

*NOTE: This draft, dated 19 April 2021, prepared by the Preparing Activity CR4, has not been approved and is subject to modification. DO NOT USE PRIOR TO APPROVAL. (Project Number: TBD).*

**METRIC**  
**MIL-STD-XXX**  
**DRAFT**

**DEPARTMENT OF DEFENSE**  
**INTERFACE STANDARD**  
**TACTICAL MICROGRID**  
**COMMUNICATIONS AND CONTROL**



APPROVED FOR PUBLIC RELEASE

This document dated 19 April 2021 was Approved for Public Release by US Army CCDC C5ISR Center. Item Number A229.

AMSC N/A

FSC 6115

FOREWORD

- a. This standard is [not yet] approved for use by all Departments and Agencies of the Department of Defense (DoD)
- b. This standard establishes criteria to enable the interoperability of hardware and software necessary to operate a tactical microgrid on the battlefield with respect to the design, intelligent control, stability and performance, security, safety of personnel, and the protection of the tactical microgrid system and equipment.
- c. This standard defines the tactical microgrid architecture using open standards to support a modular, highly cohesive system structure in order to leverage the collaborative innovation of industry, academia, and government participants along with stakeholders.
- d. This document is intended to be free of Essential Patent Claims. Although development of this standard was made with the active participation of the Private Sector, non-government contributors have certified that they have declared all essential patent claims found in this standard.
- e. Comments, suggestions, or questions on this document should be addressed to (TBD)

## Table of Contents

45		
46	1. SCOPE.....	1
47	1.1. Scope.....	1
48	1.2. Purpose.....	1
49	1.3. Application.....	1
50	1.4. Classification.....	1
51	1.5. Limitations.....	1
52	2. APPLICABLE DOCUMENTS.....	1
53	2.1. General.....	1
54	2.2. Government documents.....	2
55	2.2.1. Specifications, standards, and handbooks.....	2
56	2.2.2. Other Government documents, drawings, and publications.....	2
57	2.3. Non-Government Publications.....	3
58	2.4. Order of Precedence.....	5
59	3. DEFINITIONS.....	5
60	3.1. Acronyms.....	5
61	3.2. Definitions.....	7
62	3.2.1. Automatic Voltage Regulator (AVR).....	7
63	3.2.2. Backwards compatibility.....	7
64	3.2.3. Command.....	7
65	3.2.4. Configuration.....	7
66	3.2.5. Device.....	7
67	3.2.6. Distribution.....	7
68	3.2.7. Essential patent.....	7
69	3.2.8. Exciter.....	8
70	3.2.9. Fingerprint.....	8
71	3.2.10. Generating source.....	8
72	3.2.11. Grid.....	8
73	3.2.12. Interconnection.....	8
74	3.2.13. Interface.....	8
75	3.2.14. Interoperability.....	8
76	3.2.15. Inverter.....	8
77	3.2.16. Legacy equipment.....	8
78	3.2.17. Load.....	8

MIL-STD-XXX  
DRAFT 19 Apr 2021

79	3.2.18.	Load shedding. ....	8
80	3.2.19.	Operator.....	8
81	3.2.20.	Power Device. ....	8
82	3.2.21.	Open standard.....	8
83	3.2.22.	Performance. ....	8
84	3.2.23.	Point of Connection (POC). ....	8
85	3.2.24.	Ring-bus off-line. ....	8
86	3.2.25.	Setting. ....	9
87	3.2.26.	Source (SRC).....	9
88	3.2.27.	Storage device. ....	9
89	3.2.28.	Tactical microgrid (TM). ....	9
90	3.2.29.	Tactical Microgrid Controller (TMC). ....	9
91	3.2.30.	User. ....	9
92	3.2.31.	War-fighter. ....	9
93	4.	GENERAL REQUIREMENTS.....	9
94	4.1.	General. ....	9
95	4.2.	Types of Tactical Microgrid Equipment. ....	9
96	4.2.1.	Type I - Power Generation Sources. ....	9
97	4.2.2.	Type II – Energy Storage Systems.....	9
98	4.2.3.	Type III – Feeder Systems. ....	10
99	4.2.4.	Type IV – Distribution Systems. ....	10
100	4.2.5.	Type V – Control Systems.....	10
101	4.2.6.	Type VI – Loads. ....	10
102	4.2.7.	Type VII – Power Conversion. ....	10
103	4.3.	Interoperability.....	10
104	4.4.	Standalone Use.....	11
105	4.4.1.	Type I Power Generation Sources and Type II Energy Storage Equipment.....	11
106	4.4.2.	Type III Feeder and Type IV Distribution Systems Equipment. ....	11
107	4.5.	Degraded Operation. ....	11
108	4.6.	Monitoring and Control.....	11
109	4.7.	Cybersecurity. ....	11
110	4.7.1.	Risk Management Framework (RMF).....	11
111	4.7.2.	Operational Resilience. ....	11
112	4.7.3.	Access to firmware/software.....	12

113	4.8.	Tactical Microgrid Architecture.....	12
114	4.8.1.	Common Microgrid Control System. ....	13
115	4.8.2.	Common Microgrid Monitoring System. ....	13
116	4.8.3.	Levels of Control. ....	13
117	4.8.4.	Tactical Microgrid Control (TMC) Redundancy.....	13
118	4.8.5.	Communications Architecture. ....	13
119	4.9.	Data Model.....	14
120	4.9.1.	Device Identity.....	14
121	4.9.2.	Device Roles. ....	15
122	4.9.3.	Platforms. ....	15
123	4.9.4.	Data flows. ....	16
124	4.9.5.	Publishers and Subscribers. ....	16
125	4.9.6.	Topic and Message Flow. ....	16
126	4.9.7.	Data Types. ....	17
127	4.9.8.	Data Flow Patterns. ....	17
128	4.9.9.	Instances.....	18
129	4.9.10.	Timing.....	18
130	5.	DETAILED REQUIREMENTS.....	19
131	5.1.	Safety.....	19
132	5.2.	Human Engineering.....	20
133	5.3.	TMG Equipment Roles and Responsibilities .....	20
134	5.3.1.	Type I – Power Sources .....	20
135	5.3.2.	Type II – Energy Storage.....	21
136	5.3.3.	Type III – Feeder Systems .....	22
137	5.3.4.	Type IV – Distribution Systems .....	24
138	5.3.5.	Type V – Control Systems.....	25
139	5.3.6.	Type VI – Loads .....	27
140	5.3.7.	Type VII – Power Conversion .....	28
141	5.4.	Communications Interface .....	29
142	5.4.1.	Applications Layer.....	29
143	5.4.2.	Data Objects Layer. ....	30
144	5.4.3.	Middleware Layer.....	30
145	5.4.4.	Transport Layer.....	30
146	5.4.5.	Network Layer.....	30

147       5.4.6.   Physical Layer..... 30

148   6.   NOTES ..... 30

149       6.1.   Intended Use..... 30

150       6.2.   Acquisition requirements. .... 30

151       6.3.   Tailoring guidance..... 30

152       6.4.   Subject term (key word) Listing..... 30

153

154

1. SCOPE.

1.1. Scope. This standard establishes the communication and control interface requirements for Tactical Microgrids (TMs) to enable TMG components to operate as a single entity to provide electricity in a tactical environment. This standard defines the component interfaces of the hardware and software necessary to provide intelligent control, power quality, electrical stability, electrical performance, safety of personnel, security, and protection.

A tactical microgrid is a warfighter-operated and maintained power system consisting of a mobile, flexible group of interconnected power generation sources, distribution, energy storage, and load devices that act as a single, controllable system to provide electricity on the battlefield.

1.2. Purpose. The purpose of this DOD standard is to define an open communication and control architecture for TMs to facilitate:

- a. Interoperability between manufacturers of tactical microgrid components
- b. Safety of personnel and equipment
- c. Backwards compatibility with legacy equipment wherever possible
- d. Security of the tactical microgrid communication network and controllers
- e. Reduction of procurement, operational, sustainment, and maintenance costs

1.3. Application. This standard is applicable to tactical microgrid systems, subsystems, and individual equipment. While this standard provides interoperability and design criteria guidance, it is not intended to limit innovation in the design or selection of specific hardware, software, materials, and processes.

1.4. Classification. The Tactical Microgrids are made up of the following types of equipment:

**Table 1-1 Types of Equipment in a Tactical Microgrid**

Types of Equipment in a Tactical Microgrid	
Type I	Power Generation Sources (e.g. Engine-Generator sets, Solar, Wind, etc.)
Type II	Energy Storage Systems, (e.g. batteries, capacitors, etc.)
Type III	Feeder Systems (e.g. 3 phase inputs with 3 phase outputs)
Type IV	Distribution Systems, (e.g. 1 or 3 phase inputs with 1 phase outputs)
Type V	Control Systems w/ Communications and cabling
Type VI	Loads (e.g. Improved Environmental Control Units)
Type VII	Power Conversion

1.5. Limitations. This document does not define test procedures that verify compliance with this standard. This document is not to be referenced as the sole source for cybersecurity requirements.

2. APPLICABLE DOCUMENTS

2.1. General. The documents listed in this section are specified in sections 3, 4, or 5 of this standard. This section does not include documents cited in other sections of this standard or

MIL-STD-XXX  
DRAFT 19 Apr 2021

recommended for additional information or as examples. While every effort has been made to ensure the completeness of this list, document users are cautioned that they must meet all specified requirements of documents cited in sections 3, 4, or 5 of this standard, whether or not they are listed.

2.2. Government documents.

2.2.1. Specifications, standards, and handbooks. The following specifications, standards, and handbooks form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those cited in the solicitation or contract.

DEPARTMENT OF DEFENSE STANDARDS

MIL-STD-464	Electromagnetic Environmental Effects, Requirements For Systems
MIL-HDBK-759	Human Engineering Design Guidelines
MIL-STD-633	Standard Family of Mobile Electric Power Generating Sources General Description Information and Characteristic Data
MIL-STD-810	Test Method Standard: Environmental Engineering Considerations And Laboratory Tests
MIL-STD-882E	System Safety
MIL-STD-1332	Definitions of Tactical, Prime, Precise, and Utility Terminologies for Classification of the DOD Mobile Electric Power Engine Generator Set Family
MIL-STD-1472	Human Engineering
MIL-STD-1651	Insert Arrangements for SAE-AS50151, MIL-DTL-22992 (Classes C, J, and R), MIL-DTL-83723 (Series II) and SAE- AS95234 Circular Electrical Connectors
MIL-DTL-22992	Connectors, Plugs and Receptacles, Electrical, Waterproof, Quick Disconnect, Heavy Duty Type, General Specification For
MIL-DTL-53126 A SUPP 1	Power Distribution, Illumination Systems, Electrical General Specification for
MIL-L-85762A	Lighting, Aircraft, Interior, Night Vision Imaging System (NVIS) Compatible

(Copies of these documents are available online at <http://quicksearch.dla.mil>.)

2.2.2. Other Government documents, drawings, and publications. The following other Government documents, drawings, and publications form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those cited in the solicitation or contract.

AR385-10	The Army Safety Program
CECOMTR-96-2	Earth Grounding Pamphlet A Guide to Proper Earth Grounding Methods and Procedures for use with



MIL-STD-XXX  
DRAFT 19 Apr 2021

	Tactical System
DODI 8500.01	Cybersecurity
NIST SP 800-39	Managing Information Security Risk: Organization, Mission, and Information System View
NIST SP 800-53 Rev. 5	Security and Privacy Controls for Information Systems and Organizations
SP 800-57 Part 1 Rev. 5	Recommendation for Key Management: Part 1 – General
NIST SP 800-82 Rev. 2	Guide to Industrial Control Systems (ICS) Security
NIST SP 800-193	Platform Firmware Resilience Guidelines
Technical Manual No. 3-34.46/Marine Corps Reference Publication 3-17.7K	Theater of Operations Electrical Systems

(Copies of these documents are available online.)

2.3. Non-Government Publications. The following documents form a part of this document to the extent specified herein. Unless otherwise specified, the issues of these documents are those cited in the solicitation or contract.

AMERICAN NATIONAL STANDARDS INSTITUTE (ANSI)

ANSI/NEMA C84.1-2006	American National Standard for Electric Power Systems and Equipment - Voltage Ratings (60 Hertz)
----------------------	--

(Copies can be obtained at <https://ansi.org/>)

AMERICAN SOCIETY FOR TESTING MATERIALS

ASTM F1166 - 07(2013)	Standard Practice for Human Engineering Design for Marine Systems, Equipment, and Facilities
-----------------------	--

(Copies can be obtained at <https://www.astm.org/>)

INSTITUTE OF ELECTRICAL AND ELECTRONIC ENGINEERS (IEEE)

IEEE STD C62.41.2-2002	Recommended Practice on Characterization of Surges in Low-Voltage (1000 V and less) AC Power Circuits
IEEE-519-1992	Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems
IEEE-1547-2003	Standard for Interconnecting Distributed Resources with Electric Power Systems

MIL-STD-XXX  
DRAFT 19 Apr 2021

IEEE-260.1	Standard Letter Symbols for Units of Measurement
IEEE Std.100	Dictionary of Electrical and Electronic Terms
IEEE 754-2008	Standard for Floating-Point Arithmetic
IEEE 802.1X-2010	Standard for Local and metropolitan area networks-- Port-Based Network Access Control
IEEE STD 945-1984	Recommended Practice for Preferred Metric Units for Use in Electrical and Electronics Science and Technology.
IEEE 1003.1-2017	Standard for Information Technology - Portable Operating System Interface (POSIX(TM))
IEEE-1588-2008	Precision Time Protocol

(Copies can be obtained at <https://www.ieee.org/>)

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION

ISO/IEC 14496-22:2019	Information technology — Coding of audio-visual objects — Part 22: Open Font Format
-----------------------	--

(Copies can be obtained at <https://www.iso.org/>)

INTERNATIONAL ELECTROTECHNICAL COMMISSION (IEC)

61000-3-2	Electromagnetic compatibility (EMC) - Part 3-2: Limits - Limits for harmonic current emissions (equipment input current $\leq 16$ A per phase)
61131-3	Programming Industrial Automation Systems : Concepts and Programming Languages, Requirements for Programming Systems, Decision-Making Aids
61850	Communication Networks and Systems in Substations

(Copies can be obtained at <https://www.iec.ch>)

INTERNATIONAL TELECOMMUNICATION UNION TELECOMMUNICATION  
STANDARDIZATION SECTOR (ITU-T)

Rec I.371	Traffic control and congestion control in B-ISDN
-----------	--

(Copies can be obtained at <https://www.itu.int>)

227

228 NATIONAL FIRE PROTECTION ASSOCIATION (NFPA)

NFPA-70                      National Electrical Code

NFPA-70E                      Standard for Electrical Safety in the Workplace

NFPA 704      Standard System for the Identification of the Hazards of  
Materials for Emergency Response

229 (Copies can be obtained at <https://nfpa.org/>)

230

231 OBJECT MANAGEMENT GROUP

## 2.4.1 Unified Modeling Language (OMG UML), Infrastructure Specification

232 (Copies can be obtained at <https://www.omg.org/>)

233

234 SAE INTERNATIONAL

SAE J1939-73      Diagnostics Application Layer

235

236 (Copies can be obtained at <https://www.sae.org/>)

237

238 2.4. Order of Precedence. Unless otherwise noted herein or in the contract, in the event of a  
239 conflict between the text of this document and the references cited herein, the text of this  
240 document takes precedence. Nothing in this document, however, supersedes applicable laws and  
241 regulations unless a specific exemption has been obtained.

242 3. DEFINITIONS.

243 3.1. Acronyms.

244

ACRONYM	DEFINITION
AC	Alternating Current
AN	Application Note
ANSI/NEMA	American National Standards Institute / National Electrical Manufacturers Association
API	Application Programming Interface
AR	Army Regulation
DC	Direct Current
DDS	Data Distribution Service

MIL-STD-XXX  
DRAFT 19 Apr 2021

ACRONYM	DEFINITION
<b>DER</b>	Distributed Energy Resource
<b>DOD</b>	Department of Defense
<b>DODI</b>	Department of Defense Instruction
<b>ECU</b>	Environmental Control Unit
<b>EIO</b>	Energy Informed Operations
<b>GCRA</b>	Generic Cell Rate Algorithm
<b>IEC</b>	International Electrotechnical Commission
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>IETF</b>	Internet Engineering Task Force
<b>IoT</b>	Internet of Things
<b>IP</b>	Internet Protocol
<b>IPv6</b>	Internet Protocol Version 6
<b>ISO</b>	International Organization for Standardization
<b>IT</b>	Information Technology
<b>JSON</b>	JavaScript Object Notation
<b>MBS</b>	Maximum Burst Size
<b>MC</b>	Microgrid Controller
<b>MIL-STD-XXX</b>	Defense Standard
<b>MMC</b>	Master Microgrid Controller
<b>NEC</b>	National Electrical Code
<b>NFPA</b>	National Fire Protection Association
<b>NIST</b>	National Institute of Standards and Technology
<b>NSN</b>	National Stock Number
<b>OMG</b>	Object Management Group
<b>OpenFMB™</b>	Open Field Message Bus
<b>OSI</b>	Open Systems Interconnection
<b>OT</b>	Operational Technology
<b>PCR</b>	Peak Cell Rate
<b>PTP</b>	Precision Time Protocol
<b>QoS</b>	Quality of Service
<b>RFC</b>	Request For Comment
<b>RMF</b>	Risk Management Framework

MIL-STD-XXX  
DRAFT 19 Apr 2021

ACRONYM	DEFINITION
<b>RMS</b>	Root Mean Squared
<b>ROM</b>	Read Only Memory
<b>RoT</b>	Root of Trust
<b>SDLC</b>	System Development Lifecycle
<b>SP</b>	Special Publication
<b>TM</b>	Technical Manual
<b>TLS</b>	Transport Layer Security
<b>TM</b>	Tactical Microgrid
<b>TMC</b>	Tactical Microgrid Controller
<b>TMS</b>	Tactical Microgrid Standard
<b>TPM</b>	Trusted Platform Module
<b>UDP</b>	User Datagram Protocol
<b>UI</b>	User Interface
<b>VAC</b>	Volts AC
<b>XML</b>	Extensible Markup Language

245

246 3.2. Definitions

247 3.2.1. Automatic Voltage Regulator (AVR). A device that automatically adjusts the  
248 synchronous generator's excitation to regulate the voltage output to either a predetermined level  
249 or based on a stable reference, for example, the sinusoidal output Root Mean Squared (RMS)  
250 voltage.

251 3.2.2. Backwards compatibility. The ability of a system or component to operate with  
252 previously-fielded systems or components of the system.

253 3.2.3. Command. An action with an immediate effect that the system must attempt to execute.  
254 A failure results in an error.

255 3.2.4. Configuration. A collection of devices and connections that complete a TMG system.

256 3.2.5. Device. A standalone component that is part of the TMG system, including generating  
257 sources, Tactical Microgrid Controller (TMC), distribution, storage, and loads.

258 3.2.6. Distribution. All the equipment necessary to distribute power from the sources to the  
259 loads, including protective devices, boxes, and cabling. Distribution equipment include both  
260 Type III Feeder Systems and Type IV Distribution Systems.

261 3.2.7. Essential patent. A patent required to be obtained in order to produce a standard  
262 compliant product.

MIL-STD-XXX  
DRAFT 19 Apr 2021

- 263 3.2.8. Exciter. A device that provides electric current to produce the rotor's magnetic field in a  
264 generator, for example, a battery or small generator.
- 265 3.2.9. Fingerprint. A data type that assigns a unique identity to each TMS device, associated  
266 with a single specific device role.
- 267 3.2.10. Generating source. Equipment which consumes fuel to provide power to the TM,  
268 typically synchronous diesel generators.
- 269 3.2.11. Grid. General term for a system of components connected together. In this standard, grid  
270 refers to either a Utility Power Grid or a Microgrid.
- 271 3.2.12. Interconnection. Process of connecting military generation and distribution equipment to  
272 a powered or un-powered Power System.
- 273 3.2.13. Interface. The point where two systems meet and interact.
- 274 3.2.14. Interoperability. The ability to act coherently, effectively, and efficiently to achieve  
275 tactical, operational, and strategic objectives. The devices must communicate, provide, and use  
276 services and information between systems, regardless of manufacturer or version, without the  
277 need for operator intervention.
- 278 3.2.15. Inverter. Device that changes a direct current power source into an alternating current or  
279 voltage source.
- 280 3.2.16. Legacy equipment. In this standard, refers to any and all military generation,  
281 distribution, storage, and load equipment deployed before the release of this standard.
- 282 3.2.17. Load. A user device that is powered by the TM.
- 283 3.2.18. Load shedding. Distribution disconnects one or more loads when the generators are  
284 overloaded until the overloaded condition is removed.
- 285 3.2.19. Operator. A war-fighter tasked to install, run, and maintain the tactical microgrid.
- 286 3.2.20. Power Device. A standalone component that is part of the TMG system, including  
287 generating sources, distribution, storage, and loads. A power device does not include the TMC.
- 288 3.2.21. Open standard. Standards made available to the general public that are developed (or  
289 approved) and maintained via a collaborative and consensus driven process.
- 290 3.2.22. Performance. The capability of a device when observed under particular conditions.
- 291 3.2.23. Point of Connection (POC). The power connection between any power device and the  
292 TM.
- 293 3.2.24. Ring-bus off-line. A ring-bus microgrid configuration where one bus segment is  
294 connected but unpowered and kept as a spare. In the event a bus segment is cut, the spare bus  
295 segment would be energized.

