Time when meter gets connected with MDC (Hand Shake)
status_4_datetime	DateTime	Date & Time when request forwarded to meter from MDC
status_5_datetime	DateTime	Date & Time when meter successfully executed command
indv_status	Int	'0' for In-Progress.

		'1' for complete. '2' for Failed.
request_cancelled	Int	'0' or otherwise '1' if MDC cancels the request
request_cancel_reason	Varchar	Reason for cancelation of request
request_cancel_datetime	DateTime	Date & time of cancelation of request
response_data	Text	Optional Message

3.4.4 TRANSACTION CANCEL

The purpose of this service is to cancel the command against specific devices. MDC will cancel commands against provided devices (identified by global_device_id) if and only if the command status numbers (status_level) of these devices is 0, 1 or 2 (Section 3.4.3).

TRANSACTION NAME: ***TRANSACTION_CANCEL***

Input Parameters

Function	Argument	Data Type	Description
Param(set)	global_device_id	String	Unique ID for each device by MDM
Param	transactionid	String	Transaction ID from MDM
Param	request_datetime	String	Date & Time at which request is made
Param	privatekey	String	Encrypted Key

Return Values

Param	transactionid	String	Transaction ID from MDM
Param	status	String	0 for Failed, 1 for Success
Param	message	String	Optional Message
Param(set)	data	JSON/XML array	JSON/XML array Containing list of global_device_id, msn, indv_status & remarks for representing command output against each meter

4. IMPLEMENTATION EXAMPLES

Below are the implementation examples for all services.

4.1 AUTHORIZATION SERVICE

Request:

```
Method: POST
URL: https://host:port/authorization_service
```

Headers:

Header Name	Header Value
username	"mdm_user"
password	" WSnt!@SS^12*"
code	"223"

Response:

JSON:

```
{
  "status": "1",
  "privatekey": "7h6g5d4chyhdxg875576v5f5gv8b7bv",
  "message": "You are authenticated successfully. Private key will be valid for 30
Minutes"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
       <status>1</status>
       <privatekey>7h6g5d4chyhdxg875576v5f5gv8b7bv</privatekey>
       <message>You are authenticated successfully. Private key will be valid for 30
Minute</message>
</root>
```

4.2 AUX_RELAY_OPERATIONS

Request:

Method: POST

URL: https://host:port/aux_relay_operations

Headers:

Header Name	Header Value
transactionid	9Xcvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m96541254","m97524158","m98562412"]
request_datetime	"2018-10-15 13:00:00"
relay_operate	"0"

Response:

JSON:

```
{
  "status": "1",
  "transactionid": "9Xcvfg675d5hg85fhjjfved47f65d3s3g",
  "data": [
    {
      "global_device_id": "m96541254",
      "msn": "4096541254",
      "indv_status": "1",
      "remarks": "Command saved successfully for execution"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "0",
      "remarks": "Relay is not available in meter"
    },
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Command saved successfully for execution"
    }
  ],
  "message": "MDC will turn off all meters having individual status 1"
}
```

XML:

```
<?xml version=“1.0” encoding=“UTF-8”?>
<root>
	<status>1</status>
	<transactionid>9Xcvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
	<data>
		<element>
			<global_device_id>m96541254</global_device_id>
			<msn>4096541254</msn>
			<indv_status>1</indv_status>
			<remarks>Command saved successfully for execution</remarks>
		</element>
		<element>
			<global_device_id>m97524158</global_device_id>
			<msn>4097524158</msn>
			<indv_status>0</indv_status>
			<remarks>Relay is not available in meter</remarks>
		</element>
		<element>
			<global_device_id>m98562412</global_device_id>
			<msn>4098562412</msn>
			<indv_status>1</indv_status>
			<remarks>Command saved successfully for execution</remarks>
		</element>
	</data>
	<message>MDC will turn off all meters having individual status 1</message>
</root>
```

4.3 TRANSACTION_STATUS

Request:

Method: POST

URL: https://host:port/transaction_status

Headers:

Header Name	Header Value
transactionid	9Xcvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Response:

JSON:

```
{
  "status": "1",
  "transactionid": "9Xcvfg675d5hg85fhjjfved47f65d3s3g",
  "data": [
    {
      "transactionid": "9Xcvfg675d5hg85fhjjfved47f65d3s3g",
      "global_device_id": "m96541254",
      "msn": "4096541254",
      "type": "AUX_RELAY_OPERATIONS",
      "command_receiving_datetime": "2018-10-15 13:00:15",
      "status_level": "4",
      "status_1_datetime": "2018-10-15 13:00:40",
      "status_2_datetime": "2018-10-15 13:00:50",
      "status_3_datetime": "2018-10-15 13:01:00",
      "status_4_datetime": "2018-10-15 13:01:41",
      "status_5_datetime": "",
      "indv_status": "1",
      "request_cancelled": "0",
      "request_cancel_reason": "",
      "request_cancel_datetime": "",
      "response_data": "Contactor OFF is in process"
    },
    {
      "transactionid": "9Xcvfg675d5hg85fhjjfved47f65d3s3g",
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "type": "AUX_RELAY_OPERATIONS",
      "command_receiving_datetime": "2018-10-15 13:00:15",
      "status_level": "5",
      "status_1_datetime": "2018-10-15 13:00:40",
      "status_2_datetime": "2018-10-15 13:00:50",
      "status_3_datetime": "2018-10-15 13:01:00",
      "status_4_datetime": "2018-10-15 13:01:41",
      "status_5_datetime": "2018-10-15 13:01:45",
      "indv_status": "1",
      "request_cancelled": "0",
      "request_cancel_reason": "",
      "request_cancel_datetime": "",
      "response_data": "Contactor OFF executed successfully on meter"
    }
  ],
  "message": "AUX_RELAY_OPERATIONS response against requested meters"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
        <status>1</status>
        <transactionid>9Xcvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
        <data>
                <transactionid>9Xcvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
                <global_device_id>m96541254</global_device_id>
                <msn>4096541254</msn>
                <type>AUX_RELAY_OPERATIONS</type>
                <command_receiving_datetime>
                        2018-10-15 13:00:15
                </command_receiving_datetime>
                <status_level>4</status_level>
                <status_1_datetime>2018-10-15 13:00:40</status_1_datetime>
                <status_2_datetime>2018-10-15 13:00:50</status_2_datetime>
                <status_3_datetime>2018-10-15 13:01:00</status_3_datetime>
                <status_4_datetime>2018-10-15 13:01:41</status_4_datetime>
                <status_5_datetime></status_5_datetime>
                <indv_status>1</indv_status>
                <request_cancelled>0</request_cancelled>
                <request_cancel_reason></request_cancel_reason>
                <request_cancel_datetime></request_cancel_datetime>
                <response_data>Contactor OFF is in process</response_data>
        </data>
        <data>
                <transactionid>9Xcvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
                <global_device_id>m98562412</global_device_id>
                <msn>4098562412</msn>
                <type>AUX_RELAY_OPERATIONS</type>
                <command_receiving_datetime>
                    2018-10-15 13:00:15
                </command_receiving_datetime>
                <status_level>5</status_level>
                <status_1_datetime>2018-10-15 13:00:40</status_1_datetime>
                <status_2_datetime>2018-10-15 13:00:50</status_2_datetime>
                <status_3_datetime>2018-10-15 13:01:00</status_3_datetime>
                <status_4_datetime>2018-10-15 13:01:41</status_4_datetime>
                <status_5_datetime>2018-10-15 13:01:45</status_5_datetime>
                <indv_status>1</indv_status>
                <request_cancelled>0</request_cancelled>
                <request_cancel_reason></request_cancel_reason>
                <request_cancel_datetime></request_cancel_datetime>
                <response_data>
                    Contactor OFF executed successfully on meter
                </response_data>
            </data>
   <message>AUX_RELAY_OPERATIONS response against requested meters </message>
   </root>
```

4.4 TIME_SYNCHRONIZATION

Request:

Method: POST

URL: https://host:port/time_synchronization

Headers:

Header Name	Header Value
transactionid	7Ucvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"

Response:

JSON:

```
{
  "status": "1",
  "transactionid": "7Ucvfg675d5hg85fhjjfved47f65d3s3g",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Meter time will be updated as System Time"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": " Meter time will be updated as System Time "
    }
  ],
  "message": " Meter time will be updated as System Time for meters having individual status 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
      <status>1</status>
      <transactionid>7Ucvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
      <data>
            <element>
                  <global_device_id>m98562412</global_device_id>
                  <msn>4098562412</msn>
                  <indv_status>1</indv_status>
                  <remarks> Meter time will be updated as System Time </remarks>
            </element>
            <element>
                  <global_device_id>m97524158</global_device_id>
                  <msn>4097524158</msn>
                  <indv_status>1</indv_status>
                  <remarks> Meter time will be updated as System Time </remarks>
            </element>
      </data>
      <message> Meter time will be updated as System Time for meters having individual
status 1</message>
</root>
```

4.5 SANCTIONED_LOAD_CONTROL

Request:

Method: POST

URL: https://host:port/sanctioned_load_control

Headers:

Header Name	Header Value
transactionid	8Ncvfg675d5hg85fhjjfved47f65d3s3g
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
load_limit	"20"
maximum_retries	"10"
retry_interval	"60"
threshold_duration	"30"
retry_clear_interval	"3600"

Response:

JSON:

```
{
  "status": "1",
  "transactionid": "8Ncvfg675d5hg85fhjjfved47f65d3s3g",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Sanctioned Load will be updated"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Sanctioned Load will be updated"
    }
  ],
  "message": "Sanctioned Load will be updated for meters having individual status as 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
    <status>1</status>
    <transactionid>8Ncvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
    <data>
        <element>
            <global_device_id>m98562412</global_device_id>
            <msn>4098562412</msn>
            <indv_status>1</indv_status>
            <remarks>Sanctioned Load will be updated</remarks>
        </element>
        <element>
            <global_device_id>m97524158</global_device_id>
            <msn>4097524158</msn>
            <indv_status>1</indv_status>
            <remarks>Sanctioned Load will be updated</remarks>
        </element>
    </data>
    <message> Sanctioned Load will be updated for meters having individual status as
1</message>
</root>
```

4.6 LOAD_SHEDDING_SCHEDULING

Request:

Method: POST

URL: https://host:port/load_shedding_scheduling

Headers:

Header Name	Header Value
transactionid	R5cvfg675d5hg85fhjjfved47f65d3s3g
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412", "m97524158"]
request_datetime	"2018-10-15 13:00:00"
start_datetime	"2018-11-01 00:00:00"
end_datetime	"2018-12-01 00:00:00"

load_shedding_slabs	[
	  {
	    "action_time": "03:00:00",
	    "relay_operate": "0"
	  },
	  {
	    "action_time": "06:00:00",
	    "relay_operate": "1"
	  },
	  {
	    "action_time": "09:00:00",
	    "relay_operate": "0"
	  },
	  {
	    "action_time": "12:00:00",
	    "relay_operate": "1"
	  }
	]

Response:

JSON:

```
{
  "status": "1",
  "transactionid": "R5cvfg675d5hg85fhjjfved47f65d3s3g",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Load Shedding Schedule will be updated"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Load Shedding Schedule will be updated"
    }
  ],
  "message": "Load Shedding Schedule will be updated for meters having individual
status as 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
        <status>1</status>
        <transactionid>R5cvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
        <data>
                <element>
                        <global_device_id>m98562412</global_device_id>
                        <msn>4098562412</msn>
                        <indv_status>1</indv_status>
                        <remarks>Load Shedding Schedule will be updated</remarks>
                </element>
                <element>
                        <global_device_id>m97524158</global_device_id>
                        <msn>4097524158</msn>
                        <indv_status>1</indv_status>
                        <remarks>Load Shedding Schedule will be updated</remarks>
                </element>
        </data>
        <message>Load Shedding Schedule will be updated for meters having individual
status as 1</message> </root>
```