- 296 3.2.25. Setting. Modifies either the software or hardware behavior of the device.
- 297 3.2.26. Source (SRC). A device that provides energy to the microgrid, including both generating  
298 and renewable sources.
- 299 3.2.27. Storage device. A system that saves and stores energy to be used at a later time.
- 300 3.2.28. Tactical microgrid (TM). A tactical microgrid is a warfighter-operated and maintained  
301 power system consisting of a mobile, flexible group of interconnected power generation sources,  
302 distribution, storage, and load devices that act as a single, controllable system to provide  
303 electricity on the battlefield. Intended to be self-contained and readily deployable, the TMG may  
304 utilize alternative energy resources, energy storage, and be capable of interfacing with other  
305 grids.
- 306 3.2.29. Tactical Microgrid Controller (TMC). A device, which an operator can access to set up,  
307 run, or maintain the system, which can communicate with other devices and automatically  
308 change their settings.
- 309 3.2.30. User. A war-fighter allowed to view the current state and make minor changes to the  
310 tactical microgrid.
- 311 3.2.31. War-fighter. Military personnel with primary skills in combat that may have limited staff  
312 with generator experience. War-fighters can be both users and operators.

#### 313 4. GENERAL REQUIREMENTS

- 314 4.1. General. The Tactical Microgrid (TM) is the essential tool needed to reduce fuel  
315 consumption while producing and distributing mission critical electricity on the battlefield. The  
316 TMG must have the ability to produce and distribute electricity in a tactical environment  
317 effectively, and efficiently to enable the Warfighter to meet mission needs. This section provides  
318 general requirements for tactical microgrid interface and design to meet this need.
- 319 4.2. Types of Tactical Microgrid Equipment. There are basic components necessary for a  
320 tactical microgrid. Each component or end item has specific functions to enable the microgrid to  
321 function. These components are classified in para 1.4 and Table 1-1. The basic functions and  
322 requirements are summarized in the following paragraphs.
- 323 4.2.1. Type I - Power Generation Sources. Type I - Power Generation Sources are end items  
324 designed to produce electricity for the TMG. Each item has the capability to operate as an  
325 individual power source or can be synchronized to provide power as a TMG. As an individual  
326 power source, they can be used to provide power to a single load such as a shelter or radar  
327 system. As an element of the TMG, multiple power sources can be synchronized and controlled  
328 to turn on and off based on the load on the TMG. Type I equipment includes, but is not limited  
329 to, typical diesel engine driven generator sets, alternative energy power sources such as solar,  
330 wind, fuel cells, etc. and hybrids.
- 331 4.2.2. Type II – Energy Storage Systems. Type II – Energy Storage Systems provide stability  
332 to the TMG during transitions and peak shaving. Energy storage systems shall provide power  
333 during the period when additional power is necessary but the power source has not been added to  
334 the grid. In addition, energy storage systems shall function as an uninterruptable power supply

MIL-STD-XXX  
DRAFT 19 Apr 2021

(UPS) or as a hybrid when paired with a power source for individual loads. Energy storage systems shall include energy conversion to enable individual operation or connection to the TMG. Individual operation means not functioning as part of the TMG but can function when connected to a load.

4.2.3. Type III – Feeder Systems. Type III – Feeder Systems are a type of distribution equipment which provide the interface or connection points to the TMG for the other types TMG equipment. Feeder systems define the TMG grid itself. They can be added or subtracted to define the electrical bus necessary to provide the power to the loads directly or through the distribution systems. They can also provide energy conversion of power sources into the TMG or loads out of the TMG. Feeder systems shall include the ability to manage the circuits through independent control or through the use of Type V – Control Systems equipment. Feeder systems shall have metering and monitoring systems incorporated to enable the Warfighter to manually control the system and to automatically control the system using Type V Control Systems

4.2.4. Type IV – Distribution Systems. Type IV – Distribution Systems provide the power from the TMGs through the Type III – Feeder Systems to loads which require individual circuits for standard lighting and standard duplex receptacles. Type II equipment can be individual items or integrated into various shelters. Distribution systems shall include the ability to manage the circuits through independent control or through the use of Type V – Control Systems equipment. Distribution systems shall have metering and monitoring systems incorporated to enable the Warfighter to manually control the system and to automatically control the system using the Type V Control Systems.

4.2.5. Type V – Control Systems. Type V – Control Systems provide power management for the TMG. Control systems monitor various parameters of the TMG, such as current, and add or subtract power generation sources and/or add or subtract load to ensure power is available. Control systems shall monitor and protect the system in the event of overload or faults. Control systems shall reside as an individual item or be built into the various types of TMG equipment.

4.2.6. Type VI – Loads. Type VI – Loads consume the power provided by the TMG. Standard loads consume power but do not communicate with the TMG. Smart loads also consume power but can communicate with the TMG to protect the TMG. Smart loads allow the TMG controller to control their loads to prevent the grid from being overloaded. In addition, smart loads can notify the TMG controller about their pending activation. This notification will allow the TMG controller to ensure adequate power is available on the TMG for the additional load.

4.2.7. Type VII – Power Conversion. Type VII – Power Conversion systems enable Type I Power sources and Type II Energy Storage systems which produce different voltages and/or are Direct Current (DC), to connect to the TMG feeder bus. These systems convert the source/storage power to the common TMG bus AC voltage. These system can be separate items or can be integrated into the source/storage systems they support.

4.3. Interoperability. Individual equipment types specified above, shall communicate, and interact with one another in order to interoperate as a TMG, regardless of manufacturer or version, and without the need for operator intervention. Interactions include but are not limited to starting and shutdown, synchronization, grid connection/disconnection, parallel operation to include proper load sharing, load management, etc. Each item shall have standard power and communication connections to enable this interoperability.



4.4. Standalone Use.

4.4.1. Type I Power Generation Sources and Type II Energy Storage Equipment. Type I Power Generation Sources and Type II Energy Storage equipment shall be capable of operating independently to provide power in a standalone or spot generation configuration. Power generation sources and energy storage systems will be operated independent of the TMG for loads not on the TMG.

4.4.2. Type III Feeder and Type IV Distribution Systems Equipment. Type III Feeder and IV Distribution Systems equipment shall be capable of operation in an independent mode when they are independent of a TMG. Feeder and Distribution systems will be operated independent of the TMG if separated by a fault in the TMG or when providing power distribution for a standalone power source. In addition, Type III and Type IV equipment shall be capable or operating in a manual mode if automatic control is lost.

4.5. Degraded Operation. The TMG shall have the ability to degrade operations when experiencing grid instability, faults, or failures. The TMG shall be capable of autonomous and operator-implemented degraded operation during loss of communication events, equipment failures, and emergency conditions. These capabilities include selective isolation of equipment and segmentation of the grid into two or more smaller microgrids. The minimum degradation will be to disable the TMG controls and operate components in spot generation configurations.

4.6. Monitoring and Control. All types of TMG equipment shall provide status, configuration data, and control for individual equipment and the overall TMG to system operators. Access to this data and information shall be provided through human machine interfaces on the individual equipment and remote interfaces such as the Energy Informed Operations (EIO) dashboard or a standard notebook computer.

4.7. Cybersecurity. Cybersecurity is essential to the operation of the TMG. Although the ability to apply firmware/software upgrades is necessary, individual equipment shall be protected from the introduction of malicious firmware/software into the systems directly or through remote interfaces.

All types of TMG equipment shall conform to DOD Cybersecurity policy as specified in DODI 8500.01. Specific policy areas include: Risk Management, Operational Resilience, Integration and Interoperability, Cyberspace Defense, Performance, Identity Assurance, and Information Technology.

4.7.1. Risk Management Framework (RMF). Individual types of TMG equipment as well as the overall TMG, shall comply with DOD Risk Management Framework policy defined in DODI 8510.01. The RMF shall be based on the National Institute of Standards and Technology (NIST), RMF including Special Publication (SP) 800-39 Managing Information Security Risk: Organization, Mission, and Information System View, SP 800-53 Rev. 5 Security and Privacy Controls for Information Systems and Organizations, and SP 800-82 Rev. 2 Guide to Industrial Control Systems (ICS) Security. The RMF applies across the System Development Lifecycle (SDLC) of the individual items and how they communicate within the TMG. Applying standards-based cybersecurity as early as possible in the SDLC can also help improve its effectiveness at a reduced cost compared to waiting until later phases in the lifecycle.

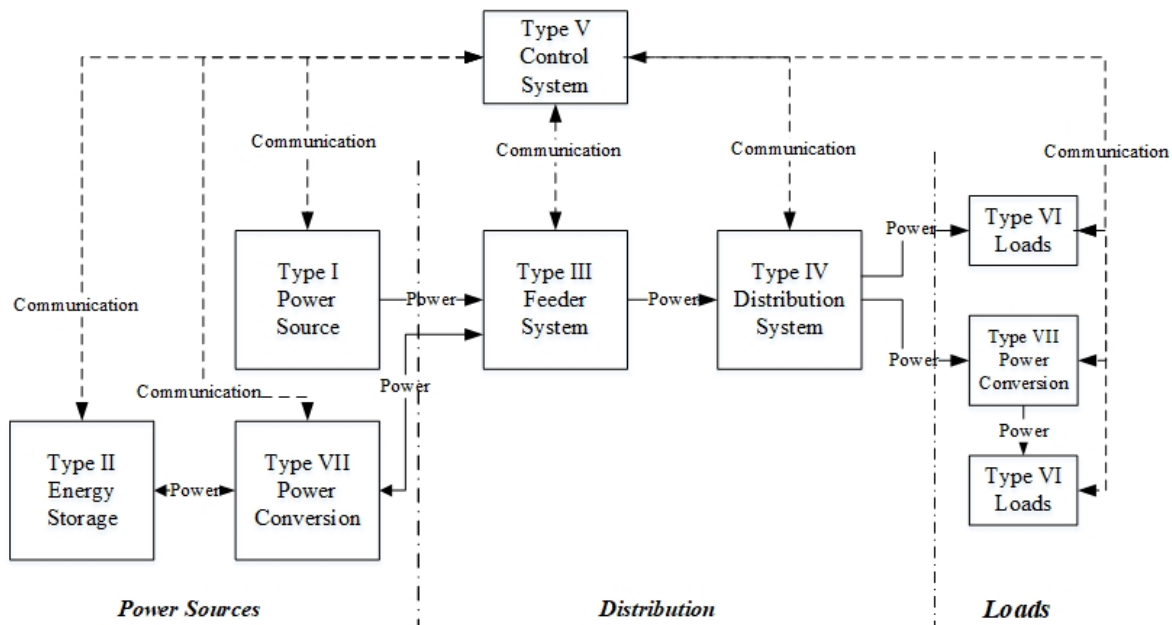
4.7.2. Operational Resilience. Individual types of TMG equipment as well as the overall TMG, shall ensure information, data, and services are available to other elements of the TMG.

Technology components (e.g., hardware and software) shall have the ability to reconfigure, optimize, and recover with defense in depth. Attempts made to reconfigure, and recover shall be logged in an audit file.

4.7.3. Access to firmware/software. Access control methods and procedures shall be implemented to mitigate cyber threats to individual equipment. Strong identities, assured access, trustworthy logging, and integrity-protected operations form the core list identified and defined within this document:

- a. Strong identities help maintain a trustworthy sense of what people, processes, and devices are taking action within the system.
- b. Assured access helps guarantee that authorized identities have appropriate access to system resources in order to, for example, initiate commands or collect telemetry.
- c. Trustworthy logging allows TMG Operators to operate and troubleshoot a running system and cyber analysts to perform forensic analysis after a cyber-event has been detected.
- d. Integrity-protected operations ensure the trustworthiness of data and operations within the system and help assure that accidental and malicious changes to the system are detectable.

4.8. Tactical Microgrid Architecture. The components of the Tactical Microgrid are required to interface with one another through a common power bus and microgrid control system communications bus. The basic architecture is shown in Figure 1 Tactical Microgrid Schematic. All types of TMG equipment shall conform to the following architecture requirements.



**Figure 1 - Tactical Microgrid Schematic**

4.8.1. Common Microgrid Control System. Type V Control Systems shall interface with other types of TMG equipment to provide automatic microgrid control of those devices. The control communications shall be accomplished over an OSI Layer 1 medium that is capable of Ethernet transmission. Standard connectors and cabling shall withstand environmental characteristics specified herein. The communication bus shall be capable of being disconnected without interrupting the operation of the TMG.

4.8.2. Common Microgrid Monitoring System. All types of TMG equipment shall have the capability to provide operational data to a central monitoring system using an OSI Layer 1 medium that is capable of Ethernet transmission. Standard connectors and cabling shall withstand environmental characteristics specified herein. The communication bus shall be capable of being disconnected without interrupting the operation of the TMG.

4.8.3. Levels of Control. The Tactical Microgrid equipment shall have internal controls and be capable of accepting TMG control from outside sources. The internal equipment controls enable the individual equipment to perform their functions independent from a TMG. The TMG control enables the primary functions of the TMG to be accomplished while connected to the individual components. In order to prevent control issues, the TMG level of control shall not override the internal control functions of the equipment. TMG level control shall activate/deactivate local equipment controls to meet TMG functionality.

4.8.4. Tactical Microgrid Control (TMC) Redundancy. Electricity is mission critical in many TMG applications. The TMG must provide continuous, reliable and stable electricity to the user. Type V Control Systems equipment provide the TMC. To ensure TMG reliability and stability, there is a need to have redundant TMCs. Redundancy is accomplished by incorporating Master and Standby modes into the TMC. This redundancy enables a second TMCs, operating in standby mode, to maintain TMG function when the TMC in the master mode fails. The relationship between TMCs is:

- All TMCs receive data from all devices.
- Only the TMC in the master mode can send commands or setting changes.
- Devices reject commands from other TMCs.
- Failure of the TMC in the master mode shall trigger the automatic selection of a different TMC to operate in the master mode. TMCs in standby mode shall monitor the TMC in the master mode commands and setting changes in order to maintain a stable grid when they become active. The TMCs in standby mode will automatically take control using these commands and settings until new commands and settings become necessary based on operational data.

4.8.4.1. Protection coordination. All microgrid devices have some level of automated, threshold-based fault protection. Protection coordination sequences contactors and engages loads/sources for power system black-start, circuit partitioning, and circuit re-closing. Protection coordination also monitors power flows to prevent an over-current condition on a Type III feeder or Type IV distribution bus.

4.8.5. Communications Architecture. All devices operating on the Tactical Microgrid shall operate on a common Ethernet communications network. The selected communication protocols are shown in Table 4.1.

**Table 4-1: Tactical Microgrid Communications Stack (TMCS)\***

Layer	TMS Implementation
Applications	Microgrid Control
Data Objects	TMCS Data Objects
Middleware	Data Distribution Service (DDS)
Transport	User Datagram Protocol (UDP)
Network	Internet Protocol Version 6 (IPv6)
Physical	Ethernet (802.3)

\*Need to add standards for DDS, UDP, IPv6, etc. Should be in Section 5.

**4.9.Data Model.** The data model is defined in terms of data flows and data types. Data flows define the motion of messages while data types define the contents of messages, as some tactical microgrid functions require coordination between devices.

The TMS data model includes all messages sent among TMS devices. These messages are used to communicate all system health, status, command, and control information. They are also recommended for communications with external networks.

The data model provides a common language used by all TMS devices. It is platform independent to allow flexibility in device design and implementation. It contains detailed data flow and data type definitions to guarantee interoperability between devices, regardless of make or mode.

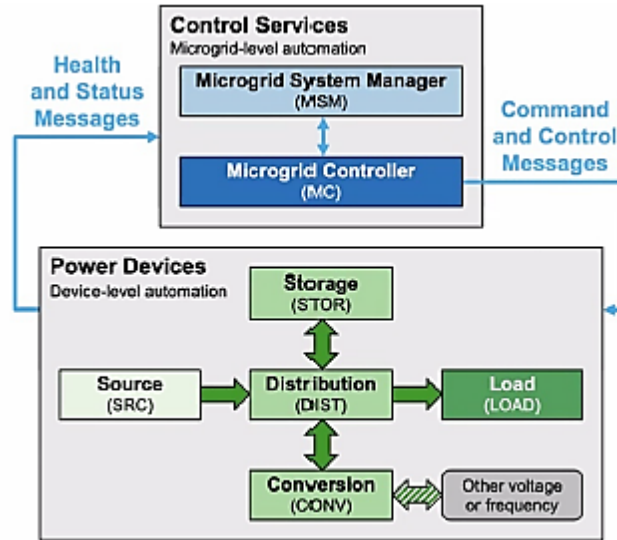
The data model allows for design and implementation flexibility while still taking into account such considerations as security, performance, simplicity, interoperability, and extensibility.

Tactical microgrid devices shall use this data model to implement all functions defined in this standard. Having a common data model is essential for interoperability between devices. Tactical microgrid devices may additionally use other communications to implement functions not defined in this standard. If such additional communications add functionality that is within the scope of this standard, then they should be proposed for inclusion in future revisions of this standard.

The roles and responsibilities for the TMG elements are defined in Section 5.3, with technical requirements defined in the appendices. Additional details are specified in the accompanying Implementation Guide.

**4.9.1. Device Identity.** In the TMS data model, every device shall have a unique identity, assigned by a Fingerprint, which is bound to a single, specific device role that has a number of built-in requirements and permissions. This approach simplifies the communications by preventing any ambiguity about how roles may interact. Under all circumstances, every TMS device role shall have its own unique Fingerprint. Since there will likely be multiple devices assigned to the same role, such as generators, the Fingerprint data type assigns a unique identity to each device, regardless of the device role. Data Types, such as Fingerprint, are discussed in Appendix A and Appendix B.

4.9.2. Device Roles. Device roles are defined building blocks within the TMS that provide modularized functionality. Each role is assigned based upon the device's capabilities in the TMS. Each device role is pre-defined in both functionality and power/communication interfaces to provide modular flexibility. Pre-defined components and interfaces allow customizable

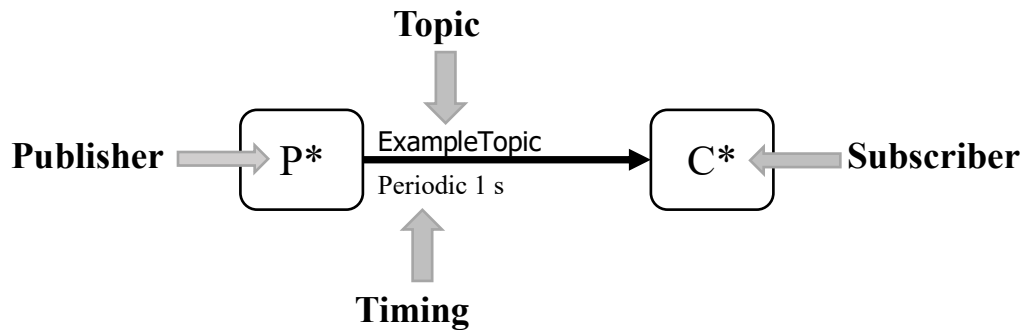


modularity in each TMS system to meet the unique needs of each microgrid installation. By using the device roles to define specific functionalities in the microgrid, any number of devices, regardless of capacity size, can be procured and assembled into a single TMS that meets the mission needs. There are two types of device roles in TMS: Control Services and Power Devices. Figure 2 below, shows the relationship between these roles.