4.7 UPDATE_TIME_OF_USE

Request:

Method: POST

URL: https://host:port/update_time_of_use

Headers:

Header Name	Header Value
transactionid	3Wcvfg675d5hg85fhjjfved47f65d3s3g
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412","m97524158"]
request_datetime	"2018-10-15 13:00:00"
activation_datetime	"2018-11-01 00:00:00"

day_profile	[
	{
	"name": "d1",
	"tariff_slabs": ["08:00","17:00","20:00"]
	},
	{
	"name": "d2",
	"tariff_slabs": ["08:00","17:00"]
	},
	{
	"name": "d3",
	"tariff_slabs": ["06:00","18:00","20:00"]
	},
	{
	"name": "d4",
	"tariff_slabs": ["08:00","12:00","17:00"]
	}
	]

week_profile	[{

	"name": "w1",
	"weekly_day_profile": [
	"d1","d1","d1","d1","d1","d2","d2"
	]
	},
	{
	"name": "w2",
	"weekly_day_profile": [
	"d1","d1","d1","d1","d1","d3","d3"
	]
	},
	{
	"name": "w3",
	"weekly_day_profile": [
	"d2","d2","d2","d1","d1","d3","d3"
	]
	}
	]

season_profile	[
	{
	"name": "s1",
	"week_profile_name": "w1",
	"start_date": "01-01"
	},
	{
	"name": "s2",
	"week_profile_name": "w1",
	"start_date": "01-04"
	},
	{
	"name": "s3",
	"week_profile_name": "w1",
	"start_date": "01-07"
	},
	{
	"name": "s4",
	"week_profile_name": "w1",
	"start_date": "01-10"
	}
	]

holiday_profile	[

```
  {
    "name": "h1",
    "date": "23-03",
    "day_profile_name": "d4"
  },
  {
    "name": "h2",
    "date": "14-08",
    "day_profile_name": "d4"
  },
  {
    "name": "h3",
    "date": "05-02",
    "day_profile_name": "d4"
  }
]
```

Response:

JSON:

```
{
  "status": "1",
  "transactionid": "3Wcvfg675d5hg85fhjjfved47f65d3s3g",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Tariff register will be applied to meter"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Tariff register will be applied to meter"
    }
  ],
  "message": "Tariff register will be applied to all meters having individual status as 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
	<status>1</status>
	<transactionid>3Wcvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
	<data>
		<element>
			<global_device_id>m98562412</global_device_id>
			<msn>4098562412</msn>
			<indv_status>1</indv_status>
			<remarks>Tariff register will be applied to meter</remarks>
		</element>
		<element>
			<global_device_id>m97524158</global_device_id>
			<msn>4097524158</msn>
			<indv_status>1</indv_status>
			<remarks>Tariff register will be applied to meter</remarks>
		</element>
	</data>
	<message> Tariff register will be applied to all meters having individual status as 1</message>
</root>
```

4.8 DEVICE_CREATION

Request:

```
Method: POST
URL: https://host:port/device_creation
```

Headers:

Header Name	Header Value
transactionid	ATTvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value

device_identity	[
	  {
	    "dsn": "4098562412",
	    "global_device_id": "m98562412"
	  },
	  {
	    "dsn": "4097524158",
	    "global_device_id": "m97524158"
	  }
	]

request_datetime	"2018-10-15 13:00:00"
initial_communication_time	"05:15:41"
communication_interval	"15"
device_type	"1"
mdi_reset_date	"1"
mdi_reset_time	"00:00:00"
sim_number	"03218090100"
sim_id	"8991006090527906588"
phase	"3"
meter_type	"2"
communication_mode	"1"
communication_type	"1"
bidirectional_device	"0"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": "ATTvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Meter Creation in Progress"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Meter Creation in Progress"
    }
  ],
  "message": "Meters with individual status 1 are created successfully"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
       <status>1</status>
       <transactionid>ATTvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
       <data>
              <element>
                     <global_device_id>m98562412</global_device_id>
                     <msn>4098562412</msn>
                     <indv_status>1</indv_status>
                     <remarks>Meter Creation in Progress</remarks>
              </element>
              <element>
                     <global_device_id>m97524158</global_device_id>
                     <msn>4097524158</msn>
                     <indv_status>1</indv_status>
                     <remarks>Meter Creation in Progress</remarks>
              </element>
       </data>
       <message> Meters with individual status 1 are created successfully </message>
</root>
```

4.9 UPDATE_IP_PORT

Request:

Method: POST

URL: https://host:port/update_ip_port

Headers:

Header Name	Header Value
transactionid	0Ycvfg675d5hg85fhjjfved47f65d3s3g
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412", "m97524158"]
request_datetime	"2018-10-15 13:00:00"
primary_ip_address	"192.168.100.120"
secondary_ip_address	"192.168.100.121"
primary_port	"8080"
secondary_port	"8080"

Response:

JSON:

```
{
  "status": "1",
  "transactionid": "0Ycvfg675d5hg85fhjjfved47f65d3s3g",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "IP Addresses will be updated"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "IP Addresses will be updated"
    }
  ],
  "message": "IP Addresses will be updated for all meters having individual status as 
1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
       <status>1</status>
       <transactionid>0Ycvfg675d5hg85fhjjfved47f65d3s3g</transactionid>
       <data>
              <element>
                     <global_device_id>m98562412</global_device_id>
                     <msn>4098562412</msn>
                     <indv_status>1</indv_status>
                     <remarks>IP Addresses will be updated</remarks>
              </element>
              <element>
                     <global_device_id>m97524158</global_device_id>
                     <msn>4097524158</msn>
                     <indv_status>1</indv_status>
                     <remarks>IP Addresses will be updated</remarks>
              </element>
       </data>
       <message> IP Addresses will be updated for all meters having individual status 
as 1</message>
</root>
```

4.10 METER_DATA_SAMPLING

Request:

Method: POST

URL: https://host:port/meter_data_sampling

Headers:

Header Name		Header Value
transactionid		ARbvfg675d5hg85fhjjfved47f5d3aRi
Privatekey		7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key		Parameters Value
global_device_id		["m98562412", "m97524158"]
request_datetime		"2018-10-15 13:00:00"
activation_datetime		"2018-11-01 00:00:00"
data_type		"LPRO"
sampling_interval		"30"
sampling_initial_time		"00"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": " ARbvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Sampling interval of meter will be changed"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Sampling interval of meter will be changed"
    }
  ],
  "message": "Sampling interval of meters with individual status as 1 will be changed
accordingly"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
       <status>1</status>
       <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
       <data>
              <element>
                     <global_device_id>m98562412</global_device_id>
                     <msn>4098562412</msn>
                     <indv_status>1</indv_status>
                     <remarks>Sampling interval of meter will be changed</remarks>
              </element>
              <element>
                     <global_device_id>m97524158</global_device_id>
                     <msn>4097524158</msn>
                     <indv_status>1</indv_status>
                     <remarks>Sampling interval of meter will be changed</remarks>
              </element>
       </data>
       <message>Sampling interval of meters with individual status as 1 will be changed
accordingly
</message>
</root>
```

4.11 ACTIVATE_METER_OPTICAL_PORT

Request:

Method: POST

URL: https://host:port/activate_meter_optical_port

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412", "m97524158"]
request_datetime	"2018-10-15 13:00:00"
optical_port_on_datetime	"2018-10-15 13:00:00"
optical_port_off_datetime	"2018-10-15 15:00:00"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": " ARbvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Optical Port has been activated"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Optical Port has been activated"
    }
  ],
  "message": "Optical Port has been activated for meters having individual status as 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
       <status>1</status>
       <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
       <data>
              <element>
                     <global_device_id>m98562412</global_device_id>
                     <msn>4098562412</msn>
                     <indv_status>1</indv_status>
                     <remarks>Optical Port has been activated</remarks>
              </element>
              <element>
                     <global_device_id>m97524158</global_device_id>
                     <msn>4097524158</msn>
                     <indv_status>1</indv_status>
                     <remarks>Optical Port has been activated</remarks>
              </element>
       </data>
       <message> Optical Port has been activated for meters having individual status as
1</message>
</root>
```

4.12 UPDATE_WAKE_UP_SIM_NUMBER

Request:

Method: POST

URL: https://host:port/update_wake_up_sim_number

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412", "m97524158"]
request_datetime	"2018-10-15 13:00:00"
wakeup_number_1	"03346071928"
wakeup_number_2	"03354445555"
wakeup_number_3	"03212345686"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": " ARbvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Wakeup numbers have been updated"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Wakeup numbers update is in progress"
    }
  ],
  "message": "Wakeup numbers have been updated for meters having individual status as 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
       <status>1</status>
       <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
       <data>
              <element>
                     <global_device_id>m98562412</global_device_id>
                     <msn>4098562412</msn>
                     <indv_status>1</indv_status>
                     <remarks>Wakeup numbers have been updated</remarks>
              </element>
              <element>
                     <global_device_id>m97524158</global_device_id>
                     <msn>4097524158</msn>
                     <indv_status>1</indv_status>
                     <remarks>Wakeup numbers update is in progress</remarks>
              </element>
       </data>
       <message> Wakeup numbers have been updated for meters having individual status as 1</message>
</root>
```

4.13 UPDATE_METER_STATUS

Request:

Method: POST

URL: https://host:port/update_meter_status

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412", "m97524158"]
request_datetime	"2018-10-15 13:00:00"
meter_activation_status	"1"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": " ARbvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Meter Status have been updated"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Meter Status have been updated"
    }
  ],
  "message": "Meter Status have been updated for meters having individual status as 1"
}
```

XML:

```
<?xml version=“1.0” encoding=“UTF-8”?>
<root>
      <status>1</status>
      <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
      <data>
            <element>
                  <global_device_id>m98562412</global_device_id>
                  <msn>4098562412</msn>
                  <indv_status>1</indv_status>
                  <remarks>Meter Status have been updated</remarks>
            </element>
            <element>
                  <global_device_id>m97524158</global_device_id>
                  <msn>4097524158</msn>
                  <indv_status>1</indv_status>
                  <remarks>Meter Status have been updated</remarks>
            </element>
      </data>
      <message> Meter Status have been updated for meters having individual status as
1</message>
</root>
```

4.14 ON_DEMAND_DATA_READ (EVENTS)

Request:

Method: POST

URL: https://host:port/on_demand_data_read

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412"]
start_datetime	"2018-10-15 13:00:00"
end_datetime	"2018-10-22 13:00:00"
type	" EVNT "

Response:

JSON:

```
{
  "status": 1,
  "transactionid": " ARbvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "event_datetime": "2018-10-16 15:14:00",
      "event_code": "111",
      "event_counter": "5",
      "event_description": "Power Fail Start",
      "mdc_read_datetime": "2018-10-16 15:15:00",
      "db_datetime": "2018-10-16 15:16:10",
      "is_sync ": "0"
    },
    {
      "global_device_id": "m98562412 ",
      "msn": "4098562412",
      "meter_datetime": "2018-10-16 16:14:00",
      "event_code": "112",
      "event_counter": "5",
      "event_description": "Power Fail End",
      "mdc_read_datetime": "2018-10-16 16:15:00",
      "db_datetime": "2018-10-16 16:15:10",
      "is_sync ": "0"
    }
  ],
  "message": "Events data from meter between provided dates"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
      <status>1</status>
      <transactionid> ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
      <data>
            <global_device_id>m98562412</global_device_id>
            <msn>4098562412</msn>
            <event_datetime>2018-10-16 15:14:00</event_datetime>
            <event_code>111</event_code>
            <event_counter>5</event_counter>
            <event_description>Power Fail Start</event_description>
            <mdc_read_datetime>2018-10-16 15:15:00</mdc_read_datetime>
            <db_datetime>2018-10-16 15:16:10</db_datetime>
            <is_sync>0</is_sync>
      </data>
      <data>
            <global_device_id>m98562412 </global_device_id>
            <msn>4098562412</msn>
            <meter_datetime>2018-10-16 16:14:00</meter_datetime>
            <event_code>112</event_code>
            <event_counter>5</event_counter>
            <event_description>Power Fail End</event_description>
            <mdc_read_datetime>2018-10-16 16:15:00</mdc_read_datetime>
            <db_datetime>2018-10-16 16:15:10</db_datetime>
            <is_sync>0</is_sync>
      </data>
      <message>Events data from meter between provided dates</message>
</root>
```

4.15 ON_DEMAND_DATA_READ (LOAD PROFILE)

Request:

Method: POST

URL: https://host:port/on_demand_data_read

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	"m98562412"
start_datetime	"2018-10-15 13:00:00"
end_datetime	"2018-10-22 13:00:00"
type	"LPRO"

Response:

JSON:

```
{
  "transactionid": "3fdcc1a20f284fe395436b9e5ef49fca",
  "status": "1",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "meter_datetime": "2020-06-15 09:15:00",
      "channel_id": 1,
      "interval": 15,
      "frequency": "",
      "active_energy_pos_t1": "0.543",
      "active_energy_pos_t2": "42.489",
      "active_energy_pos_t3": "",
      "active_energy_pos_t4": "",
      "active_energy_pos_tl": "43.033",
      "active_energy_neg_t1": "",
      "active_energy_neg_t2": "",
      "active_energy_neg_t3": "",
      "active_energy_neg_t4": "",
      "active_energy_neg_tl": "",
      "active_energy_abs_t1": "0.543",
      "active_energy_abs_t2": "42.489",
      "active_energy_abs_t3": "",
      "active_energy_abs_t4": "",
      "active_energy_abs_tl": "43.033",
      "reactive_energy_pos_t1": "0",
      "reactive_energy_pos_t2": "28.055",
      "reactive_energy_pos_t3": "",
      "reactive_energy_pos_t4": "",
      "reactive_energy_pos_tl": "28.055",
      "reactive_energy_neg_t1": "",
      "reactive_energy_neg_t2": "",
      "reactive_energy_neg_t3": "",
      "reactive_energy_neg_t4": "",
      "reactive_energy_neg_tl": "",
      "reactive_energy_abs_t1": "0",
      "reactive_energy_abs_t2": "28.055",
      "reactive_energy_abs_t3": "",
      "reactive_energy_abs_t4": "",
      "reactive_energy_abs_tl": "28.055",
      "active_mdi_pos_t1": "0.024",
      "active_mdi_pos_t2": "0.025",
      "active_mdi_pos_t3": "",
      "active_mdi_pos_t4": "",
      "active_mdi_pos_tl": "0.025",
      "active_mdi_neg_t1": "",
      "active_mdi_neg_t2": "",
      "active_mdi_neg_t3": "",
      "active_mdi_neg_t4": "",
      "active_mdi_neg_tl": "",
      "active_mdi_abs_t1": "0.024",
      "active_mdi_abs_t2": "0.025",
      "active_mdi_abs_t3": "",
      "active_mdi_abs_t4": "",
        "active_mdi_abs_tl": "0.025",
        "cumulative_mdi_pos_t1": "0.024",
        "cumulative_mdi_pos_t2": "18.822",
        "cumulative_mdi_pos_t3": "",
        "cumulative_mdi_pos_t4": "",
        "cumulative_mdi_pos_tl": "18.822",
        "cumulative_mdi_neg_t1": "",
        "cumulative_mdi_neg_t2": "",
        "cumulative_mdi_neg_t3": "",
        "cumulative_mdi_neg_t4": "",
        "cumulative_mdi_neg_tl": "",
        "cumulative_mdi_abs_t1": "0.024",
        "cumulative_mdi_abs_t2": "18.822",
        "cumulative_mdi_abs_t3": "",
        "cumulative_mdi_abs_t4": "",
        "cumulative_mdi_abs_tl": "18.822",
        "current_phase_a": "",
        "current_phase_b": "",
        "current_phase_c": "",
        "voltage_phase_a": "",
        "voltage_phase_b": "",
        "voltage_phase_c": "",
        "active_pwr_pos_phase_a": "",
        "active_pwr_pos_phase_b": "",
        "active_pwr_pos_phase_c": "",
        "aggregate_active_pwr_pos": "0",
        "active_pwr_neg_phase_a": "",
        "active_pwr_neg_phase_b": "",
        "active_pwr_neg_phase_c": "",
        "aggregate_active_pwr_neg": "",
        "aggregate_active_pwr_abs": "0",
        "reactive_pwr_pos_phase_a": "",
        "reactive_pwr_pos_phase_b": "",
        "reactive_pwr_pos_phase_c": "",
        "aggregate_reactive_pwr_pos": "0",
        "reactive_pwr_neg_phase_a": "",
        "reactive_pwr_neg_phase_b": "",
        "reactive_pwr_neg_phase_c": "",
        "aggregate_reactive_pwr_neg": "",
        "aggregate_reactive_pwr_abs": "0",
        "average_pf": "",
        "mdc_read_datetime": "2020-06-15 09:21:07",
        "db_datetime": "2020-06-15 09:21:07",
        "is_sync ": "0"
```