**Figure 2 TMS Device Roles**

4.9.3. Platforms. A platform is a group of TMS devices assembled into a larger physical package. Platforms may integrate multiple device roles. Each device on a platform shall have a separate identity and role. Multi-role products shall have separate identities and device roles for each type of functionality. Multi-role products shall communicate as if they were multiple physically-separate devices. Each device role shall have its own data connection and run on a separate processor so each role can be physically isolated. If multiple devices are packaged into a single platform or multi-role product, they may share hardware or software components. If such sharing does occur, then all messages shall be sent separately for each contained device identity and role. Messages may be received singly and re-shared using internal mechanisms.

4.9.4. Data flows. TMS devices shall communicate through pre-defined data flows to share all operating information such as power commands and status. For simplicity, the device roles may be shown in the data flows throughout this document with abbreviated names - either P\* for the Power Device role or C\* for the Control Service device role - to represent any device in that category. Every data flow contains one or more topics. Each topic is defined with a name, publishers that send data, subscribers that receive data, a data type that specifies what values can be transferred, and timing that specifies when data is transferred. Data flows shall be defined using a publisher/subscriber (pub/sub) paradigm that defines the data flows in terms of logical connections. The underlying software and hardware may use buffers, filters, and a mix of broadcast, multicast, and point-to-point connections to implement the flow. These



*Figure 3 - Data Flow Notation*

details are specified by the OMG DDS standard as used by TMS. A data flow containing a single topic is provided in Figure 3. Each topic has the following key features:

**Publisher** is the Device Role(s) that sends these messages.

**Subscriber** is the Device Role(s) that receives these messages.

**Topic** is the name/identifier for these messages and the message flow.

Data Flow Pattern identifies the basic patterns of the message data that flows between devices,

**Timing** is the condition that causes messages to be sent and the reliability of the message flow.

4.9.5. Publishers and Subscribers. Device Roles shall be used to specify the source of the message, which is the publisher, and the receiver of the message, which is the subscriber, for each data flow. While a device role can be both a publisher and subscriber, typically in an individual flow it is one or the other. Conceptually, all publishers send all messages to all subscribers on a topic.

4.9.6. Topic and Message Flow. Topics shall be used to identify the data flows. Topics define a small group of TMS data flow message topics that have closely related values and match the publishers to the subscribers for each message chain. Multiple data flow topics are used together in larger patterns to achieve system-level TMS behaviors. Each data flow topic shall have a single unique data type with defined event trigger and reliability metrics,

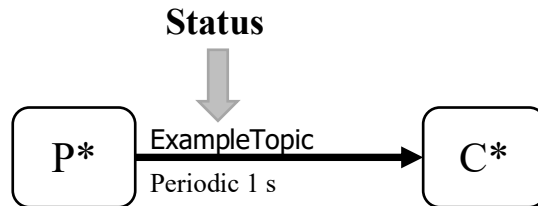
which are discussed below. The same data type shall be used by multiple data flow topics, providing a common connection between similar data flows.

The arrows in the data flow diagram indicate the flow of the message stream for the messages sent and received. The arrowhead indicates the direction that messages are being sent. The message stream for the data flows is published following specific data flow patterns that are pre-defined in the TMS. Arrows indicate messages that can be sent from both the initiating publisher and the subscriber in response to the publisher. Data flow patterns, including the subscriber response patterns, are discussed in further detail below.

**4.9.7. Data Types.** Devices shall use the TMS-defined data type that is assigned to each topic. The type identifies possible values for the messages where each message may have a different value. Platform-independent logical types are defined in Appendix B, Detailed Data Model Requirements. Platform-specific details such as data APIs and network serialization are defined in Appendix C, DDS Requirements.

**4.9.8. Data Flow Patterns.** The data flows identify and describe the system-level behaviors of TMS communication between devices through a set of pre-defined messages. There are three basic patterns of data flows that messages between devices follow: Status, Request/Response, and Custom.

**4.9.8.1. Status data flow pattern.** The status data flow messages shall provide status of the device that publishes the message. Figure 4 provides an example of a status data flow pattern. These messages do not require a reply or response from any subscribing devices. The timing of these messages shall occur when an event triggers the message to be sent. Examples of event triggers include when a device starts up or if an error condition occurs.

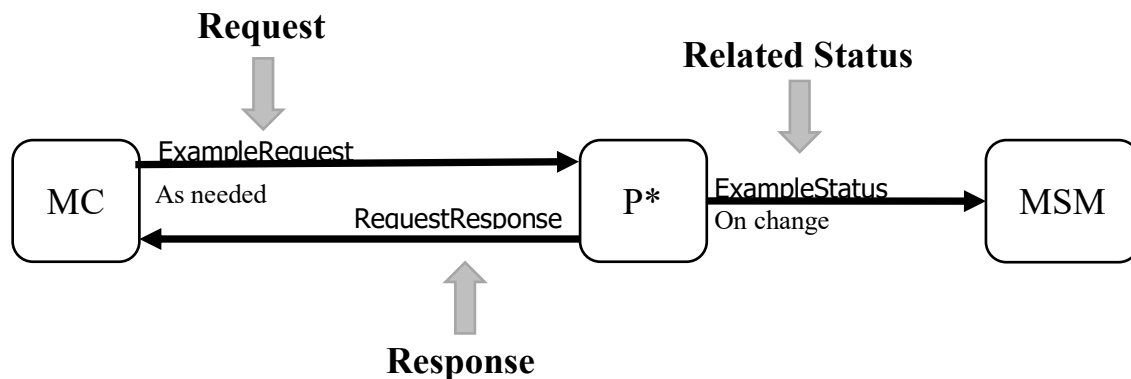


**Figure 4 - Sample Status Data Flow Pattern**

4.9.8.2. Request/Response data flow pattern. Request/Response messages shall require interaction between devices in the form of command and acknowledgment. Figure 5 provides an example of a request/response data flow pattern, where MC is the Microgrid Controller, P\* is the Power Device role, and MSM is the Microgrid System Manager.

A command is a request having immediate effect. Other requests may be deferred for later execution. A reply is a data sample update sent in response to a request.

Other responses include actions taken to complete a request. A Request is sent from one device to another, which responds to the request. Related status may be sent to a third device. An example of this data flow pattern is a health status request sent to a TMS device from the user interface to which the device responds and also sends additional related status.



*Figure 5 - Sample Request/Response Data Flow Pattern*

4.9.8.3. Custom data flow pattern. Custom messages shall have a unique interaction pattern that doesn't match the Status or Request/Response patterns. Each custom data flow pattern shall have message interactions that are unique and specific to that one condition. Typically custom patterns define a TMS system-level behavior.

4.9.9. Instances. Some data flows have a single value that applies to the entire microgrid. Other data flows have different values for each device in the microgrid or even for each power connector in a single device. These different values are called sample instances. Editorial note: Working on a summary of the various key values and where they are used. This is being revised in the tech refresh.

4.9.10. Timing Effective communications requires a careful balance between sending messages frequently, so important information is not delayed, and limiting the frequency and size of messages, so resources are not overloaded. Each topic shall have timing requirements that define when new messages are sent, how frequently new messages can be sent, and when old messages are re-sent. These timing requirements shall be used to define behaviors and to derive minimum resource allocations for software and hardware in TMS devices. Topic data values change over time. Each new value is called a sample. When a publisher updates a sample, one or more messages will be sent to transmit the new value. There is not a direct, one-to-one mapping between sample updates and messages. For example, the communications software may bundle multiple samples into a single message,



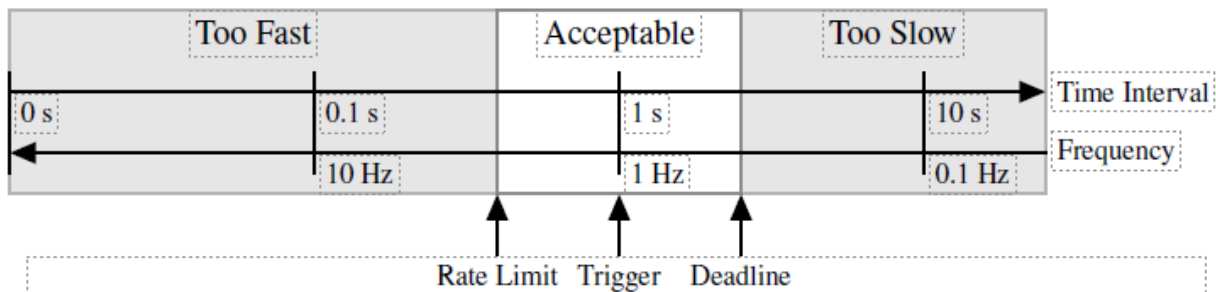
split a single sample across multiple messages, or re-transmit samples when messages are lost. Any such mapping is compliant as long as the overall timing requirements are met. The OMG DDS middleware allows the re-use of timing standards and software implementation. Timing requirements apply to the business logic in a TMS application, the OMG DDS middleware software library, operating system, and network hardware. There shall be four major timing requirements: data trigger, rate class, burst size, and quality of service (QoS). The first three requirements shall be defined in terms of sample updates. The QoS defines how these translate into messages. See Appendix A.8.5.

**4.9.10.1. Event trigger.** An event trigger is a pre-defined condition that causes sample updates. Fixed rate samples shall be triggered by a periodic timer set to a nominal rate. Variable rate samples shall be triggered by a given type of event. Details such as the event trigger for each topic type, rate limits, and timing parameters are defined in Appendix A, Data Model.

**4.9.10.2. Rate limit** The rate limit shall provide minimum separation time interval between sample updates. Upper bound on the rate of transmission and hence resource utilization. The publisher shall not send messages until after this minimum separation time interval. The network or subscriber may drop messages sent at a higher rate

**4.9.10.4. Deadline.** Deadline is the maximum separation time interval between sample updates. Lower bound on the rate of transmission and hence resource utilization. The publisher shall send updates before this maximum separation time interval. The subscriber may trigger an error-handling behavior if updates are not received on time.

**4.9.10.5. Quality of Service (QoS).** Quality of service is the policy defining how data samples are prioritized, buffered, sent, and re-sent as messages. Accounts for constraints in network connectivity. A fully specified policy is called a profile. The QoS profile may include information from the rate limit and deadline.



*Figure 7 Example Timing for a Periodic 1s topic*

## 5. DETAILED REQUIREMENTS

**5.1. Safety** The TMG interfaces required in this standard shall not cause harm to the Warfighter or damage equipment. Equipment developed in accordance with this standard shall comply with Army Regulation 385-10, The Army Safety Program. As a minimum, all interfaces shall meet various safety requirements as specified in MIL-STD-882 System Safety; NFPA-70 National Electric Code and NFPA-70E Standard for Electrical Safety in the Workplace.

5.2. Human Engineering The human interfaces required to implement the requirements specified herein, shall meet the design criteria requirements of MIL-STD-1472 Human Engineering.

5.3. TMG Equipment Roles and Responsibilities Each type of TMG equipment has specific roles and responsibilities when connected to the TMG. The roles and responsibilities define the functions of the item. There are related data flows in DDS which enable the functions to be performed. Detailed Data flows used to enable the equipment to meet their roles and responsibilities are defined in Appendix B. ~~Sample code for data flows are provided in Appendix ?.~~ The roles and responsibilities for each type of TMG equipment are as follows:

5.3.1. Type I – Power Sources The role of the power source is to provide power to the grid. Power sources are responsible for producing quality electricity for the TM. Power sources shall, accept commands from and provide operational parameters to Type V Control Systems in order to optimize microgrid operation. The power source shall provide power in accordance with its individual specification and shall be capable of accepting and implementing commands from Type V equipment either internal or external to the item.

5.3.1.1. Standard Power Source Functions The following standard power source functions and related commands shall be used to enable the power source role to be met:

**Table 5-1: Type I Standard Power Source Functions**

	Function	Reporting	Data Flows*
a.	Discovery	Publish/Subscribe	Appendix A, Discovery
b.	Start	Publish/Subscribe	Appendix A, Remote Start/Stop
c.	Feeder System Contactor Control	Publish/Subscribe	Appendix A; X.i
d.	Parallel Operation	Publish/Subscribe	Appendix A, Remote Start/Stop, AC Load Sharing, DC Load Sharing, Tuning Parameters
e.	Monitor Operational Parameters	Publish/Subscribe	Appendix A, Diagnostics, AC Measurements, DC Measurements, Hardware Measurements
f.	Drop Load	Publish/Subscribe	Appendix A, AC Load Sharing, DC Load Sharing
g.	Shutdown	Publish/Subscribe	Appendix A, Remote Start/Stop

\*Needs to be updated based on New Appendix A Refs.

- a. Discovery. The discovery function enables the device to be recognized as a member of the TM. This function initiates communication from the device to other members and determines acceptance and identification.
- b. Start. The start function enables the TMG controller to direct the power source device to prepare to provide power to the TMG bus. The start function initiates the operation of a power source prior to connection to the TM. This function varies with the type of

power source. For example, with a standard engine driven generator set, this function turns starts the engine and ensures frequency and voltage are produced. Whereas, with a wind turbine, this functions enables the wind turbine to turn and ensures frequency and voltage are produced.

- c. Feeder System Contactor Control. This function is a safety mechanism used to prevent power from being fed back to the load terminals of a non-operating power source. It enables the power source to send activation/deactivation direction to Type III Feeder System contactor to connect the power system to the TMG buss.
- d. Parallel Operation. The parallel operation function enables the TMG controller to direct the power source to connect power to the TMG buss. The parallel operation function synchronizes the power source to the TM, safely connects it to the TM, and enables proper load sharing between power sources and the TM.
- e. Monitor Operational Parameters. The monitor operational parameters function enables the TMG controller to monitor the operation of the power source. The monitor operational parameters function provides the operational conditions of the power source to the Type V control systems. These parameters include but are not limited to: voltage, frequency, current, total power, etc. In addition, any onboard health/diagnostics/prognostics or protective system indications shall also be reported.
- f. Drop Load. The drop load function enables the TMG controller to direct the power source to disconnect power from the TMG bus as needed. Dropping load disconnects power source from the bus, but does not shutdown the power source. This function prevents the need to restart the power source if the power is needed on the grid.
- g. Shutdown. The shutdown function enables the TMG controller to direct the power source to disconnect the load and shutdown the device. Shutdown shall stop an engine from operating or a wind turbine from rotating. The control systems for the power sources shall go into a standby mode where the TMG controller can maintain communication and send control commands.

**5.3.2. Type II – Energy Storage** The role of the energy storage systems (ESS) is to temporarily provide power during peak load demands or during operations when acoustic signature needs to be reduced. ESSs are responsible for storing excess energy and providing it back to the TMG as needed. ESSs shall provide power when the load exceeds current power production and additional power sources are needed to meet the load demand. ESSs shall accept commands from and provide operational parameters to Type V Control Systems in order to optimize microgrid operation. The ESS shall provide power in accordance with its individual specification and shall be capable of accepting and implementing commands from Type V equipment either internal or external to the item.

**5.3.2.1. Standard Energy Storage System Functions** The following standard energy storage system functions and related commands shall be used to enable the ESS role to be met:

**Table 5-2: Type II Standard Energy Storage System Functions**

	Function	Reporting	Data Flows*
a.	Discovery	Publish/Subscribe	Appendix A, para X.1

**Table 5-2: Type II Standard Energy Storage System Functions**

	Function	Reporting	Data Flows*
b.	Feeder System Contactor Control	Publish/Subscribe	Appendix A; X.i
c.	Parallel Operation	Publish/Subscribe	Appendix A, para X.1
d.	Monitor Operational Parameters	Publish/Subscribe	Appendix A, para X.2
e.	Drop Load	Publish/Subscribe	Appendix A, para X.3

\*Needs to be updated based on New Appendix A Refs.

- a. Discovery. The discovery function enables the device to be recognized as a member of the TM. This function initiates communication from the device to other members and determines acceptance and identification.
- b. Feeder System Contactor Control. This function is a safety mechanism used to prevent power from being fed back to the load terminals of a non-operating ESS. It enables the ESS to send activation/deactivation direction to Type III Feeder System contactor to connect the ESS to the TMG buss.
- c. Parallel Operation. The parallel operation function enables the TMG controller to direct the ESS to connect power to the TMG buss. The parallel operation function synchronizes the ESS to the TM, safely connects it to the TM, and enables proper load sharing between ESSs and the TMG buss.
- d. Monitor Operational Parameters. The monitor operational parameters function enables the TMG controller to monitor the operation of the ESS. The monitor operational parameters function provides the operational conditions of the ESS to the Type V control systems. These parameters include but are not limited to: voltage, frequency, current, total power, etc. In addition, any onboard health/diagnostics/prognostics or protective system indications shall also be reported.
- e. Drop Load. The drop load function enables the TMG controller to direct the power source to disconnect power from the TMG bus as needed. Dropping load disconnects the ESS from the bus, but does not shut down the ESS. This function prevents the need to restart the ESS if the power is needed on the grid.

5.3.3. Type III – Feeder Systems The role of feeder system is to establish the interconnected power and communications bus for the TM. Feeder systems are responsible for providing electricity from the source to the load in a safe manner. Feeder systems shall accept power from the Type I Power Sources and Type II Energy Storage Systems and distribute it to the Type IV Distribution Systems and various standard and Type VI loads. Feeder systems shall accept commands from and provide operational parameters to Type V Control Systems in order to optimize microgrid operation. Feeder systems shall accept commands from Type I and Type II equipment to control power isolation contactors. The feeder system shall distribute power in

accordance with its individual specification and shall be capable of accepting and implementing commands from Type V equipment either internal or external to the item.

In addition to the power interface, the Type III feeders systems will function as the communications hub for the TM. Each power connection will have a matching communications port to enable all of the TMG elements to communicate with each other.

**5.3.3.1. Standard Feeder System Functions** The following standard feeder system functions and related commands shall be used to enable the feeder system role to be met:

**Table 5-3: Type III Feeder System Functions**

	Function	Reporting	Data Flows*
a.	Discovery	Publish/Subscribe	Appendix A, Discovery
b.	Feeder System Contactor Control	Publish/Subscribe	Appendix A; X.i
c.	Close Circuit	Publish/Subscribe	Appendix A; X.i
d.	Monitor Operational Parameters	Publish/Subscribe-o	Appendix A, Diagnostics, AC Measurements, DC Measurements, Hardware Measurements
e.	Open Circuit	Publish/Subscribe	Appendix A, AC Load Sharing, DC Load Sharing

\*Needs to be updated based on New Appendix A Refs.

- a. Discovery. The discovery function enables the device to be recognized as a member of the TM. This function initiates communication from the device to other members and determines acceptance and identification.
- b. Feeder System Contactor Control. This function is a safety mechanism used to prevent power from being fed back to the load terminals of a non-operating power source or ESS. It enables the feeder system to receive contactor activation/deactivation direction from the Type I and Type II equipment to connect power to the TMG buss.
- c. Close Circuit. The close circuit function enables the TMG controller to direct the feeder system to connect power from the power bus to the individual power circuits.
- d. Monitor Operational Parameters. The monitor operational parameters function enables the TMG controller to monitor the operation of the feeder system. The monitor operational parameters function provides the operational conditions of the feeder system to the Type V control systems. These parameters include but are not limited to: voltage, frequency, current, total power, state of circuit (open/closed/tripped), etc. In addition, any onboard health/diagnostics/prognostics or protective system indications shall also be reported.
- e. Open Circuit. The open circuit function enables the TMG controller to direct the feeder system to disconnect power from the individual power circuits.

5.3.4. Type IV – Distribution Systems The role of distribution systems is to provide the power interface from the Type III feeder system to the load and function as the communications hub for smart load to communicate with the other elements of the TM. Distribution systems are responsible of distributing electricity from the TMG buss, through individual branch circuits, to the loads. Distribution systems shall divide the TMG power bus into individual branch circuits and shall provide the final overcurrent protection of those circuits to meet load requirements. Distribution Systems shall accept commands from and provide operational parameters to Type V Control Systems in order to optimize microgrid operation. The distribution system shall distribute power in accordance with its individual specification and shall be capable of accepting and implementing commands from Type V equipment either internal or external to the item.

In addition to the power interface, the Type IV distribution systems will function as the communications hub for the smart loads and of elements of the TM. Each power connection will have a matching communications port to enable all of the smart loads to communicate with the other elements of the TM.