```
    },
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "meter_datetime": "2020-06-15 09:30:00",
      "channel_id": 1,
      "interval": 15,
      "frequency": "49.81",
      "active_energy_pos_t1": "",
      "active_energy_pos_t2": "",
      "active_energy_pos_t3": "",
      "active_energy_pos_t4": "",
      "active_energy_pos_tl": "",
      "active_energy_neg_t1": "",
      "active_energy_neg_t2": "",
      "active_energy_neg_t3": "",
      "active_energy_neg_t4": "",
      "active_energy_neg_tl": "",
      "active_energy_abs_t1": "",
      "active_energy_abs_t2": "",
      "active_energy_abs_t3": "",
      "active_energy_abs_t4": "",
      "active_energy_abs_tl": "",
      "reactive_energy_pos_t1": "",
      "reactive_energy_pos_t2": "",
      "reactive_energy_pos_t3": "",
      "reactive_energy_pos_t4": "",
      "reactive_energy_pos_tl": "",
      "reactive_energy_neg_t1": "",
      "reactive_energy_neg_t2": "",
      "reactive_energy_neg_t3": "",
      "reactive_energy_neg_t4": "",
      "reactive_energy_neg_tl": "",
      "reactive_energy_abs_t1": "",
      "reactive_energy_abs_t2": "",
      "reactive_energy_abs_t3": "",
      "reactive_energy_abs_t4": "",
      "reactive_energy_abs_tl": "",
      "active_mdi_pos_t1": "",
      "active_mdi_pos_t2": "",
      "active_mdi_pos_t3": "",
      "active_mdi_pos_t4": "",
        "active_mdi_pos_tl": "",
        "active_mdi_neg_t1": "",
        "active_mdi_neg_t2": "",
        "active_mdi_neg_t3": "",
        "active_mdi_neg_t4": "",
        "active_mdi_neg_tl": "",
        "active_mdi_abs_t1": "",
        "active_mdi_abs_t2": "",
        "active_mdi_abs_t3": "",
        "active_mdi_abs_t4": "",
        "active_mdi_abs_tl": "",
        "cumulative_mdi_pos_t1": "",
        "cumulative_mdi_pos_t2": "",
        "cumulative_mdi_pos_t3": "",
        "cumulative_mdi_pos_t4": "",
        "cumulative_mdi_pos_tl": "",
        "cumulative_mdi_neg_t1": "",
        "cumulative_mdi_neg_t2": "",
        "cumulative_mdi_neg_t3": "",
        "cumulative_mdi_neg_t4": "",
        "cumulative_mdi_neg_tl": "",
        "cumulative_mdi_abs_t1": "",
        "cumulative_mdi_abs_t2": "",
        "cumulative_mdi_abs_t3": "",
        "cumulative_mdi_abs_t4": "",
        "cumulative_mdi_abs_tl": "",
        "current_phase_a": "0",
        "current_phase_b": "0.89",
        "current_phase_c": "1.71",
        "voltage_phase_a": "225.4",
        "voltage_phase_b": "222.1",
        "voltage_phase_c": "230.7",
        "active_pwr_pos_phase_a": "",
        "active_pwr_pos_phase_b": "",
        "active_pwr_pos_phase_c": "",
        "aggregate_active_pwr_pos": "0.49",
        "active_pwr_neg_phase_a": "",
        "active_pwr_neg_phase_b": "",
        "active_pwr_neg_phase_c": "",
        "aggregate_active_pwr_neg": "",
        "aggregate_active_pwr_abs": "0.49",
        "reactive_pwr_pos_phase_a": "",
        "reactive_pwr_pos_phase_b": "",
        "reactive_pwr_pos_phase_c": "",
      "aggregate_reactive_pwr_pos": "0.18",
      "reactive_pwr_neg_phase_a": "",
      "reactive_pwr_neg_phase_b": "",
      "reactive_pwr_neg_phase_c": "",
      "aggregate_reactive_pwr_neg": "",
      "aggregate_reactive_pwr_abs": "0.18",
      "average_pf": "0.84",
      "mdc_read_datetime": "2020-06-15 09:21:17",
      "db_datetime": "2020-06-15 09:21:17",
      "is_sync ": "0"
    }
  ],
  "message": "On Demand Load Profile from meter between provided dates"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
        <data>
                <element>
                        <active_energy_abs_t1>0.543</active_energy_abs_t1>
                        <active_energy_abs_t2>42.489</active_energy_abs_t2>
                        <active_energy_abs_t3 />
                        <active_energy_abs_t4 />
                        <active_energy_abs_tl>43.033</active_energy_abs_tl>
                        <active_energy_neg_t1 />
                        <active_energy_neg_t2 />
                        <active_energy_neg_t3 />
                        <active_energy_neg_t4 />
                        <active_energy_neg_tl />
                        <active_energy_pos_t1>0.543</active_energy_pos_t1>
                        <active_energy_pos_t2>42.489</active_energy_pos_t2>
                        <active_energy_pos_t3 />
                        <active_energy_pos_t4 />
                        <active_energy_pos_tl>43.033</active_energy_pos_tl>
                        <active_mdi_abs_t1>0.024</active_mdi_abs_t1>
                        <active_mdi_abs_t2>0.025</active_mdi_abs_t2>
                        <active_mdi_abs_t3 />
                        <active_mdi_abs_t4 />
                        <active_mdi_abs_tl>0.025</active_mdi_abs_tl>
                        <active_mdi_neg_t1 />
                        <active_mdi_neg_t2 />
                        <active_mdi_neg_t3 />
                        <active_mdi_neg_t4 />
```