5.3.4.1. Standard Distribution System Functions The following standard distribution system functions and related commands shall be used to enable the distribution system role to be met:

**Table 5-4: Type IV Standard Distribution System Functions**

	Function	Reporting	Data Flow*
a.	Discovery	Publish/Subscribe	Appendix A, Discovery
b.	Close Circuit	Publish/Subscribe	Appendix A; X.i
c.	Monitor Operational Parameters	Publish/Subscribe	Appendix A, Diagnostics, AC Measurements, DC Measurements, Hardware Measurements
d.	Open Circuit	Publish/Subscribe	Appendix A, AC Load Sharing, DC Load Sharing

\*Needs to be updated based on New Appendix A Refs.

- a. Discovery. The discovery function enables the device to be recognized as a member of the TM. This function initiates communication from the device to other members and determines acceptance and identification.
- b. Close Circuit. The close circuit function enables the TMG controller to direct the feeder system to connect power from the power bus to the individual branch circuits.
- c. Monitor Operational Parameters. The monitor operational parameters function enables the TMG controller to monitor the operation of the distribution system. The monitor operational parameters function provides the operational conditions of the distribution system to the Type V control systems. These parameters include but are not limited to: voltage, frequency, current, total power, state of circuit (open/closed/tripped), etc. In addition, any onboard health/diagnostics/prognostics or protective system indications shall also be reported.
- d. Open Circuit. The open circuit function enables the TMG controller to direct the feeder system to disconnect power from the individual branch circuits.



5.3.5. Type V – Control Systems The role of the control system is to communicate with all of the TMG equipment in order to optimize TMG operation. The control system is responsible for controlling the TMG membership; identifying each member; receiving operational data from each member; analyzing the data; and providing commands back to the members in order to meet optimization goals. In addition, the control system is responsible for providing the user interface in order to control the TM. The control system shall utilize the commands specified herein to control the functions of the individual equipment of the TM. The control system shall provide monitoring and controls in accordance with its individual specification and be capable of accepting data and initiating commands to all TMG equipment. Standard Control System Source Functions The following standard control system functions and related commands shall be used to enable the TMG optimization role to be met:

**Table 5-5: Type V Standard Control System Functions**

	Function	Reporting	Data Flows*
a.	Discovery	Publish/Subscribe	Appendix A, Discovery
b.	Start	Publish/Subscribe	Appendix A, Remote Start/Stop
c.	Intent to Connect	Publish/Subscribe	Appendix A
d.	Parallel Operation	Publish/Subscribe	Appendix A, Remote Start/Stop, AC Load Sharing, DC Load Sharing, Tuning Parameters
e.	Close Circuit	Publish/Subscribe	Appendix A; X.i
f.	Monitor Operational Parameters	Publish/Subscribe	Appendix A, Diagnostics, AC Measurements, DC Measurements, Hardware Measurements
g.	Auto Restart	Publish/Subscribe	Appendix A, XX.XX
h.	Open Circuit	Publish/Subscribe	Appendix A, AC Load Sharing, DC Load Sharing
i.	Drop Load	Publish/Subscribe	Appendix A, AC Load Sharing, DC Load Sharing
j.	Shutdown	Publish/Subscribe	Appendix A, Remote Start/Stop

\*Needs to be updated based on New Appendix A Refs.

- a. Discovery. The discovery function enables the device to be recognized as a member of the TM. This function initiates communication from the device to other members and determines acceptance and identification.
- b. Start. The start function enables the TMG controller to direct the power source device to prepare to provide power to the TMG bus. The start function initiates the operation of a power source prior to connection to the TM. This function varies with the type of power source. For example, with a standard engine driven generator set, this function turns starts the engine and ensures frequency and voltage are produced. Whereas, with a wind turbine, this functions enables the wind turbine to turn and ensures frequency and voltage are produced.
- c. Intent to connect. The Intent to connect function enables the TMG controller to

MIL-STD-XXX  
DRAFT 19 Apr 2021

determine if the TMG has available power to meet the load demands. TMG controller shall review the message, evaluate the available power, and provide permission if power is available.

- d. Parallel Operation. The parallel operation function enables the TMG controller to direct the power source to connect power to the TMG buss. The parallel operation function synchronizes the power source to the TM, safely connects power source to the TM, and enables proper load sharing between power sources.
- e. Close Circuit. The close circuit function enables the TMG controller to direct the feeder system to connect power from the power bus to the individual branch circuits.
- f. Monitor Operational Parameters. The monitor operational parameters function enables the TMG controller to monitor the operation of the power source. The monitor operational parameters function provides the operational conditions of the power source to the Type V control systems. These parameters include but are not limited to voltage, frequency, current, total power, etc. In addition, any onboard health/diagnostics/prognostics or protective system indications shall also be reported.
- g. Auto Restart. The auto restart function provides for a controlled restart of the TMG if the entire grid goes down. The controller shall attempt to restart the grid in an orderly fashion unless it cannot be restore within a fixed amount of time.
- h. Open Circuit. The open circuit function enables the TMG controller to direct the feeder system to disconnect power from the individual branch circuits.
- i. Drop Load. The drop load function enables the TMG controller to direct the power source to disconnect power from the TMG bus as needed. Dropping load disconnects the load from the bus, but does not shutdown the power source. This function prevents the need to restart the power source if the power is needed on the grid.
- j. Shutdown. The shutdown function enables the TMG controller to direct the power source to disconnect the load and shutdown the device. Shutdown shall stop an engine from operating or a wind turbine from rotating. The control systems for the power sources shall go into a standby mode where the TMG controller can maintain communication and send control commands.

5.3.5.2. Operator Interface FunctionType V Control systems shall provide the operator interface for the TM. The operator shall have the ability to monitor the TM, define the control parameters, and take action to correct various issues with the TMG through the control system. This interface shall be accomplish as specified in **Appendix C - Operator Interface**.



**5.3.6. Type VI – Loads** Type VI loads are power consumers which comply with the communications requirements of this standard. The role of Type VI loads is to use the power from the TMs to meet mission needs. The responsibility of the TMS compliant loads is to inform the Type V control system of intent to activate and use electricity or deactivate if already operating. These loads shall communicate with the Type V control systems to announce their intent to activate or deactivate to allow the Type V control system to ensure enough power is available on the grid. Loads accept commands from the Type V Control Systems in order to optimize microgrid operation. The load shall consume power in accordance with its individual specification and be capable of accepting commands from Type V equipment either internal or external to the item.

Non TMS compliant loads can be connected to the TMG, but the control functions are implemented through the Type III Feeder and Type IV Distribution systems. The control systems shall connect and disconnect these loads by activating/deactivating the circuits where they are connected. No direct communication with these loads is necessary.

**5.3.6.1. Standard Load Functions** The following standard load functions and related commands shall be used to enable the load role to be met:

**Table 6-5: Type VI Standard Control System Functions**

	Function	Reporting	Data Flows*
a.	Discovery	Publish/Subscribe	Appendix A, Discovery
b.	Intent to connect	Publish/Subscribe	Appendix A XI
c.	Close Circuit	Publish/Subscribe	Appendix A; X.i
d.	Monitor Operational Parameters	Publish/Subscribe	Appendix A, Diagnostics, AC Measurements, DC Measurements, Hardware Measurements
e.	Open Circuit	Publish/Subscribe	Appendix A, AC Load Sharing, DC Load Sharing

\*Needs to be updated based on New Appendix A Refs.

- a. Discovery. The discovery function enables the device to be recognized as a member of the TM. This function initiates communication from the device to other members and determines acceptance and identification.
- b. Intent to connect. The Intent to connect function enables the load to request permission from the TMG controller to connect to the TMG. TMG controller shall review the message, evaluate the available power, and provide permission if power is available.
- c. Close Circuit. The close circuit function enables the TMG controller to direct the load to connect to the TMG.
- d. Monitor Operational Parameters. The monitor operational parameters function enables the load to provide various operational parameters and equipment health status to the TMG controller.
- e. Open Circuit. The open circuit function enables the TMG controller to direct the feeder system to disconnect power from the individual branch circuits.

**5.3.7. Type VII – Power Conversion** The role of Power Conversion equipment is the enable Type I power sources or Type II energy storage systems with different voltages and currents to connect to the TMG. Power conversion equipment shall be required to provide operational parameters and accept commands from the Type V Control Systems in order to optimize microgrid operation. The power conversion systems provide power in accordance with its individual specification and be capable of accepting commands from Type V equipment either internal or external to the item.

**5.3.7.1. Standard Power Conversion Functions** Since power conversion equipment must function with Type I power sources or Type II energy storage systems, the standard functions shall be the same standard functions as the Type I and Type II equipment. The following standard power source functions and related commands shall be used to enable the power conversion role to be met:

**Table 5-7: Type VII Standard Power Conversion Functions**

	Function	Reporting	Data Flows*
a.	Discovery	Publish/Subscribe	Appendix A, Discovery
b.	Start	Publish/Subscribe	Appendix A, Remote Start/Stop
c.	Feeder System Contactor Control	Publish/Subscribe	Appendix A; X.i
d.	Parallel Operation	Publish/Subscribe	Appendix A, Remote Start/Stop, AC Load Sharing, DC Load Sharing, Tuning Parameters
e.	Monitor Operational Parameters	Publish/Subscribe	Appendix A, Diagnostics, AC Measurements, DC Measurements, Hardware Measurements
f.	Drop Load	Publish/Subscribe	Appendix A, AC Load Sharing, DC Load Sharing
g.	Shutdown	Publish/Subscribe	Appendix A, Remote Start/Stop

\*Needs to be updated based on New Appendix A Refs.

- a. Discovery. The discovery function enables the device to be recognized as a member of the TM. This function initiates communication from the device to other members and determines acceptance and identification.
- b. Start. The start function enables the TMG controller to direct the power source device to prepare to provide power to the TMG bus. The start function initiates the operation of a power source prior to connection to the TM. This function varies with the type of power source. For example, with a standard engine driven generator set, this function turns starts the engine and ensures frequency and voltage are produced. Whereas, with a wind turbine, this functions enables the wind turbine to turn and ensures frequency and voltage are produced.
- c. Feeder System Contactor Control. This function is a safety mechanism used to

prevent power from being fed back to the load terminals of a non-operating power source. It enables the power source to send activation/deactivation direction to Type III Feeder System contactor to connect the power system to the TMG buss.

- d. Parallel Operation. The parallel operation function enables the TMG controller to direct the power source to connect power to the TMG buss. The parallel operation function synchronizes the power source to the TM, safely connects it to the TM, and enables proper load sharing between power sources and the TM.
- e. Monitor Operational Parameters. The monitor operational parameters function enables the TMG controller to monitor the operation of the power source. The monitor operational parameters function provides the operational conditions of the power source to the Type V control systems. These parameters include but are not limited to: voltage, frequency, current, total power, etc. In addition, any onboard health/diagnostics/prognostics or protective system indications shall also be reported.
- f. Drop Load. The drop load function enables the TMG controller to direct the power source to disconnect power from the TMG bus as needed. Dropping load disconnects power source from the bus, but does not shutdown the power source. This function prevents the need to restart the power source if the power is needed on the grid.
- g. Shutdown. The shutdown function enables the TMG controller to direct the power source to disconnect the load and shutdown the device. Shutdown shall stop an engine from operating or a wind turbine from rotating. The control systems for the power sources shall go into a standby mode where the TMG controller can maintain communication and send control commands.

**5.4. Communications Interface** The communications interface (CI) is how the communications and controls will be transmitted from one device to another within the TMG. The CI is described in terms of the Information technology - Open Systems Interconnection (OSI) model as defined in ISO/IEC 7498. The model contains various layers that define the communications. Table 5-8 Tactical Microgrid Communications Stack defines the various layers and protocols within the layer used in the TMG.

**Table 5-8: Tactical Microgrid Communications Stack\***

OSI Layer	TMSC Implementation
Application	Device Controller
Presentation	TMSC Data Model
Session	OMG Data Distribution Service (DDS)
Transport	User Datagram Protocol (UDP)
Network	Internet Protocol Version 6 (IPv6)
Physical	Ethernet (802.3)

\*Need to add standards for DDS, UDP, IPv6, etc. Should be in Section

**5.4.1. Application Layer.** The application layer enables each device controller to share information with other devices in the TMG to make decisions and control the TMG. Each type of device may have its own implementation.

944

945 5.4.2. Presentation Layer. The data model provides a common language for communication  
946 between TMG devices. Data flows define the movement of information and data types define  
947 the range of possible values.

948 5.4.3. Session Layer. The OMG Data Distribution Service (OMG DDS) provides portable APIs  
949 for sending and receiving information. The DDS-RTPS provides interoperable quality of service  
950 over transport layer.

951 5.4.4. Transport Layer. The User Datagram Protocol (UDP) provides efficient communications  
952 over the network. Reliability is handled by OMG DDS as required.

953 5.4.5. Network Layer. The network layer provides the functional and procedural means for  
954 connectionless-mode or connection mode transmission among transport-entities and, therefore,  
955 provides to the transport-entities independence of routing and relay considerations. The network  
956 layer shall use Internet Protocol Version 6 (IPv6) IAW RFC8200 (STD86).

957 5.4.6. Physical Layer. The physical layer provides the mechanical, electrical, functional and  
958 procedural means to activate, maintain, and de-activate physical-connections for bit transmission  
959 between data-link-entities. The physical layer shall be Ethernet (802.3) based IAW IEEE 802.3.  
960 Each TMG item shall have two connection point to allow for multiple items to be daisy chained.

961 ADDITIONAL PARAGRAPHS MAY BE NEEDED.

## 962 6. NOTES

963 (This section contains information of a general or explanatory nature that may be helpful, but is not  
964 mandatory.)

965 6.1. Intended Use. Standards covered by this standard are intended for use in acquisition to  
966 establish requirements for military-unique communication interfaces for Tactical Microgrid  
967 equipment.

968 6.2. Acquisition requirements. Acquisition documents should specify the title, number, and date  
969 of this standard.

970 6.3. Tailoring guidance. To ensure proper application of this standard, invitation for bids,  
971 request for proposals, and contractual statements of work should tailor the requirements in  
972 sections 4 and 5 of this standard to exclude any unnecessary requirements. For example, if the  
973 statement of work requires the development of a revised standard, then all material related to  
974 notices should be excluded.

975 6.4. Subject term (key word) Listing.

976 Microgrid  
977 Energy Storage  
978 Generator  
979 Feeder System  
980 Load  
981

MIL-STD-XXX  
DRAFT 19 Apr 2021

CONCLUDING MATERIAL

Custodians:

Army – CR4

Navy – ?

Air Force – ?

Review activities:

Army – ?

Navy – ?

Air Force – ?

Preparing activity:

Army-CR4

NOTE: The activities listed above were interested in this document as of the date of this document. Since organizations and responsibilities can change, you should verify the currency of the information above using the ASSIST Online database at <https://assist.dla.mil>.

A1

METRIC
--------

A2

MIL-STD-TMS

A3

# DEPARTMENT OF DEFENSE INTERFACE STANDARD

A4

A5

## APPENDICES

A6

## TMS DATA MODEL

A7 Contents

A8	<b>A General Data Model Requirements</b>	<b>22</b>
A9	A.1 Introduction . . . . .	22
A10	A.2 Scope and Purpose . . . . .	22
A11	A.3 Implementation Considerations . . . . .	23
A12	A.4 General Requirements . . . . .	24
A13	A.5 Device Identity . . . . .	24
A14	A.6 Device Roles . . . . .	24
A15	A.6.1 Control Services . . . . .	25
A16	A.6.2 Power Devices . . . . .	25
A17	A.6.3 Platforms and Multi-Role Products . . . . .	26
A18	A.7 Consensus . . . . .	27
A19	A.8 Data Flows . . . . .	29
A20	A.8.1 Publishers and Subscribers . . . . .	30
A21	A.8.2 Topic and Message Flow . . . . .	30
A22	A.8.3 Data Flow Patterns . . . . .	30
A23	A.8.3.1 Status Data Flows . . . . .	31
A24	A.8.3.2 Request/Response Data Flows . . . . .	31
A25	A.8.3.3 Request/Response with Status Data Flows . . . . .	32
A26	A.8.3.4 Custom Data Flows . . . . .	32
A27	A.8.4 Instances and Key Values . . . . .	32
A28	A.8.5 Timing . . . . .	33
A29	A.8.5.1 Data Trigger . . . . .	35
A30	A.8.5.2 Rate Class . . . . .	36
A31	A.8.5.3 Burst Size . . . . .	39
A32	A.8.5.4 Quality of Service (QoS) . . . . .	40
A33	A.8.6 Quality of Service Options . . . . .	40
A34	A.8.6.1 Ordering . . . . .	40
A35	A.8.6.2 Coherency . . . . .	40
A36	A.8.6.3 Durability . . . . .	41
A37	A.8.6.4 History . . . . .	41
A38	A.8.6.5 Liveliness . . . . .	41
A39	A.8.6.6 Real-Time . . . . .	41
A40	A.8.6.7 Reliability . . . . .	41

A41	A.8.7	Optional Topics . . . . .	41
A42	A.9	Data Types . . . . .	43
A43	A.9.1	Type Hierarchy . . . . .	43
A44	A.9.2	Type Names . . . . .	43
A45	A.9.3	Type Namespaces . . . . .	44
A46	A.9.4	Units of Measure . . . . .	45
A47	A.9.5	Primitive Data Types . . . . .	46
A48	A.9.5.1	Boolean . . . . .	46
A49	A.9.5.2	Character . . . . .	46
A50	A.9.5.3	Octet . . . . .	47
A51	A.9.5.4	Integer Numbers . . . . .	47
A52	A.9.5.5	Floating-Point Numbers . . . . .	48
A53	A.9.5.6	Enumeration . . . . .	48
A54	A.9.6	Constructed Data Types . . . . .	48
A55	A.9.6.1	Attributes . . . . .	49
A56	A.9.6.2	Constants . . . . .	49
A57	A.9.6.3	Structure . . . . .	49
A58	A.9.6.4	Array . . . . .	50
A59	A.9.6.5	Sequence . . . . .	50
A60	A.9.6.6	String . . . . .	51
A61	A.9.6.7	Optional . . . . .	52
A62	A.9.6.8	Typedef . . . . .	52
A63	A.9.7	Domain-Specific Data Types . . . . .	52
A64	A.9.8	Default Values . . . . .	53
A65	A.9.9	Annotations . . . . .	53
A66	A.9.9.1	Units . . . . .	53
A67	A.9.9.2	Extensibility . . . . .	53
A68	A.10	Resource Allocation . . . . .	54
A69	A.11	Data Model Extensions . . . . .	55
A70	<b>B</b>	<b>Detailed Data Model Requirements</b>	<b>57</b>
A71	B.1	Device Role Schematics . . . . .	59
A72	B.1.1	Source Device Role . . . . .	59
A73	B.1.2	Distribution Device Role . . . . .	61
A74	B.1.3	Storage Device Role . . . . .	62
A75	B.1.4	Load Device Role . . . . .	63
A76	B.1.5	Conversion Device Role . . . . .	63
A77	B.2	Request/Response Pattern . . . . .	64
A78	B.2.1	Data Flows . . . . .	65
A79	B.2.2	Data Types . . . . .	66
A80	B.2.2.1	tms.Reply . . . . .	67
A81	B.2.2.2	tms.Fingerprint . . . . .	68



*TMS Data Model*  
*DRAFT 19 Apr 2021*

A82	B.2.2.3	tms.ConfigId	68
A83	B.2.2.4	tms.PowerPortNumber	69
A84	B.2.2.5	tms.RequestSequence	69
A85	B.2.2.6	tms.ReplyStatus	70
A86	B.2.2.7	tms.String32	72
A87	B.3	Identity	72
A88	B.3.1	Data Types	72
A89	B.3.1.1	tms.Fingerprint	72
A90	B.4	Heartbeat	73
A91	B.4.1	Data Flows	73
A92	B.4.2	Data Types	74
A93	B.4.2.1	tms.Heartbeat	74
A94	B.4.2.2	tms.Fingerprint	74
A95	B.5	Device Announcement	74
A96	B.5.1	Data Flows	75
A97	B.5.2	Data Types	76
A98	B.5.2.1	tms.DeviceInfo	77
A99	B.5.2.2	tms.Fingerprint	78
A100	B.5.2.3	tms.DeviceRole	79
A101	B.5.2.4	tms.ProductInfo	80
A102	B.5.2.5	tms.NatoStockNumber	81
A103	B.5.2.6	tms.GlobalTradeItemNumber	81
A104	B.5.2.7	tms.String32	81
A105	B.5.2.8	tms.TopicInfo	82
A106	B.5.2.9	tms.TopicConditionSequence	82
A107	B.5.2.10	tms.String1-32	83
A108	B.5.2.11	tms.ControlHardwareInfo	83
A109	B.5.2.12	tms.PowerHardwareInfo	84
A110	B.5.2.13	tms.EngineInfo	85
A111	B.5.2.14	tms.FuelInfo	86
A112	B.5.2.15	tms.GeneratorInfo	86
A113	B.5.2.16	tms.EnergyStorageInfo	87
A114	B.5.2.17	tms.PowerElectronicsInfo	88
A115	B.5.2.18	tms.TransformerInfo	89
A116	B.5.2.19	tms.ThermalInfo	89
A117	B.5.2.20	tms.ThermalZoneSequence	89
A118	B.5.2.21	tms.ParameterMetadataSequence	90
A119	B.5.2.22	tms.ParameterMetadata	91
A120	B.5.2.23	tms.String16	92
A121	B.5.2.24	tms.EnumLabelsSequence	92
A122	B.5.2.25	tms.ControlServiceInfo	92
A123	B.5.2.26	tms.MicrogridControllerInfo	93

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A124	B.5.2.27	tms.MicrogridControllerFeatureSequence . . . . .	93
A125	B.5.2.28	tms.MicrogridControllerFeature . . . . .	93
A126	B.5.2.29	tms.MicrogridDashboardInfo . . . . .	94
A127	B.5.2.30	tms.MicrogridDashboardFeatureSequence . . . . .	94
A128	B.5.2.31	tms.MicrogridDashboardFeature . . . . .	94
A129	B.5.2.32	tms.PowerDeviceInfo . . . . .	95
A130	B.5.2.33	tms.PowerPortInfoSequence . . . . .	96
A131	B.5.2.34	tms.PowerPortInfo . . . . .	97
A132	B.5.2.35	tms.PowerPortNumber . . . . .	99
A133	B.5.2.36	tms.PowerPortDirectionality . . . . .	99
A134	B.5.2.37	tms.PowerConnectorType . . . . .	100
A135	B.5.2.38	tms.PowerConnectorPolarity . . . . .	101
A136	B.5.2.39	tms.PowerConnectorPhases . . . . .	101
A137	B.5.2.40	tms.SwitchFeatureSequence . . . . .	102
A138	B.5.2.41	tms.SwitchFeature . . . . .	102
A139	B.5.2.42	tms.ConnectorFeatureSequence . . . . .	103
A140	B.5.2.43	tms.ConnectorFeature . . . . .	103
A141	B.5.2.44	tms.GroundingInfoSequence . . . . .	103
A142	B.5.2.45	tms.GroundingInfo . . . . .	104
A143	B.5.2.46	tms.GroundingCircuitNumber . . . . .	104
A144	B.5.2.47	tms.GroundingDesignType . . . . .	105
A145	B.5.2.48	tms.PowerPortNumberSequence . . . . .	105
A146	B.5.2.49	tms.ConversionInfo . . . . .	106
A147	B.5.2.50	tms.ConversionFeatureSequence . . . . .	106
A148	B.5.2.51	tms.ConversionFeature . . . . .	106
A149	B.5.2.52	tms.ActiveConversionInfo . . . . .	107
A150	B.5.2.53	tms.LoadSharingInfoSequence . . . . .	107
A151	B.5.2.54	tms.LoadSharingInfo . . . . .	108
A152	B.5.2.55	tms.PassiveConversionInfo . . . . .	110
A153	B.5.2.56	tms.PowerPortConversionInfoSequence . . . . .	110
A154	B.5.2.57	tms.PowerPortConversionInfo . . . . .	110
A155	B.5.2.58	tms.ConversionTapInfoSequence . . . . .	111
A156	B.5.2.59	tms.ConversionTapInfo . . . . .	111
A157	B.5.2.60	tms.TapNumber . . . . .	111
A158	B.5.2.61	tms.DistributionInfo . . . . .	112
A159	B.5.2.62	tms.DistributionFeatureSequence . . . . .	112
A160	B.5.2.63	tms.DistributionFeature . . . . .	112
A161	B.5.2.64	tms.SourceInfo . . . . .	113
A162	B.5.2.65	tms.SourceFeatureSequence . . . . .	113
A163	B.5.2.66	tms.SourceFeature . . . . .	114
A164	B.5.2.67	tms.StorageInfo . . . . .	114
A165	B.5.2.68	tms.StorageFeatureSequence . . . . .	115

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A166	B.5.2.69	tms.StorageFeature	115
A167	B.5.2.70	tms.LoadInfo	116
A168	B.5.2.71	tms.LoadFeatureSequence	116
A169	B.5.2.72	tms.LoadFeature	117
A170	B.6	Nickname	117
A171	B.6.1	Data Flows	117
A172	B.6.2	Data Types	118
A173	B.6.2.1	tms.FingerprintNickname	119
A174	B.6.2.2	tms.ChangeNicknameRequest	120
A175	B.6.2.3	tms.Reply	120
A176	B.6.2.4	tms.Fingerprint	120
A177	B.6.2.5	tms.String32	120
A178	B.6.2.6	tms.DeviceRequest	121
A179	B.6.2.7	tms.RequestSequence	121
A180	B.6.2.8	tms.ConfigId	121
A181	B.6.2.9	tms.PowerPortNumber	121
A182	B.6.2.10	tms.ReplyStatus	121
A183	B.7	Device Icon	121
A184	B.7.1	Data Flows	121
A185	B.7.2	Data Types	122
A186	B.7.2.1	tms.DeviceIcon	123
A187	B.7.2.2	tms.Fingerprint	123
A188	B.7.2.3	tms.String32	123
A189	B.7.2.4	tms.OctetSequence	124
A190	B.8	Control Hardware Status	124
A191	B.8.1	Data Flows	124
A192	B.8.2	Data Types	125
A193	B.8.2.1	tms.ControlHardwareState	126
A194	B.8.2.2	tms.Fingerprint	126
A195	B.8.2.3	tms.ClockMonotonic	127
A196	B.9	Power Hardware Status	127
A197	B.9.1	Data Flows	127
A198	B.9.2	Data Types	128
A199	B.9.2.1	tms.PowerHardwareState	129
A200	B.9.2.2	tms.Fingerprint	129
A201	B.9.2.3	tms.ClockMonotonic	130
A202	B.9.2.4	tms.EngineState	130
A203	B.9.2.5	tms.FuelState	131
A204	B.9.2.6	tms.GeneratorState	131
A205	B.9.2.7	tms.EnergyStorageState	132
A206	B.9.2.8	tms.PowerElectronicsState	132
A207	B.9.2.9	tms.TransformerState	133

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A208	B.9.2.10 tms.ThermalState . . . . .	133
A209	B.9.2.11 tms.ThermalLoadSequence . . . . .	134
A210	B.10 Storage State . . . . .	134
A211	B.10.1 Data Flows . . . . .	134
A212	B.10.2 Data Types . . . . .	135
A213	B.10.2.1 tms.stor.StorageState . . . . .	136
A214	B.10.2.2 tms.Fingerprint . . . . .	138
A215	B.10.2.3 tms.Curve2D . . . . .	139
A216	B.10.2.4 tms.Point2DSequence . . . . .	140
A217	B.10.2.5 tms.Point2D . . . . .	140
A218	B.10.2.6 tms.stor.StorageCellStateSequence . . . . .	140
A219	B.10.2.7 tms.stor.StorageCellState . . . . .	141
A220	B.10.2.8 tms.stor.StorageCellNumber . . . . .	142
A221	B.11 Operator Input . . . . .	142
A222	B.11.1 Data Flows . . . . .	142
A223	B.11.2 Data Types . . . . .	144
A224	B.11.2.1 tms.OperatorIntentState . . . . .	145
A225	B.11.2.2 tms.OperatorIntentRequest . . . . .	146
A226	B.11.2.3 tms.Reply . . . . .	146
A227	B.11.2.4 tms.Fingerprint . . . . .	146
A228	B.11.2.5 tms.OperatorIntent . . . . .	147
A229	B.11.2.6 tms.GridRequest . . . . .	147
A230	B.11.2.7 tms.OperatorIntentType . . . . .	148
A231	B.11.2.8 tms.MicrogridIntent . . . . .	148
A232	B.11.2.9 tms.OperatingMode . . . . .	149
A233	B.11.2.10 tms.LoadSharingIntent . . . . .	150
A234	B.11.2.11 tms.Duration . . . . .	151
A235	B.11.2.12 tms.OperatorPriorityType . . . . .	151
A236	B.11.2.13 tms.StorageIntent . . . . .	152
A237	B.11.2.14 tms.ClockMonotonic . . . . .	152
A238	B.11.2.15 tms.DeviceFingerprintSequence . . . . .	153
A239	B.11.2.16 tms.DeviceFingerprint . . . . .	153
A240	B.11.2.17 tms.DeviceIntentSequence . . . . .	153
A241	B.11.2.18 tms.DeviceIntent . . . . .	154
A242	B.11.2.19 tms.OperatorPriority . . . . .	154
A243	B.11.2.20 tms.PowerPortIntentSequence . . . . .	155
A244	B.11.2.21 tms.PowerPortIntent . . . . .	155
A245	B.11.2.22 tms.PowerPortNumber . . . . .	155
A246	B.11.2.23 tms.RequestSequence . . . . .	155
A247	B.11.2.24 tms.ConfigId . . . . .	155
A248	B.11.2.25 tms.ReplyStatus . . . . .	155
A249	B.11.2.26 tms.String32 . . . . .	156

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A250	B.12 Diagnostics Reporting	156
A251	B.12.1 Data Flows	156
A252	B.12.2 Data Types	157
A253	B.12.2.1 tms.ActiveDiagnosticMessages	158
A254	B.12.2.2 tms.Fingerprint	158
A255	B.12.2.3 tms.DiagnosticTroubleCodeSequence	158
A256	B.12.2.4 tms.DiagnosticTroubleCode	159
A257	B.12.2.5 tms.SuspectParameterNumber	160
A258	B.12.2.6 tms.FailureModeIndicator	160
A259	B.12.2.7 tms.ClockMonotonic	160
A260	B.12.2.8 tms.DtcSeverity	161
A261	B.12.2.9 tms.String32	161
A262	B.13 Microgrid Controller Selection	161
A263	B.13.1 Data Flows	162
A264	B.13.2 Data Types	162
A265	B.13.2.1 tms.StandardConfigMaster	163
A266	B.13.2.2 tms.Fingerprint	163
A267	B.14 Read Configuration Settings	163
A268	B.14.1 Data Flows	163
A269	B.14.2 Data Types	165
A270	B.14.2.1 tms.GetConfigContentsRequest	165
A271	B.14.2.2 tms.GetConfigContentsResponse	166
A272	B.14.2.3 tms.DeviceConfigRequest	166
A273	B.14.2.4 tms.Fingerprint	166
A274	B.14.2.5 tms.ConfigId	167
A275	B.14.2.6 tms.RequestSequence	167
A276	B.14.2.7 tms.ReplyStatus	167
A277	B.14.2.8 tms.String32	167
A278	B.14.2.9 tms.PowerDeviceConfig	168
A279	B.14.2.10 tms.PowerSwitchCommandSequence	169
A280	B.14.2.11 tms.PowerSwitchCommand	170
A281	B.14.2.12 tms.PowerPortConfigRequest	171
A282	B.14.2.13 tms.PowerPortNumber	171
A283	B.14.2.14 tms.DesiredCircuitContinuity	172
A284	B.14.2.15 tms.SourceTransitionRequest	173
A285	B.14.2.16 tms.SourceTransition	174
A286	B.14.2.17 tms.ControlParameterRequest	175
A287	B.14.2.18 tms.ParameterValueSequence	175
A288	B.14.2.19 tms.ParameterValue	176
A289	B.14.2.20 tms.String1-32	176
A290	B.14.2.21 tms.GroundingCommandSequence	176
A291	B.14.2.22 tms.GroundingCommand	177

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A292	B.14.2.23	tms.GroundingCircuitNumber	177
A293	B.14.2.24	tms.ac.LoadSharingRequest	178
A294	B.14.2.25	tms.ac.LoadSharing	178
A295	B.14.2.26	tms.ControlCurve	179
A296	B.14.2.27	tms.Curve2D	179
A297	B.14.2.28	tms.Point2DSequence	179
A298	B.14.2.29	tms.Point2D	179
A299	B.14.2.30	tms.DLSConfig	180
A300	B.14.2.31	tms.TopicList	181
A301	B.14.2.32	tms.TopicName	181
A302	B.14.2.33	tms.dc.LoadSharingRequest	181
A303	B.14.2.34	tms.dc.LoadSharing	182
A304	B.14.2.35	tms.stor.StorageControlRequest	183
A305	B.14.2.36	tms.stor.StorageControlParameters	184
A306	B.14.2.37	tms.Duration	186
A307	B.14.2.38	tms.stor.StorageOperatingState	186
A308	B.14.2.39	tms.stor.UnderFrequencyStopCharge	186
A309	B.14.2.40	tms.stor.OverFrequencyStopDischarge	187
A310	B.15 Copy Configuration Settings		187
A311	B.15.1	Data Flows	187
A312	B.15.2	Data Types	188
A313	B.15.2.1	tms.CopyConfigRequest	189
A314	B.15.2.2	tms.Reply	189
A315	B.15.2.3	tms.DeviceConfigRequest	189
A316	B.15.2.4	tms.Fingerprint	189
A317	B.15.2.5	tms.ConfigId	189
A318	B.15.2.6	tms.RequestSequence	190
A319	B.15.2.7	tms.PowerPortNumber	190
A320	B.15.2.8	tms.ReplyStatus	190
A321	B.15.2.9	tms.String32	190
A322	B.16 Device Parameters		190
A323	B.16.1	Data Flows	190
A324	B.16.2	Data Types	191
A325	B.16.2.1	tms.ControlParameterStatus	192
A326	B.16.2.2	tms.ControlParameterRequest	192
A327	B.16.2.3	tms.Reply	192
A328	B.16.2.4	tms.MetricParameterStatus	193
A329	B.16.2.5	tms.Fingerprint	193
A330	B.16.2.6	tms.ConfigId	193
A331	B.16.2.7	tms.ParameterValueSequence	193
A332	B.16.2.8	tms.ParameterValue	193
A333	B.16.2.9	tms.String1-32	193

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A334	B.16.2.10	tms.DeviceConfigRequest	193
A335	B.16.2.11	tms.RequestSequence	193
A336	B.16.2.12	tms.PowerPortNumber	194
A337	B.16.2.13	tms.ReplyStatus	194
A338	B.16.2.14	tms.String32	194
A339	B.17	Grounding Circuits	194
A340	B.17.1	Data Flows	194
A341	B.17.2	Data Types	195
A342	B.17.2.1	tms.DeviceGroundingStatus	196
A343	B.17.2.2	tms.GroundingCommand	196
A344	B.17.2.3	tms.Reply	196
A345	B.17.2.4	tms.Fingerprint	196
A346	B.17.2.5	tms.GroundingStatusSequence	196
A347	B.17.2.6	tms.GroundingStatus	197
A348	B.17.2.7	tms.GroundingCircuitNumber	197
A349	B.17.2.8	tms.PowerSwitchStatus	198
A350	B.17.2.9	tms.CircuitContinuity	199
A351	B.17.2.10	tms.PowerSwitchReason	199
A352	B.17.2.11	tms.GroundFaultDetection	200
A353	B.17.2.12	tms.DeviceConfigRequest	200
A354	B.17.2.13	tms.ConfigId	200
A355	B.17.2.14	tms.RequestSequence	200
A356	B.17.2.15	tms.DesiredCircuitContinuity	200
A357	B.17.2.16	tms.PowerPortNumber	200
A358	B.17.2.17	tms.ReplyStatus	201
A359	B.17.2.18	tms.String32	201
A360	B.18	Remote Start/Stop	201
A361	B.18.1	Data Flows	201
A362	B.18.2	Data Types	202
A363	B.18.2.1	tms.SourceTransitionState	203
A364	B.18.2.2	tms.SourceTransitionRequest	203
A365	B.18.2.3	tms.Reply	203
A366	B.18.2.4	tms.Fingerprint	203
A367	B.18.2.5	tms.SourceState	204
A368	B.18.2.6	tms.DeviceConfigRequest	204
A369	B.18.2.7	tms.ConfigId	204
A370	B.18.2.8	tms.RequestSequence	204
A371	B.18.2.9	tms.SourceTransition	204
A372	B.18.2.10	tms.PowerPortNumber	204
A373	B.18.2.11	tms.ReplyStatus	204
A374	B.18.2.12	tms.String32	204
A375	B.19	Power Switches and Connectors	205

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A376	B.19.1 Data Flows . . . . .	205
A377	B.19.2 Data Types . . . . .	206
A378	B.19.2.1 tms.DevicePowerPortStatuses . . . . .	206
A379	B.19.2.2 tms.PowerSwitchCommand . . . . .	207
A380	B.19.2.3 tms.Reply . . . . .	207
A381	B.19.2.4 tms.Fingerprint . . . . .	207
A382	B.19.2.5 tms.PowerPortStatusSequence . . . . .	207
A383	B.19.2.6 tms.PowerPortStatus . . . . .	208
A384	B.19.2.7 tms.PowerPortNumber . . . . .	208
A385	B.19.2.8 tms.CableSenseStatus . . . . .	209
A386	B.19.2.9 tms.PowerPortId . . . . .	210
A387	B.19.2.10 tms.PowerSwitchStatus . . . . .	210
A388	B.19.2.11 tms.CircuitContinuity . . . . .	210
A389	B.19.2.12 tms.PowerSwitchReason . . . . .	210
A390	B.19.2.13 tms.PowerPortConfigRequest . . . . .	210
A391	B.19.2.14 tms.ConfigId . . . . .	210
A392	B.19.2.15 tms.RequestSequence . . . . .	210
A393	B.19.2.16 tms.DesiredCircuitContinuity . . . . .	210
A394	B.19.2.17 tms.ReplyStatus . . . . .	211
A395	B.19.2.18 tms.String32 . . . . .	211
A396	B.20 AC Measurements . . . . .	211
A397	B.20.1 Data Flows . . . . .	211
A398	B.20.2 Data Types . . . . .	212
A399	B.20.2.1 tms.ac.DevicePowerMeasurements . . . . .	213
A400	B.20.2.2 tms.Fingerprint . . . . .	213
A401	B.20.2.3 tms.ClockMonotonic . . . . .	213
A402	B.20.2.4 tms.ac.PowerPortMeasurementSequence . . . . .	213
A403	B.20.2.5 tms.ac.PowerPortMeasurement . . . . .	214
A404	B.20.2.6 tms.PowerPortNumber . . . . .	214
A405	B.20.2.7 tms.ac.PowerMeasurement . . . . .	214
A406	B.20.2.8 tms.ac.PowerMeasurementLineSequence . . . . .	216
A407	B.20.2.9 tms.ac.PowerMeasurementLine . . . . .	216
A408	B.20.2.10 tms.ac.PhaseAnglesSequence . . . . .	219
A409	B.21 DC Measurements . . . . .	219
A410	B.21.1 Data Flows . . . . .	219
A411	B.21.2 Data Types . . . . .	220
A412	B.21.2.1 tms.dc.DevicePowerMeasurements . . . . .	220
A413	B.21.2.2 tms.Fingerprint . . . . .	220
A414	B.21.2.3 tms.ClockMonotonic . . . . .	221
A415	B.21.2.4 tms.dc.PowerPortMeasurementSequence . . . . .	221
A416	B.21.2.5 tms.dc.PowerPortMeasurement . . . . .	221
A417	B.21.2.6 tms.PowerPortNumber . . . . .	221