```
                    <active_mdi_neg_tl />
                    <active_mdi_pos_t1>0.024</active_mdi_pos_t1>
                    <active_mdi_pos_t2>0.025</active_mdi_pos_t2>
                    <active_mdi_pos_t3 />
                    <active_mdi_pos_t4 />
                    <active_mdi_pos_tl>0.025</active_mdi_pos_tl>
                    <active_pwr_neg_phase_a />
                    <active_pwr_neg_phase_b />
                    <active_pwr_neg_phase_c />
                    <active_pwr_pos_phase_a />
                    <active_pwr_pos_phase_b />
                    <active_pwr_pos_phase_c />
                    <aggregate_active_pwr_abs>0</aggregate_active_pwr_abs>
                    <aggregate_active_pwr_neg />
                    <aggregate_active_pwr_pos>0</aggregate_active_pwr_pos>
                    <aggregate_reactive_pwr_abs>0</aggregate_reactive_pwr_abs>
                    <aggregate_reactive_pwr_neg />
                    <aggregate_reactive_pwr_pos>0</aggregate_reactive_pwr_pos>
                    <average_pf />
                    <channel_id>1</channel_id>
                    <cumulative_mdi_abs_t1>0.024</cumulative_mdi_abs_t1>
                    <cumulative_mdi_abs_t2>18.822</cumulative_mdi_abs_t2>
                    <cumulative_mdi_abs_t3 />
                    <cumulative_mdi_abs_t4 />
                    <cumulative_mdi_abs_tl>18.822</cumulative_mdi_abs_tl>
                    <cumulative_mdi_neg_t1 />
                    <cumulative_mdi_neg_t2 />
                    <cumulative_mdi_neg_t3 />
                    <cumulative_mdi_neg_t4 />
                    <cumulative_mdi_neg_tl />
                    <cumulative_mdi_pos_t1>0.024</cumulative_mdi_pos_t1>
                    <cumulative_mdi_pos_t2>18.822</cumulative_mdi_pos_t2>
                    <cumulative_mdi_pos_t3 />
                    <cumulative_mdi_pos_t4 />
                    <cumulative_mdi_pos_tl>18.822</cumulative_mdi_pos_tl>
                    <current_phase_a />
                    <current_phase_b />
                    <current_phase_c />
                    <db_datetime>2020-06-15 09:21:07</db_datetime>
                    <frequency />
                    <global_device_id>m98562412</global_device_id>
                    <indv_status>1</indv_status>
                    <interval>15</interval>
                    <mdc_read_datetime>2020-06-15 09:21:07</mdc_read_datetime>
```

```
                <meter_datetime>2020-06-15 09:15:00</meter_datetime>
                <msn>4098562412</msn>
                <reactive_energy_abs_t1>0</reactive_energy_abs_t1>
                <reactive_energy_abs_t2>28.055</reactive_energy_abs_t2>
                <reactive_energy_abs_t3 />
                <reactive_energy_abs_t4 />
                <reactive_energy_abs_tl>28.055</reactive_energy_abs_tl>
                <reactive_energy_neg_t1 />
                <reactive_energy_neg_t2 />
                <reactive_energy_neg_t3 />
                <reactive_energy_neg_t4 />
                <reactive_energy_neg_tl />
                <reactive_energy_pos_t1>0</reactive_energy_pos_t1>
                <reactive_energy_pos_t2>28.055</reactive_energy_pos_t2>
                <reactive_energy_pos_t3 />
                <reactive_energy_pos_t4 />
                <reactive_energy_pos_tl>28.055</reactive_energy_pos_tl>
                <reactive_pwr_neg_phase_a />
                <reactive_pwr_neg_phase_b />
                <reactive_pwr_neg_phase_c />
                <reactive_pwr_pos_phase_a />
                <reactive_pwr_pos_phase_b />
                <reactive_pwr_pos_phase_c />
                <voltage_phase_a />
                <voltage_phase_b />
                <voltage_phase_c />
                <is_sync>0</is_sync>
        </element>
        <element>
                <active_energy_abs_t1 />
                <active_energy_abs_t2 />
                <active_energy_abs_t3 />
                <active_energy_abs_t4 />
                <active_energy_abs_tl />
                <active_energy_neg_t1 />
                <active_energy_neg_t2 />
                <active_energy_neg_t3 />
                <active_energy_neg_t4 />
                <active_energy_neg_tl />
                <active_energy_pos_t1 />
                <active_energy_pos_t2 />
                <active_energy_pos_t3 />
                <active_energy_pos_t4 />
                <active_energy_pos_tl />
                <active_mdi_abs_t1 />
                <active_mdi_abs_t2 />
                <active_mdi_abs_t3 />
                <active_mdi_abs_t4 />
                <active_mdi_abs_tl />
                <active_mdi_neg_t1 />
                <active_mdi_neg_t2 />
                <active_mdi_neg_t3 />
                <active_mdi_neg_t4 />
                <active_mdi_neg_tl />
                <active_mdi_pos_t1 />
                <active_mdi_pos_t2 />
                <active_mdi_pos_t3 />
                <active_mdi_pos_t4 />
                <active_mdi_pos_tl />
                <active_pwr_neg_phase_a />
                <active_pwr_neg_phase_b />
                <active_pwr_neg_phase_c />
                <active_pwr_pos_phase_a />
                <active_pwr_pos_phase_b />
                <active_pwr_pos_phase_c />
                <aggregate_active_pwr_abs>0.49</aggregate_active_pwr_abs>
                <aggregate_active_pwr_neg />
                <aggregate_active_pwr_pos>0.49</aggregate_active_pwr_pos>
                <aggregate_reactive_pwr_abs>0.18</aggregate_reactive_pwr_abs>
                <aggregate_reactive_pwr_neg />
                <aggregate_reactive_pwr_pos>0.18</aggregate_reactive_pwr_pos>
                <average_pf>0.84</average_pf>
                <channel_id>1</channel_id>
                <cumulative_mdi_abs_t1 />
                <cumulative_mdi_abs_t2 />
                <cumulative_mdi_abs_t3 />
                <cumulative_mdi_abs_t4 />
                <cumulative_mdi_abs_tl />
                <cumulative_mdi_neg_t1 />
                <cumulative_mdi_neg_t2 />
                <cumulative_mdi_neg_t3 />
                <cumulative_mdi_neg_t4 />
                <cumulative_mdi_neg_tl />
                <cumulative_mdi_pos_t1 />
                <cumulative_mdi_pos_t2 />
                <cumulative_mdi_pos_t3 />
                <cumulative_mdi_pos_t4 />
                <cumulative_mdi_pos_tl />
```

```
            <current_phase_a>0</current_phase_a>
            <current_phase_b>0.89</current_phase_b>
            <current_phase_c>1.71</current_phase_c>
            <db_datetime>2020-06-15 09:21:17</db_datetime>
            <frequency>49.81</frequency>
            <global_device_id>m98562412</global_device_id>
            <indv_status>1</indv_status>
            <interval>15</interval>
            <mdc_read_datetime>2020-06-15 09:21:17</mdc_read_datetime>
            <meter_datetime>2020-06-15 09:30:00</meter_datetime>
            <msn>4098562412</msn>
            <reactive_energy_abs_t1 />
            <reactive_energy_abs_t2 />
            <reactive_energy_abs_t3 />
            <reactive_energy_abs_t4 />
            <reactive_energy_abs_tl />
            <reactive_energy_neg_t1 />
            <reactive_energy_neg_t2 />
            <reactive_energy_neg_t3 />
            <reactive_energy_neg_t4 />
            <reactive_energy_neg_tl />
            <reactive_energy_pos_t1 />
            <reactive_energy_pos_t2 />
            <reactive_energy_pos_t3 />
            <reactive_energy_pos_t4 />
            <reactive_energy_pos_tl />
            <reactive_pwr_neg_phase_a />
            <reactive_pwr_neg_phase_b />
            <reactive_pwr_neg_phase_c />
            <reactive_pwr_pos_phase_a />
            <reactive_pwr_pos_phase_b />
            <reactive_pwr_pos_phase_c />
            <voltage_phase_a>225.4</voltage_phase_a>
            <voltage_phase_b>222.1</voltage_phase_b>
            <voltage_phase_c>230.7</voltage_phase_c>
            <is_sync>0</is_sync>
        </element>
    </data>
    <message>On Demand Load Profile from meter between provided dates</message>
    <status>1</status>
    <transactionid>3fdcc1a20f284fe395436b9e5ef49fca</transactionid>
</root>
```

4.16 ON_DEMAND_PARAMETER_READ (MTST)

Request:

Method: POST

URL: https://host:port/on_demand_parameter_read

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	"m98562412"
type	"MTST"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": "ARbvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "mtst_datetime": "2018-10-16 11:00:00",
      "mtst_meter_activation_status": "1"
    }
  ],
  "message": "Meter Current Status"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
       <status>1</status>
       <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
       <data>
              <global_device_id>m98562412</global_device_id>
              <msn>4098562412</msn>
              <mtst_datetime>2018-10-16 11:00:00</mtst_datetime>
              <mtst_meter_activation_status>1</mtst_meter_activation_status>
       </data>
       <message>Meter Current Status</message>
</root>
```

4.17 ON_DEMAND_PARAMETER_READ (MSIM)

Request:

```
Method: POST
URL: https://host:port/on_demand_parameter_read
```

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3aRi
privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	"m98562412"
type	"MSIM"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": "ARbvfg675d5hg85fhjjfved47f5d3aRi",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "msim_id": "8991006090527906588"
    }
  ],
  "message": "Meter SIM ID is provided"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
      <status>1</status>
      <transactionid>ARbvfg675d5hg85fhjjfved47f5d3aRi</transactionid>
      <data>
            <global_device_id>m98562412</global_device_id>
            <msn>4098562412</msn>
            <msim_id>8991006090527906588</msim_id>
      </data>
      <message> SIM ID is provided </message>
</root>
```

4.18 APMS_TRIPPING_EVENTS

Request:

Method: POST

URL: https://host:port/apms_tripping_events

Headers:

Header Name	Header Value
transactionid	ARbvfg675d5hg85fhjjfved47f5d3O
Privatekey	7h6g5d4chyhdxg875576v5f5gv8b7bv

Body [Form Data (www-url-form-encoded)]:

Parameters Key	Parameters Value
global_device_id	["m98562412", "m97524158"]
request_datetime	"2020-10-15 13:00:00"
type	"OVFC"
critical_event_threshold_limit	"270"
critical_event_log_time	"180"
tripping_event_threshold_limit	"300"
tripping_event_log_time	"240"
enable_tripping	"1"

Response:

JSON:

```
{
  "status": 1,
  "transactionid": "ARbvfg675d5hg85fhjjfved47f5d3O",
  "data": [
    {
      "global_device_id": "m98562412",
      "msn": "4098562412",
      "indv_status": "1",
      "remarks": "Over Voltage Function will be applied"
    },
    {
      "global_device_id": "m97524158",
      "msn": "4097524158",
      "indv_status": "1",
      "remarks": "Over Voltage Function will be applied"
    }
  ],
  "message": "Over Voltage Function will be applied on devices having individual status
as 1"
}
```

XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<root>
       <status>1</status>
       <transactionid>ARbvfg675d5hg85fhjjfved47f5d3O</transactionid>
       <data>
              <element>
                     <global_device_id>m98562412</global_device_id>
                     <msn>4098562412</msn>
                     <indv_status>1</indv_status>
                     <remarks>Over Voltage Function will be applied</remarks>
              </element>
              <element>
                     <global_device_id>m97524158</global_device_id>
                     <msn>4097524158</msn>
                     <indv_status>1</indv_status>
                     <remarks>Over Voltage Function will be applied</remarks>
              </element>
       </data>
       <message>Over Voltage Function will be applied on devices having individual
status as 1</message>
</root>
```