*TMS Data Model*  
*DRAFT 19 Apr 2021*

A418	B.21.2.7 tms.dc.PowerMeasurement . . . . .	222
A419	B.21.2.8 tms.CircuitContinuity . . . . .	222
A420	B.22 Power Topology . . . . .	222
A421	B.22.1 Data Flows . . . . .	222
A422	B.22.2 Data Types . . . . .	224
A423	B.22.2.1 tms.PowerConnectionList . . . . .	224
A424	B.22.2.2 tms.Fingerprint . . . . .	224
A425	B.22.2.3 tms.PowerConnectionSequence . . . . .	225
A426	B.22.2.4 tms.PowerConnection . . . . .	225
A427	B.22.2.5 tms.PowerConnectionId . . . . .	226
A428	B.22.2.6 tms.PowerPortId . . . . .	226
A429	B.22.2.7 tms.PowerPortNumber . . . . .	226
A430	B.22.2.8 tms.PowerConnectionDetectionType . . . . .	227
A431	B.23 AC Load Sharing . . . . .	227
A432	B.23.1 Data Flows . . . . .	227
A433	B.23.2 Data Types . . . . .	228
A434	B.23.2.1 tms.ac.LoadSharingStatus . . . . .	229
A435	B.23.2.2 tms.ac.LoadSharingRequest . . . . .	229
A436	B.23.2.3 tms.Reply . . . . .	229
A437	B.23.2.4 tms.Fingerprint . . . . .	229
A438	B.23.2.5 tms.ConfigId . . . . .	229
A439	B.23.2.6 tms.ac.LoadSharing . . . . .	229
A440	B.23.2.7 tms.ControlCurve . . . . .	230
A441	B.23.2.8 tms.Curve2D . . . . .	230
A442	B.23.2.9 tms.Point2DSequence . . . . .	230
A443	B.23.2.10 tms.Point2D . . . . .	230
A444	B.23.2.11 tms.DLSConfig . . . . .	230
A445	B.23.2.12 tms.TopicList . . . . .	230
A446	B.23.2.13 tms.TopicName . . . . .	230
A447	B.23.2.14 tms.PowerPortConfigRequest . . . . .	230
A448	B.23.2.15 tms.PowerPortNumber . . . . .	230
A449	B.23.2.16 tms.RequestSequence . . . . .	230
A450	B.23.2.17 tms.ReplyStatus . . . . .	230
A451	B.23.2.18 tms.String32 . . . . .	230
A452	B.24 DC Load Sharing . . . . .	231
A453	B.24.1 Data Flows . . . . .	231
A454	B.24.2 Data Types . . . . .	232
A455	B.24.2.1 tms.dc.LoadSharingStatus . . . . .	232
A456	B.24.2.2 tms.dc.LoadSharingRequest . . . . .	233
A457	B.24.2.3 tms.Reply . . . . .	233
A458	B.24.2.4 tms.Fingerprint . . . . .	233
A459	B.24.2.5 tms.ConfigId . . . . .	233

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A460	B.24.2.6 tms.dc.LoadSharing . . . . .	233
A461	B.24.2.7 tms.ControlCurve . . . . .	233
A462	B.24.2.8 tms.Curve2D . . . . .	233
A463	B.24.2.9 tms.Point2DSequence . . . . .	233
A464	B.24.2.10tms.Point2D . . . . .	233
A465	B.24.2.11tms.PowerPortConfigRequest . . . . .	233
A466	B.24.2.12tms.PowerPortNumber . . . . .	233
A467	B.24.2.13tms.RequestSequence . . . . .	233
A468	B.24.2.14tms.ReplyStatus . . . . .	234
A469	B.24.2.15tms.String32 . . . . .	234
A470	B.25 Load Policies . . . . .	234
A471	B.26 Clock Status . . . . .	234
A472	B.26.1 Data Flows . . . . .	234
A473	B.26.2 Data Types . . . . .	235
A474	B.26.2.1 tms.ClockStatus . . . . .	236
A475	B.26.2.2 tms.Fingerprint . . . . .	237
A476	B.26.2.3 tms.ClockMonotonic . . . . .	237
A477	B.26.2.4 tms.ClockMonotonicPersistenceType . . . . .	238
A478	B.26.2.5 tms.ClockRealtime . . . . .	238
A479	B.27 Black Start . . . . .	239
A480	B.27.1 Data Flows . . . . .	239
A481	B.27.2 Data Types . . . . .	240
A482	B.27.2.1 tms.AuthorizationToEnergizeOutcome . . . . .	241
A483	B.27.2.2 tms.AuthorizationToEnergizeRequest . . . . .	243
A484	B.27.2.3 tms.AuthorizationToEnergizeResponse . . . . .	244
A485	B.27.2.4 tms.DeviceRequest . . . . .	245
A486	B.27.2.5 tms.Fingerprint . . . . .	245
A487	B.27.2.6 tms.ClockMonotonic . . . . .	245
A488	B.27.2.7 tms.ClockRealtime . . . . .	245
A489	B.27.2.8 tms.GridRequest . . . . .	245
A490	B.27.2.9 tms.RequestSequence . . . . .	245
A491	B.27.2.10tms.PowerPortNumberSequence . . . . .	245
A492	B.27.2.11tms.PowerPortNumber . . . . .	245
A493	B.28 Microgrid Membership . . . . .	245
A494	B.28.1 Data Flows . . . . .	246
A495	B.28.2 Data Types . . . . .	248
A496	B.28.2.1 tms.MicrogridMembershipApproval . . . . .	249
A497	B.28.2.2 tms.MicrogridMembershipRequest . . . . .	250
A498	B.28.2.3 tms.Reply . . . . .	250
A499	B.28.2.4 tms.GridRequest . . . . .	250
A500	B.28.2.5 tms.Fingerprint . . . . .	250
A501	B.28.2.6 tms.RequestSequence . . . . .	250

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A502	B.28.2.7 tms.MicrogridMembership . . . . .	251
A503	B.28.2.8 tms.MicrogridMembershipResult . . . . .	251
A504	B.28.2.9 tms.String32 . . . . .	251
A505	B.28.2.10tms.ConfigId . . . . .	251
A506	B.28.2.11tms.PowerPortNumber . . . . .	252
A507	B.28.2.12tms.ReplyStatus . . . . .	252
A508	B.29 Energy Storage Control . . . . .	252
A509	B.29.1 Data Flows . . . . .	252
A510	B.29.2 Data Types . . . . .	253
A511	B.29.2.1 tms.stor.StorageControlStatus . . . . .	253
A512	B.29.2.2 tms.stor.StorageControlRequest . . . . .	254
A513	B.29.2.3 tms.Reply . . . . .	254
A514	B.29.2.4 tms.Fingerprint . . . . .	254
A515	B.29.2.5 tms.ConfigId . . . . .	254
A516	B.29.2.6 tms.stor.StorageControlParameters . . . . .	254
A517	B.29.2.7 tms.Duration . . . . .	254
A518	B.29.2.8 tms.stor.StorageOperatingState . . . . .	254
A519	B.29.2.9 tms.stor.UnderFrequencyStopCharge . . . . .	254
A520	B.29.2.10tms.stor.OverFrequencyStopDischarge . . . . .	254
A521	B.29.2.11tms.DeviceConfigRequest . . . . .	254
A522	B.29.2.12tms.RequestSequence . . . . .	254
A523	B.29.2.13tms.PowerPortNumber . . . . .	254
A524	B.29.2.14tms.ReplyStatus . . . . .	255
A525	B.29.2.15tms.String32 . . . . .	255
A526	B.30 Cross Reference . . . . .	256
A527	B.30.1 Data Types in All Namespaces . . . . .	256
A528	B.30.2 Data Types in Namespace tms . . . . .	258
A529	B.30.3 Data Types in Namespace tms.ac . . . . .	261
A530	B.30.4 Data Types in Namespace tms.dc . . . . .	261
A531	B.30.5 Defined Constants . . . . .	262
A532	<b>C Cybersecurity Requirements</b>	<b>263</b>
A533	<b>D OMG DDS Requirements</b>	<b>264</b>
A534	D.1 Domains . . . . .	264
A535	D.2 Topics . . . . .	264
A536	D.3 IDL . . . . .	265
A537	D.3.1 Data Types . . . . .	265
A538	D.3.2 Annotations . . . . .	266
A539	D.3.2.1 Extensibility . . . . .	266
A540	D.3.2.2 Topic Usage . . . . .	266
A541	D.3.2.3 Key Values . . . . .	266

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A542	D.3.2.4 Bounded Values . . . . .	267
A543	D.3.2.5 Units of Measure . . . . .	267
A544	D.4 Quality of Service . . . . .	267
A545	D.4.1 PublishLast Profile . . . . .	268
A546	D.4.2 Rare Profile . . . . .	270
A547	D.4.3 Slow Profile . . . . .	272
A548	D.4.4 Medium Profile . . . . .	274
A549	D.4.5 Continuous Profile . . . . .	276
A550	D.4.6 Command Profile . . . . .	278
A551	D.4.7 Response Profile . . . . .	280
A552	D.5 DDS-Security . . . . .	281
A553	<b>E Machine Readable Files</b>	<b>282</b>
A554	E.1 IDL . . . . .	282
A555	E.2 Quality of Service XML . . . . .	340

A556 List of Figures

A557	A.1 TMS Device Roles . . . . .	25
A558	A.2 Overview of the request / reply / status communications pattern. . . . .	27
A559	A.3 Data Flow Notation . . . . .	29
A560	A.4 Status Data Flow Pattern Example . . . . .	31
A561	A.5 Request/Response Data Flow Pattern Example . . . . .	31
A562	A.6 Request/Response with Status Data Flow Pattern Example . . . . .	32
A563	A.7 Example timing for a periodic 1 s topic. . . . .	36
A564	A.8 Example publisher send windows. . . . .	37
A565	A.9 Example subscriber receive windows. . . . .	37
A566	A.10 Example Encoding of a Sequence as a Structure with Constants . . . . .	51
A567	B.1 Schematic diagram of the Source device role. . . . .	59
A568	B.2 Schematic diagram of the Distribution device role. . . . .	62
A569	B.3 Schematic diagram of the Storage device role. . . . .	63
A570	B.4 Schematic diagram of the Load device role. . . . .	63
A571	B.5 Schematic diagram of the Conversion device role. . . . .	64
A572	B.6 Request/Response Pattern data flow . . . . .	65
A573	B.7 Heartbeat data flow . . . . .	73
A574	B.8 Device Announcement data flow . . . . .	75
A575	B.9 Nickname data flow . . . . .	117
A576	B.10 Device Icon data flow . . . . .	122
A577	B.11 Control Hardware Status data flow . . . . .	124
A578	B.12 Power Hardware Status data flow . . . . .	127
A579	B.13 Storage State data flow . . . . .	134
A580	B.14 Example <b>Curve2D</b> with three points. . . . .	139
A581	B.15 Operator Input data flow . . . . .	142
A582	B.16 Diagnostics Reporting data flow . . . . .	156
A583	B.17 Microgrid Controller Selection data flow . . . . .	162
A584	B.18 Read Configuration Settings data flow . . . . .	164
A585	B.19 Copy Configuration Settings data flow . . . . .	187
A586	B.20 Device Parameters data flow . . . . .	190
A587	B.21 Device Parameters data flow . . . . .	190
A588	B.22 Grounding Circuits data flow . . . . .	194
A589	B.23 Remote Start/Stop data flow . . . . .	201

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A590	B.24 Power Switches and Connectors data flow . . . . .	205
A591	B.25 AC Measurements data flow . . . . .	211
A592	B.26 Phase Angles for a 208/120V Wye System . . . . .	215
A593	B.27 Power Factor Lead Angle . . . . .	217
A594	B.28 Power Factor Lag Angle . . . . .	218
A595	B.29 DC Measurements data flow . . . . .	219
A596	B.30 Power Topology data flow . . . . .	222
A597	B.31 Power Topology data flow . . . . .	223
A598	B.32 Power Topology data flow . . . . .	223
A599	B.33 AC Load Sharing data flow . . . . .	227
A600	B.34 DC Load Sharing data flow . . . . .	231
A601	B.35 Clock Status data flow . . . . .	234
A602	B.36 Black Start data flow . . . . .	239
A603	B.37 Microgrid Membership data flow . . . . .	246
A604	B.38 Energy Storage Control data flow . . . . .	252

A605 List of Tables

A606	A.1 Times for microgrid consensus. . . . .	28
A607	A.2 Overview of all data types used as key values. . . . .	33
A608	A.3 Specifications for each rate class. . . . .	38
A609	A.4 Relationships between data triggers and rate classes. . . . .	39
A610	A.5 Namespaces defined by TMS. . . . .	44
A611	A.6 Units of Measure . . . . .	45
A612	A.7 Overview of Primitive Data Types and their General Usage . . . . .	46
A613	A.8 UCS Portable Character Set . . . . .	47
A614	A.9 Required Integer Types . . . . .	48
A615	A.10 Naming convention for pre-defined length constants. . . . .	49
A616	A.11 Default Values for all Data Types . . . . .	53
A617	A.12 Appendable Extensibility for all Data Types . . . . .	54
A618	A.13 Example quantities for a small microgrid. . . . .	55
A619	B.1 Overview of All Data Model Topics . . . . .	58
A620	B.2 Data Model Sections for Source Role Components . . . . .	61
A621	B.3 Description of the Request/Response Pattern topics. . . . .	66
A622	B.4 Participants of the Request/Response Pattern topics. . . . .	66
A623	B.5 Timing of the Request/Response Pattern topics. . . . .	66
A624	B.6 Data types of the Request/Response Pattern topics. . . . .	66
A625	B.7 Description of the Heartbeat topics. . . . .	73
A626	B.8 Participants of the Heartbeat topics. . . . .	73
A627	B.9 Timing of the Heartbeat topics. . . . .	73
A628	B.10 Data types of the Heartbeat topics. . . . .	74
A629	B.11 Description of the Device Announcement topics. . . . .	75
A630	B.12 Participants of the Device Announcement topics. . . . .	75
A631	B.13 Timing of the Device Announcement topics. . . . .	76
A632	B.14 Data types of the Device Announcement topics. . . . .	76
A633	B.15 Description of the Nickname topics. . . . .	118
A634	B.16 Participants of the Nickname topics. . . . .	118
A635	B.17 Timing of the Nickname topics. . . . .	118
A636	B.18 Data types of the Nickname topics. . . . .	119
A637	B.19 Description of the Device Icon topics. . . . .	122
A638	B.20 Participants of the Device Icon topics. . . . .	122

A639	B.21 Timing of the Device Icon topics. . . . .	122
A640	B.22 Data types of the Device Icon topics. . . . .	122
A641	B.23 Description of the Control Hardware Status topics. . . . .	124
A642	B.24 Participants of the Control Hardware Status topics. . . . .	124
A643	B.25 Timing of the Control Hardware Status topics. . . . .	124
A644	B.26 Data types of the Control Hardware Status topics. . . . .	125
A645	B.27 Description of the Power Hardware Status topics. . . . .	127
A646	B.28 Participants of the Power Hardware Status topics. . . . .	128
A647	B.29 Timing of the Power Hardware Status topics. . . . .	128
A648	B.30 Data types of the Power Hardware Status topics. . . . .	128
A649	B.31 Description of the Storage State topics. . . . .	134
A650	B.32 Participants of the Storage State topics. . . . .	134
A651	B.33 Timing of the Storage State topics. . . . .	134
A652	B.34 Data types of the Storage State topics. . . . .	135
A653	B.35 Description of the Operator Input topics. . . . .	143
A654	B.36 Participants of the Operator Input topics. . . . .	144
A655	B.37 Timing of the Operator Input topics. . . . .	144
A656	B.38 Data types of the Operator Input topics. . . . .	144
A657	B.39 Description of the Diagnostics Reporting topics. . . . .	156
A658	B.40 Participants of the Diagnostics Reporting topics. . . . .	157
A659	B.41 Timing of the Diagnostics Reporting topics. . . . .	157
A660	B.42 Data types of the Diagnostics Reporting topics. . . . .	157
A661	B.43 Description of the Microgrid Controller Selection topics. . . . .	162
A662	B.44 Participants of the Microgrid Controller Selection topics. . . . .	162
A663	B.45 Timing of the Microgrid Controller Selection topics. . . . .	162
A664	B.46 Data types of the Microgrid Controller Selection topics. . . . .	162
A665	B.47 Description of the Read Configuration Settings topics. . . . .	164
A666	B.48 Participants of the Read Configuration Settings topics. . . . .	164
A667	B.49 Timing of the Read Configuration Settings topics. . . . .	164
A668	B.50 Data types of the Read Configuration Settings topics. . . . .	165
A669	B.51 Description of the Copy Configuration Settings topics. . . . .	188
A670	B.52 Participants of the Copy Configuration Settings topics. . . . .	188
A671	B.53 Timing of the Copy Configuration Settings topics. . . . .	188
A672	B.54 Data types of the Copy Configuration Settings topics. . . . .	188
A673	B.55 Description of the Device Parameters topics. . . . .	191
A674	B.56 Participants of the Device Parameters topics. . . . .	191
A675	B.57 Timing of the Device Parameters topics. . . . .	191
A676	B.58 Data types of the Device Parameters topics. . . . .	192
A677	B.59 Description of the Grounding Circuits topics. . . . .	194
A678	B.60 Participants of the Grounding Circuits topics. . . . .	195
A679	B.61 Timing of the Grounding Circuits topics. . . . .	195
A680	B.62 Data types of the Grounding Circuits topics. . . . .	195



A681	B.63 Description of the Remote Start/Stop topics. . . . .	201
A682	B.64 Participants of the Remote Start/Stop topics. . . . .	201
A683	B.65 Timing of the Remote Start/Stop topics. . . . .	202
A684	B.66 Data types of the Remote Start/Stop topics. . . . .	202
A685	B.67 Description of the Power Switches and Connectors topics. . . . .	205
A686	B.68 Participants of the Power Switches and Connectors topics. . . . .	205
A687	B.69 Timing of the Power Switches and Connectors topics. . . . .	206
A688	B.70 Data types of the Power Switches and Connectors topics. . . . .	206
A689	B.71 Description of the AC Measurements topics. . . . .	211
A690	B.72 Participants of the AC Measurements topics. . . . .	212
A691	B.73 Timing of the AC Measurements topics. . . . .	212
A692	B.74 Data types of the AC Measurements topics. . . . .	212
A693	B.75 Voltage, current, and angle measurements for PHASE_SINGLE. . . . .	215
A694	B.76 Voltage, current, and angle measurements for PHASE_SPLIT. . . . .	215
A695	B.77 Voltage, current, and angle measurements for PHASE_3DELTA. . . . .	215
A696	B.78 Voltage, current, and angle measurements for PHASE_3WYE. . . . .	215
A697	B.79 Description of the DC Measurements topics. . . . .	219
A698	B.80 Participants of the DC Measurements topics. . . . .	219
A699	B.81 Timing of the DC Measurements topics. . . . .	220
A700	B.82 Data types of the DC Measurements topics. . . . .	220
A701	B.83 Description of the Power Topology topics. . . . .	223
A702	B.84 Participants of the Power Topology topics. . . . .	223
A703	B.85 Timing of the Power Topology topics. . . . .	224
A704	B.86 Data types of the Power Topology topics. . . . .	224
A705	B.87 Description of the AC Load Sharing topics. . . . .	228
A706	B.88 Participants of the AC Load Sharing topics. . . . .	228
A707	B.89 Timing of the AC Load Sharing topics. . . . .	228
A708	B.90 Data types of the AC Load Sharing topics. . . . .	228
A709	B.91 Description of the DC Load Sharing topics. . . . .	231
A710	B.92 Participants of the DC Load Sharing topics. . . . .	231
A711	B.93 Timing of the DC Load Sharing topics. . . . .	232
A712	B.94 Data types of the DC Load Sharing topics. . . . .	232
A713	B.95 Description of the Clock Status topics. . . . .	234
A714	B.96 Participants of the Clock Status topics. . . . .	234
A715	B.97 Timing of the Clock Status topics. . . . .	235
A716	B.98 Data types of the Clock Status topics. . . . .	235
A717	B.99 Description of the Black Start topics. . . . .	239
A718	B.100 Participants of the Black Start topics. . . . .	240
A719	B.101 Timing of the Black Start topics. . . . .	240
A720	B.102 Data types of the Black Start topics. . . . .	240
A721	B.103 Description of the Microgrid Membership topics. . . . .	247
A722	B.104 Participants of the Microgrid Membership topics. . . . .	247

A723	B.105	Timing of the Microgrid Membership topics. . . . .	248
A724	B.106	Data types of the Microgrid Membership topics. . . . .	248
A725	B.107	Description of the Energy Storage Control topics. . . . .	252
A726	B.108	Participants of the Energy Storage Control topics. . . . .	252
A727	B.109	Timing of the Energy Storage Control topics. . . . .	253
A728	B.110	Data types of the Energy Storage Control topics. . . . .	253
A729	D.1	Required DDS Standards . . . . .	264
A730	D.2	DDS Security Standards . . . . .	264
A731	D.3	IDL Type Substitutions . . . . .	265
A732	D.4	IDL Annodations for Extensibility . . . . .	266
A733	D.5	IDL Annotations for Topic Usage . . . . .	266
A734	D.6	IDL Annotations for Bounded Values . . . . .	267
A735	D.7	OMG DDS QoS Policies for the PublishLast Profile . . . . .	268
A736	D.8	OMG DDS QoS Policies for the Rare Profile . . . . .	270
A737	D.9	OMG DDS QoS Policies for the Slow Profile . . . . .	272
A738	D.10	OMG DDS QoS Policies for the Medium Profile . . . . .	274
A739	D.11	OMG DDS QoS Policies for the Continuous Profile . . . . .	276
A740	D.12	OMG DDS QoS Policies for the Command Profile . . . . .	278
A741	D.13	OMG DDS QoS Policies for the Response Profile . . . . .	280

## A. General Data Model Requirements

### A.1 Introduction

The TMS defines an *architecture* for building microgrid *systems*. Each microgrid system defines one or more *devices* that connect together. Each device implements a single *role* that provides a modular unit of functionality. Each role has defined interfaces that support both standalone and coordinated behaviors. Multiple devices, implementing the same or different roles, may be packaged together into a single *platform*. This architecture is designed to enable interoperability between devices and platforms from all systems, subject to power constraints.

Individual devices perform essential TMS functions using only internal inputs and outputs. System-level TMS performance may be enhanced by sending coordination messages between devices. The TMS *data model* defines the language for these messages.

The data model is defined in terms of *data flows* and *data types*. Data flows define the motion of messages while data types define the contents of messages.

This Appendix documents the TMS data model in several sections.

- Chapter [A](#) defines requirements for the general-purpose communications infrastructure.
- Chapter [B](#) defines requirements for domain-specific messages.
- Chapter [C](#) defines requirements for cybersecurity.
- Chapter [D](#) defines how OMG DDS is used to meet these requirements. TMS device software may be roughly categorized into business logic and network logic. The OMG DDS middleware encapsulates most of the network logic, thus allowing software re-use and simplifying the business logic.
- Chapter [E](#) provides machine-readable definitions of the data model.

### A.2 Scope and Purpose

This document defines the TMS data model, including all messages sent between TMS devices. These messages are used to communicate all system health, status, command, and control information. They are also recommended for communications with external networks.

The data model provides a common language used by all TMS devices. It is platform-independent to allow flexibility in device design and implementation. It contains detailed data flow and data type definitions to guarantee interoperability between devices, regardless of make or mode.

A773 A.3 Implementation Considerations

A774 The data model allows for design and implementation flexibility while still providing  
A775 interoperability. This section identifies high-level objectives for any TMS design and  
A776 implementation plan. These points should be carefully considered in all decisions.

A777 **Security** The design should ensure safe and trustworthy operations of the TMS by  
A778 promoting the availability, integrity, and positive control of microgrid resources. It should  
A779 remain resilient in the face of device compromise and hamper the spread of malware infection.

A780 Many security properties must be integrated to provide seamless operation,  
A781 interoperability, and a high level of resilience. The root of trust must include cryptographic  
A782 identities for hardware, software, and users. Data flows must be enforced with authentication,  
A783 authorization, and integrity protection. Logging must provide a high-integrity, trustworthy  
A784 forensic record of events from production through operation.

A785 **Performance** The design should support the desired functionality in an easily-proven  
A786 manner. It should not unduly burden either operator or equipment. It should be robust  
A787 against potential faults and support graceful degradation. It should support reliability,  
A788 availability, and maintainability.

A789 **Simplicity** The design should be simple to understand and implement. Usage patterns  
A790 should be predictable and consistent. Each part of the data model trades increased  
A791 functionality for increased overall complexity. Simpler solutions improve this trade-off.

A792 **Interoperability** The design should be clear and unambiguous. Standard interaction  
A793 patterns should be well defined. Incompatible dialects should be avoided by reducing options.  
A794 In the ideal case, any two devices, developed in isolation and only linked by this standard,  
A795 will work together the first time they are connected.

A796 **Extensibility** The design should be flexible and support a range of possible solutions.  
A797 Room should be left for innovation and product differentiation. The design should provide  
A798 mechanism, without enforcing too much policy. Options may be left in places where there are  
A799 compelling alternatives. Well-chosen constraints increase extensibility by providing system-  
A800 level guarantees.

A801 **Verification and Validation** Validation artifacts should demonstrate that all design  
A802 thresholds and objectives were met. Verification tests should demonstrate conformance of  
A803 devices to the TMS data model. The design should support clear and rigorous verification  
A804 and validation. Both verification and validation results should provide an accurate prediction  
A805 of operational performance in the field and interoperability with other possible device  
A806 variations.

#### A807 A.4 General Requirements

A808 The general requirements discusses the TMS data model architecture and the basic structure  
A809 of communication between TMS devices. The data flows and data types defined in this  
A810 standard identify the basic communication requirements for TMS.

A811 This chapter defines how the data model is constructed. Chapter [B](#) defines the data  
A812 model at a conceptual level. Chapter [D](#) provides the low-level details of how the data model  
A813 is implemented.

#### A814 A.5 Device Identity

A815 In the TMS data model, every device has a single identity bound to a single, specific device  
A816 role that has a number of built-in requirements and permissions. This approach simplifies  
A817 the communications by preventing any ambiguity about how roles may interact.

A818 Under all circumstances, every TMS device role shall have its own unique **Fingerprint**.  
A819 Since there will likely be multiple devices assigned to the same role, such as generators, the  
A820 **Fingerprint** data type assigns a unique identity to each device, irregardless of the device  
A821 role. Data Types, such as **Fingerprint**, are discussed in Chapter [B](#).

#### A822 A.6 Device Roles

A823 *Device roles* are defined building blocks within the TMS that provide modularized  
A824 functionality. Each role is assigned based upon the device's capabilities as defined in  
A825 the TMS. Each device role is pre-defined in both functionality and power/communication  
A826 interfaces to provide modular flexibility.

A827 Pre-defined components and interfaces allow customizable modularity in each TMS  
A828 system to meet the unique needs of each microgrid installation. By using the device roles to  
A829 define specific functionalities in the microgrid, any number of devices, regardless of capacity  
A830 size, can be procured and assembled into a single TMS that meets the mission needs.

A831 There are two types of device roles in TMS: Control Services and Power Devices.  
A832 Figure [A.1](#) shows the relationship between these roles.

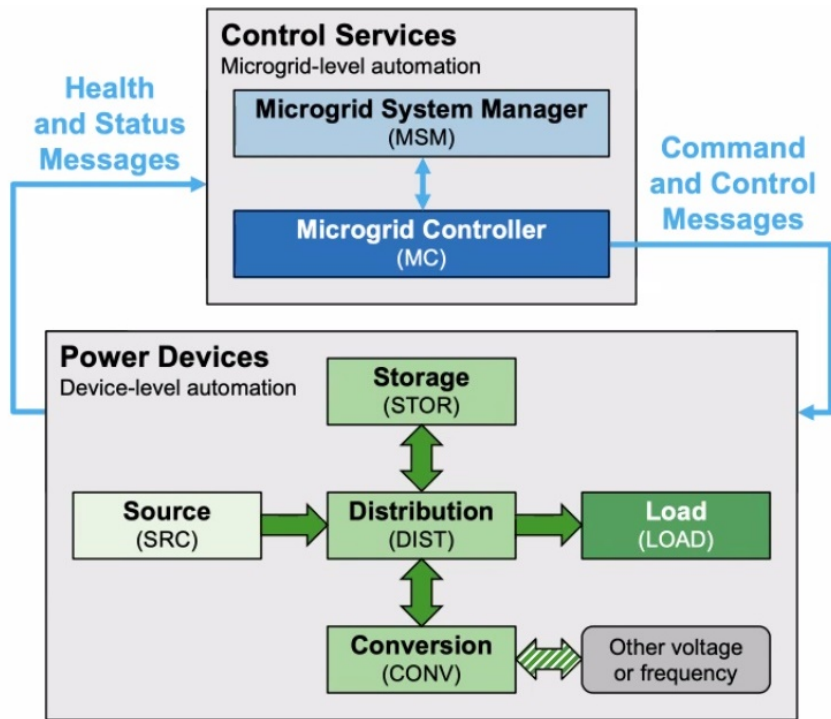


Figure A.1: TMS Device Roles

#### A.6.1 Control Services

Control Services device roles embody functionality that is not specific to a particular type of power hardware. They provide mostly microgrid-level functionality and control the message and command flow.

The Control Services roles include:

- **Microgrid Controllers (MC)** are authorized to send configuration settings and commands to other TMS devices. This represents the automated control within the TMS network.
- **Microgrid Dashboard (MD)** represents the user interface (GUI) and the interactions with operators or users.

#### A.6.2 Power Devices

Power Devices roles are specific to particular types of power hardware. They provide mostly device-level functionality and include everything that uses, generates, or stores power.

The Power Device roles include:

- **Source (SRC)** devices provide power to the microgrid and include diesel gensets, solar panels, and wind turbines. Note that while an energy storage unit (ESU) can act as a source, TMS considers ESUs exclusively as Storage device roles.

- **Distribution (DIST)** devices connect other power devices together and commonly provide metering and switching capabilities.
- **Storage (STOR)** devices are charged or discharged with energy from the microgrid to store or consume power within the microgrid. Storage devices include batteries and flywheels.
- **Load (LOAD)** devices consume power from the microgrid. The purpose of the microgrid is to power load devices. As only loads with active control and status capabilities can operate as TMS devices, most loads will not implement TMS.
- **Conversion (CONV)** devices modify the input power to output a different frequency or voltage for other devices. Conversion devices include inverters, rectifiers, and transformers.

More details on the power device roles, including schematics of their major hardware components, are provided in Section [B.1](#).

### A.6.3 Platforms and Multi-Role Products

While every device in the TMS data model has a unique identity and fills a single, specific device role to simplify communications and prevent ambiguity about role interactions, there are cases where grouping multiple TMS devices into a single physical or virtual container is desirable. The reasons for such groupings are typically cost, size, shipping, and presentation.

Examples of platforms and multi-role products include:

- Multiple devices that are mounted together on a larger *platform*, such as:
  - A vehicle with separate SRC, DIST, and MC devices mounted on the chassis.
  - A building with LOAD, DIST, and backup SRC equipment.
- Products that package multiple device roles into a single *multi-role product*, such as:
  - A generator with a built-in microgrid controller has both SRC and MC device roles.
  - A control computer incorporating MC and MD device roles.
  - A simulation computer with separate identities for each simulated device role.
- Products that have one device role but may be split into multiple identities of the device role, such as:
  - A large DIST device with the power ports split into multiple DIST device role identities.

The data model represents all such relationships using the `ProductInfo.platformId` data type, described in detail in Chapter B. The `platformId` for each device points to its associated platform or multi-role product.

Each device on a platform shall have a separate identity and role. Multi-role products shall have separate identities and device roles for each type of functionality. Multi-role products shall communicate as if they were multiple physically-separate devices. Products with a built-in MC shall allow the internal MC to be disabled so that it can accept commands from an external MC.

It is recommended that each:

- Device role has its own data connection and runs on a separate processor so each role can be physically isolated.
- TMS power device has a minimum of two data connections, one for networking with other TMS devices and one for maintenance access.
- TMS DIST device has a network switch with one connector for each power port so every TMS device that connects to power can also connect to data.

If multiple devices are packaged into a single platform or multi-role product, they may share hardware or software components. If such sharing does occur, then all messages shall be sent separately for each contained device identity and role. Messages may be received singly and re-shared using internal mechanisms.

## A.7 Consensus

Consensus is the mechanism whereby devices agree on the desired state of the microgrid.

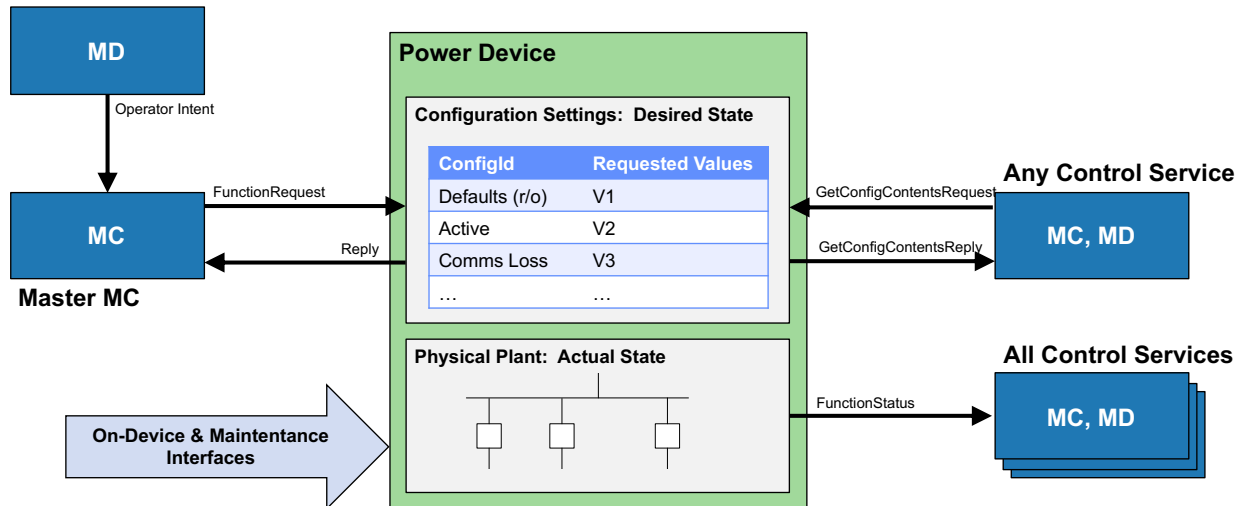


Figure A.2: Overview of the request / reply / status communications pattern.

Each power device independently reports its state to all MCs. This state includes internal device status, configuration settings, and the `StandardConfigMaster`. MCs can confirm settings by sending a `GetConfigContentsRequest`. See Section B.14 for more information.



A905 Each power device independently selects which MC will be its master. This selection  
A906 is based on communication liveliness and MC prioritization. This selection is triggered by  
A907 communication events and it is made after a timeout has completed. During these timeouts,  
A908 the device continues accepting commands from the MC it was following before the trigger  
A909 occurred.

A910 When a new MC starts communicating, a device shall select a master MC after the  
A911 new-MC delay of 3s. When the master MC stops communicating, a device shall select a  
A912 master MC after the lost-MC delay of 6s. When a non-master MC stops communicating, no  
A913 selection is scheduled.

A914 If another MC starts communicating while any selection timeout is active, then the device  
A915 must act as if the selection had been made at the scheduled time.

A916 Whenever a selection of master MC is made, the highest-priority MC that has  
A917 communicated in the last 1s shall be chosen as the master MC.

A918 When there has been no communication from any MC in 10s, a power device shall declare  
A919 communications loss and enter the **NO\_COMM** configuration.

Table A.1: Times for microgrid consensus.

Timer Name	Interval	Timeout
Normal Heartbeat	1 s	-
New-MC Delay	-	3 s
Lost-MC Delay	-	6 s
Power Device Comms Loss Delay	-	10 s

## A.8 Data Flows

TMS devices shall communicate through pre-defined *data flows* to share all operating information such as power commands and status. Every data flow contains one or more *topics*. Each topic is defined with a name, *publishers* that send data, *subscribers* that receive data, a *data type* that specifies what values can be transferred, and *timing* that specifies when data is transferred.

This publish/subscribe (pub/sub) paradigm defines the data flows in terms of logical connections. The underlying software and hardware may use buffers, filters, and a mix of broadcast, multicast, and point-to-point connections to implement the flow. These details are specified by the OMG DDS standard as used by TMS.

Each TMS data flow defines a small group of messages that have closely related values. Multiple data flows are used together in larger patterns to achieve system-level TMS behaviors. For example, the discovery flow enables all the status and control flows. For another example, power measurements may exceed a threshold and trigger a corrective command.

This section defines the generic data flow concepts that are used for the TMS data model. The domain-specific data flows are defined in Chapter B. The OMG DDS data flow settings are defined in Chapter D.

Data flows are illustrated using the notation provided in Figure A.3, with identifiers pointing to the topic name, publisher, subscriber, and timing information. Note that in the data flow diagrams, Power\* refers to any Power Device Role (SRC, DIST, STOR, LOAD, CONV) and Control\* refers to any Control Service Device Role (MC, MSM).

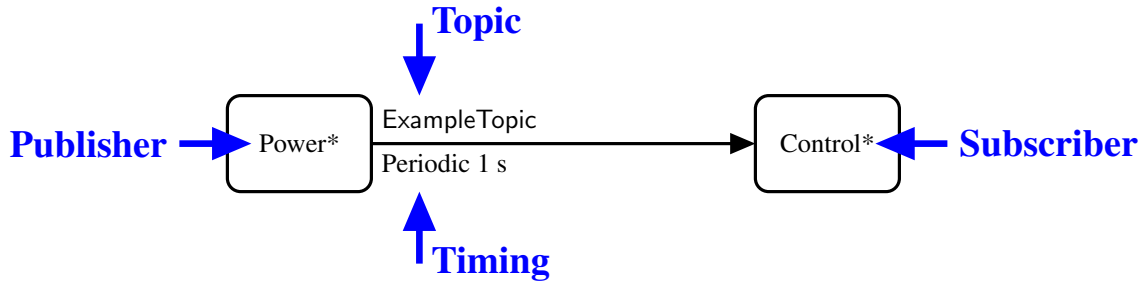


Figure A.3: Data Flow Notation

This figure shows a data flow containing a single topic. Each topic is labeled with the following features.

- **Publisher** is the Device Role(s) that sends these messages.
- **Subscriber** is the Device Role(s) that receives these messages.
- **Topic** is the name/identifier for these messages and the message flow.
- **Timing** is the condition that causes messages to be sent and the reliability of the message flow.

### A949 A.8.1 Publishers and Subscribers

A950 Device Roles are used to specify the publisher, or source of the message, and the subscriber,  
A951 or receiver of the message, for each data flow. While a device role can be both a publisher  
A952 and a subscriber, typically in an individual flow it is one or the other. Conceptually, all  
A953 publishers send all messages to all subscribers on a topic.

A954 For simplicity, the device roles may be shown in the data flows throughout this document  
A955 with abbreviated names - either Power\* for the Power Device role or Control\* for the  
A956 Control Service device role - to represent any device in that category. Refer to [A.6](#) for  
A957 more information on device roles.

### A958 A.8.2 Topic and Message Flow

A959 Topics are used to identify the data flows. Topics define a small group of TMS data flow  
A960 message topics that have closely related values and match the publishers to the subscribers  
A961 for each message chain. Multiple data flow topics are used together in larger patterns to  
A962 achieve system-level TMS behaviors.

A963 Each data flow topic has a single unique data type with defined trigger and reliability  
A964 metrics, which are discussed in [A.8.5](#). The same data type can be used by multiple data  
A965 flow topics, providing a common connection between similar data flows.

A966 Topic data values change over time. Each new value is called a *sample*. When a publisher  
A967 updates a sample, one or more messages will be sent to transmit the new value. There is  
A968 not a direct, one-to-one mapping between sample updates and messages. For example, the  
A969 communications software may bundle multiple samples into a single message, split a single  
A970 sample across multiple messages, or re-transmit samples when messages are lost. Any such  
A971 mapping is compliant as long as the overall timing requirements are met.

A972 The arrows in the data flow diagram indicate the flow of the message stream. The  
A973 arrowhead indicates the direction that messages are being sent. The message stream for  
A974 the data flows is published following specific data flow patterns that are pre-defined in the  
A975 TMS. Arrows indicate messages that can be sent from both the initiating publisher and  
A976 the subscriber in response to the publisher. Data flow patterns, including the subscriber  
A977 response patterns, are discussed in further detail in [A.8.3](#).

### A978 A.8.3 Data Flow Patterns

A979 The data flows identify and describe the system-level behaviors of TMS communication  
A980 between devices through a set of pre-defined messages. There are three basic patterns  
A981 of data flows that messages between devices follow - Status, Request/Response, and  
A982 Request/Response with Related Status - each of which is discussed below. These three  
A983 patterns describe the majority of TMS messaging. A fourth Custom data flow is used for  
A984 communication that follows a behavioral-specific pattern for a single case or condition.

### A985 A.8.3.1 Status Data Flows

A986 The Status data flow, shown in Figure A.4, shows the Status pattern, a basic announcement  
A987 which requires no reply or response.

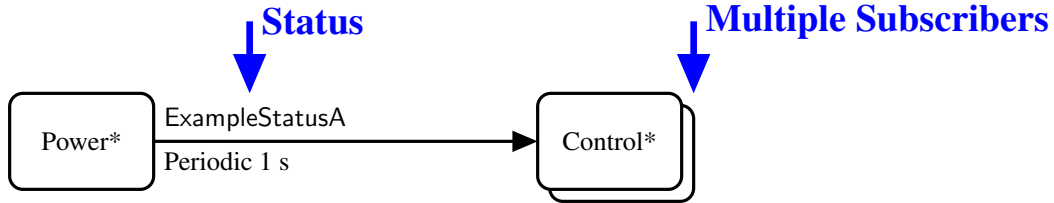


Figure A.4: Status Data Flow Pattern Example

A988 Status messages are sent from the publisher to the subscriber when an event trigger  
A989 initiates the action.

A990 Examples of event triggers include when a device starts up or if an error condition  
A991 occurs. Some event triggers are conditional, for example, a change in grid conditions or  
A992 power measurements will only count as an event trigger if a message is sent in response to  
A993 the change. Event triggers are discussed in more detail in Section A.8.5.

### A994 A.8.3.2 Request/Response Data Flows

A995 The Request/Response data flow, shown in Figure A.5, shows the Request/Response pattern,  
A996 which requires interaction between devices in the form of command and acknowledgment.

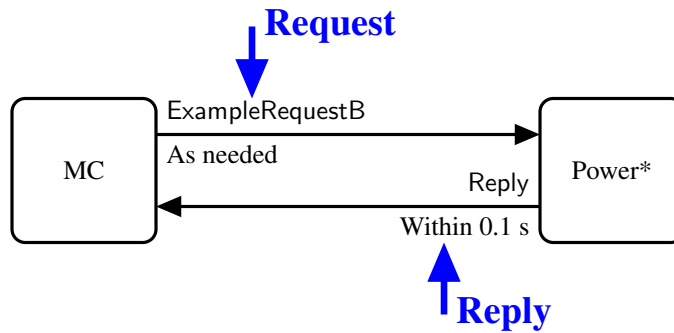


Figure A.5: Request/Response Data Flow Pattern Example

A997 A command is a request having immediate effect. Other requests may be deferred for  
A998 later execution. A reply is a data sample update sent in response to a request. Other  
A999 responses include actions taken to complete a request.

A1000 An example of this pattern is Read Configuration Settings, Section B.14. The Publisher  
A1001 requests device to send all command stored in the configuration and device responds with  
A1002 the requested values.

A1003 A.8.3.3 Request/Response with Status Data Flows

A1004 The Request/Response with Status data flow, shown in Figure A.6, shows the  
A1005 Request/Response pattern combined with the Status pattern, which requires interaction  
A1006 between devices in the form of command and acknowledgment with additional related  
A1007 information included.

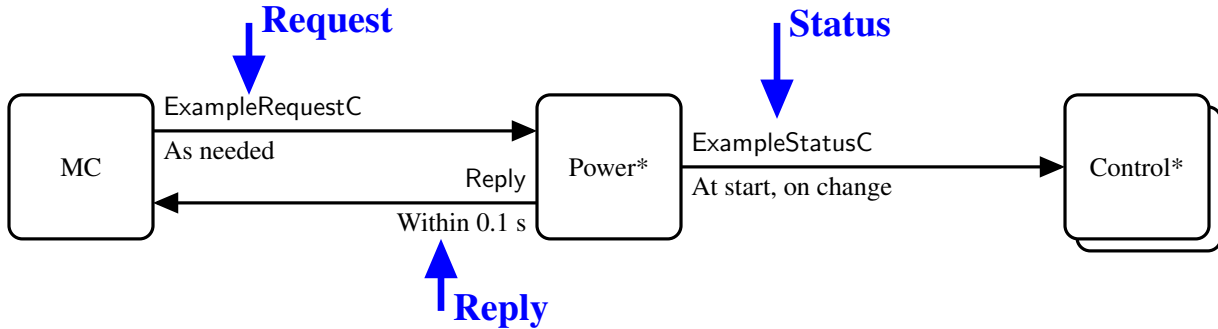


Figure A.6: Request/Response with Status Data Flow Pattern Example

A1008 When a command is sent by the publisher, the expected subscriber responds to the  
A1009 publisher and related Status information is then sent to additional subscribers.

A1010 An example of this data flow pattern is a health status request sent to a TMS device  
A1011 from the user interface to which the device responds and also sends additional related status.

A1012 A.8.3.4 Custom Data Flows

A1013 Some messages have a unique interaction pattern that doesn't match the Status or  
A1014 Request/Response patterns. These custom data flow patterns have message interactions  
A1015 that are unique and specific to that one condition or event. Typically custom patterns define  
A1016 a TMS system-level behavior. Black Start is an example of a custom data flow pattern.

A1017 A.8.4 Instances and Key Values

A1018 Some topics have a single value that applies to the entire microgrid. Other topics have  
A1019 different values for each device in the microgrid or even for each power connector in a single  
A1020 device. When a topic has different values at multiple locations, the different values are called  
A1021 sample *instances*. The location is distinguished from the rest of the data by marking it as  
A1022 a *key value*. In effect, key values are used to address, index, or "look up" information at  
A1023 different locations in the microgrid. Key values may also be used to improve performance by  
A1024 filtering messages between devices and by maintaining separate send/receive buffers within  
A1025 each devices.

A1026 When multiple data samples arrive on a topic, the key values are compared to determine  
A1027 whether each sample represents a new instance or an update to an existing instance. A  
A1028 conservative bound on the number of possible instances can be determined statically by

A1029 looking at the maximum number of devices, configurations, and ports. A tighter bound can  
A1030 be determined dynamically by looking at the device announcements.

A1031 Six data types are used as key values. These are outlined in Table A.2 and explained  
A1032 further in the following paragraphs.

Table A.2: Overview of all data types used as key values.

Data Type	Contents				Definition
	Origin	Target	Config	Port	
Fingerprint	Yes	No	No	No	Section B.2.2.2
GridRequest	Yes	No	No	No	Section B.11.2.6
DeviceRequest	Yes	Yes	No	No	Section B.6.2.6
DeviceConfigRequest	Yes	Yes	Yes	No	Section B.14.2.3
PowerPortConfigRequest	Yes	Yes	Yes	Yes	Section B.14.2.12
Reply	Yes	Yes	Yes	Yes	Section B.2.2.1

A1033 1. **Fingerprint.** This key represents the device publishing its status to the rest of the  
A1034 microgrid.

A1035 2. **GridRequest.** This key represents the device issuing a request to all devices on the  
A1036 microgrid.

A1037 3. **DeviceRequest.** This key represents the device issuing a request and the device that  
A1038 should act on it.

A1039 4. **DeviceConfigRequest.** This key represents the device issuing a request, the device  
A1040 that should act on it, and the configuration that should be updated.

A1041 5. **PowerPortConfigRequest.** This key represents the device issuing a request, the device  
A1042 that should act on it, the configuration and power port that should be updated.

A1043 6. **Reply.** This key contains values that are copied from the related request. Values not  
A1044 present in the request are filled with defaults.

#### A1045 A.8.5 Timing

A1046 Effective microgrid communications requires a careful balance of resource utilization.  
A1047 Information must be sent promptly and without delay, while keeping message size and  
A1048 frequency within resource constraints.

A1049 Four parameters define the timing requirements for each topic.

A1050 1. **Data Trigger.** Pre-defined condition that causes sample updates. Fixed rate topics  
A1051 are triggered by a periodic timer set to a nominal rate. Variable rate topics are triggered  
A1052 by a given type of event.

A1053 2. **Rate Class.** Bounds on the time interval between successive sample updates. A *rate*  
A1054 *limit* prevents excessive resource utilization, and a *deadline* ensures regular updates.

A1055 3. **Burst Size.** Number of successive sample updates that can be sent faster than the  
A1056 rate limit. Allows normally slow rates to occasionally operate at a higher rate.

A1057 4. **Quality of Service (QoS) Profile.** Policy for how data samples are prioritized,  
A1058 buffered, sent, and re-sent as messages. Accounts for constraints in network  
A1059 connectivity. The QoS profile may include some trigger, rate class, or burst size  
A1060 information for use by the communications middleware.

A1061 These timing requirements shall be used to define behavioral requirements and to  
A1062 derive minimum resource allocations for hardware and software in TMS devices. The  
A1063 application software, OMG DDS middleware, operating system, processor, memory, and  
A1064 network hardware must all work together to meet these timing requirements.

A1065 Timing requirements apply separately to each topic, publisher, and sample instance.  
A1066 There are no constraints on timing between topics, except for paired request / reply topics.  
A1067 There are no constraints on timing between publishers, though the subscriber may support  
A1068 a limited number of publishers and discard messages when that number is exceeded. There  
A1069 are no constraints on timing between sample instances, unless otherwise specified.

A1070 Some timing requirements are intentionally not specified in this interface standard. For  
A1071 example, there are almost no requirements on the timing between different devices or topics,  
A1072 so they can be sequenced together in creative ways. Also, all requirements are defined in  
A1073 terms of elapsed time, so there is no dependency on clock synchronization.

A1074 Procurement documents will specify the maximum delay from the time when a device  
A1075 is powered on until the time when it starts sending messages. This maximum delay may  
A1076 depend on operational requirements and equipment type. Simpler fixed-purpose equipment  
A1077 may have an objective delay of 3 s or less, as larger numbers need to be quickly deployed.  
A1078 Other general-purpose equipment may have an objective delay of a few minutes or less, as  
A1079 smaller numbers are used less often or with less urgency.

A1080 A.8.5.1 Data Trigger

A1081 Every topic has an assigned *data trigger*, a pre-defined condition that causes the publisher  
A1082 to send new data samples and thus new messages. The data trigger is specified in the timing  
A1083 table for each data flow, and it is shown underneath each topic arrow in the data flow  
A1084 diagrams.

A1085 Publishers should send sample updates when indicated by the data trigger. Whether the  
A1086 publisher sends sample updates as indicated or for any other reason, the aggregate traffic  
A1087 shall still remain within the specified rate class and burst size.

A1088 Eight data triggers are defined. Four are variable rate and four are fixed rate.

A1089 1. **At Start.** Values are set during start-up. Messages containint the start-up value are  
A1090 sent to other devices that connect at a later time. Values are “constant” during normal  
A1091 operation and may only change when the device restarts.

A1092 2. **At Start, On Change.** Values are set during start-up and may update during  
A1093 operation of the device, such as in response to received commands or internal  
A1094 measurements. Each device is responsible for monitoring its internal state and updating  
A1095 these values promptly, subject to the rate class.

A1096 3. **As Needed.** Values are set and update in response to manual actions or automatic  
A1097 logic. There is no requirement for a start-up value.

A1098 4. **Within 0.1 s.** Values update immediately after receiving another message. A 0.1-  
A1099 second response time is allowed. Generally used for reply topics.

A1100 5. **Periodic 0.1 s.** Values update every 0.1 seconds, even if the only change is time and  
A1101 the message merely re-asserts the previous values.

A1102 6. **Periodic 1 s.** Values update every 1 seconds, even if the only change is time and the  
A1103 message merely re-asserts the previous values.

A1104 7. **Periodic 10 s.** Values update every 10 seconds, even if the only change is time and  
A1105 the message merely re-asserts the previous values.

A1106 8. **Periodic 900 s.** Values update every 900 seconds, equivalently every 15 minutes, even  
A1107 if the only change is time and the message merely re-asserts the previous values.



#### A1108 A.8.5.2 Rate Class

A1109 Every topic has an assigned *rate class*, pre-defined limits on the time interval between  
A1110 successive sample updates. The rate class is specified in the timing table for each data  
A1111 flow, and it is not shown in the data flow diagrams.

A1112 All topics have a minimum separation time interval called a *rate limit* that prevents  
A1113 excessive resource utilization. Rate limits require some amount of time to pass between  
A1114 successive sample updates thus reserving time for each update to be handled before the next  
A1115 arrives. Topics with a periodic data trigger also have a maximum separation time interval  
A1116 called a *deadline* that ensures regular updates. The rate limit and deadline are enforced at  
A1117 the publisher and may be enforced at the subscriber or elsewhere in the network.

A1118 Publishers shall ensure that the time between successive sample updates is greater than  
A1119 or equal to the rate limit for every topic. Publishers should use a timer or rate limit algorithm  
A1120 to ensure adequate time between samples, possibly delaying or merging sample updates as  
A1121 needed. Subscribers or network routers may discard samples that arrive too quickly.

A1122 Publishers shall ensure that the time between successive sample updates is less than or  
A1123 equal to the deadline for every topic having a periodic data trigger. Deadlines do not apply  
A1124 to any topic having a non-periodic data trigger. Publishers should re-publish old values as  
A1125 necessary before a timeout occurs. Subscribers or network routers may trigger an alert or  
A1126 fault recovery action when samples arrive too slowly.

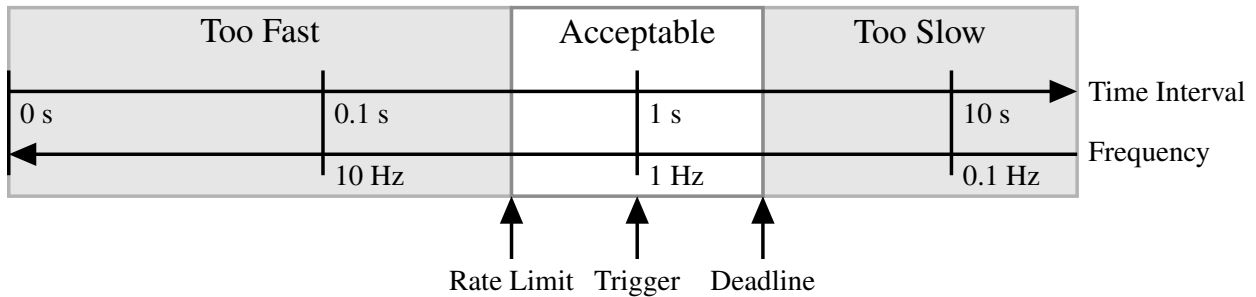


Figure A.7: Example timing for a periodic 1 s topic.

A1127 The publisher should allow at least a 10 ms margin in the rate limit and deadline for  
A1128 network jitter, changes in the end-to-end timing of successive messages. The subscriber is  
A1129 not required to allow any margins. Example timings are shown in Figure A.8 and Figure A.9.

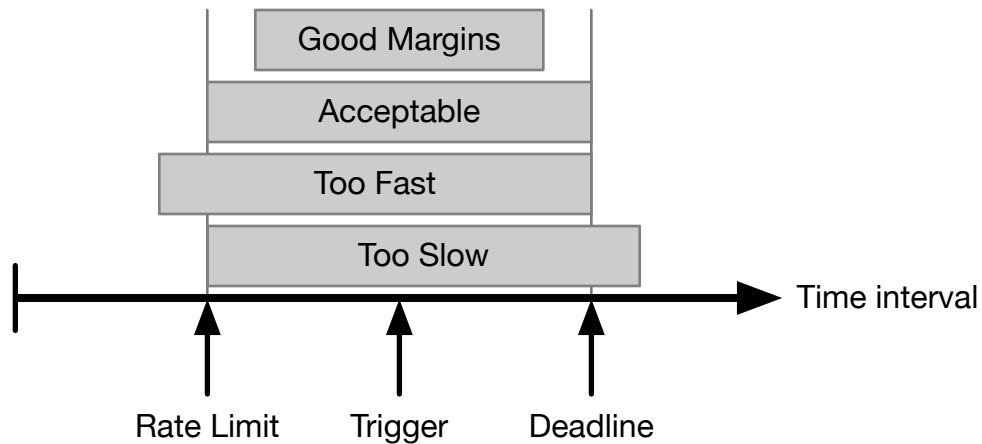


Figure A.8: Example publisher send windows.

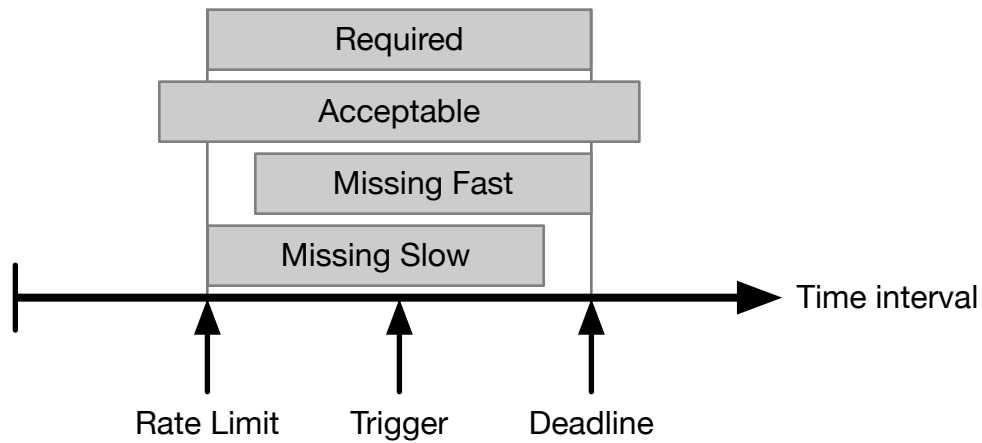


Figure A.9: Example subscriber receive windows.

A1130 Six rate classes are defined.

- A1131 1. **All.** All update rates are acceptable, with no rate limit or deadline.
- A1132 2. **0.1 s.** Nominal update rate of 0.1 s or slower.
- A1133 3. **1 s.** Nominal update rate of 1 s or slower.
- A1134 4. **10 s.** Nominal update rate of 10 s or slower.
- A1135 5. **900 s.** Nominal update rate of 900 s, equivalently 15 m, or slower.
- A1136 6. **Once.** No nominal update rate. Only a single value may be sent.

A1137 A small set of pre-defined rates was selected to provide broad coverage and simple  
A1138 implementation. The 0.1 s rate class is fast enough to coordinate load sharing, even with

A1139 30% sample loss. Faster rates may require specialized hardware or software. The 900 s rate  
A1140 class is slow enough that network traffic is negligible and long-term logging is feasible. Slower  
A1141 rates can use the 900 s rate class.

A1142 Each rate class is defined in terms of four parameters, two guideline values and two  
A1143 constraints.

A1144 1. **Nominal Rate.** Nominal separation time interval between successive sample updates.

A1145 2. **Jitter.** Allowed variation in timing between sample updates.

A1146 3. **Sustained Rate.** Minimum separation time interval between successive sample  
A1147 updates. Applies to all data triggers. Limits the maximum rate.

A1148 4. **Deadline.** Maximum separation time interval between successive sample updates.  
A1149 Only applies to periodic data triggers. Limits the minimum rate.

Table A.3: Specifications for each rate class.

Rate Class	Guidelines		Constraints	
	Nominal Rate	Jitter	Sustained Rate	Deadline
All	$\geq 0$ s	N/A	0 s	Infinite
0.1 s	$\geq 0.1$ s	$\leq 0.01$ s	0.09 s	2 s
1 s	$\geq 1$ s	$\leq 0.1$ s	0.9 s	3 s
10 s	$\geq 10$ s	$\leq 0.5$ s	9.5 s	20 s
900 s	$\geq 900$ s	$\leq 50$ s	850 s	2000 s
Once	N/A	N/A	Infinite	Infinite

A1150 *Implementation Note: Many TMS devices will not require any special effort to stay*  
A1151 *within the rate limits. For those that do, ITU-T Recommendation I.371 provides efficient*  
A1152 *virtual scheduling algorithms and leaky bucket algorithms that may be used to implement rate*  
A1153 *limiting. The TMS timing is compatible with the Generic Cell Rate Algorithm (GCRA),*  
A1154 *Sustained Cell Rate (SCR), and Maximum Burst Size (MBS).*

A1155 The rate classes are selected to be consistent with the data triggers. Fixed-rate data  
A1156 triggers are matched to rate classes with the same nominal rate. Variable-rate data triggers  
A1157 are matched to a rate class based on the sustained rate. These relationships are shown in  
A1158 Figure [A.4](#).

Table A.4: Relationships between data triggers and rate classes.

Data Trigger	Rate Class	Deadline
At Start	Once	No
At Start, On Change	<i>varies</i>	No
As Needed	<i>varies</i>	No
Within 0.1 s	<i>varies</i>	No
Periodic 0.1 s	0.1s	Yes
Periodic 1 s	1 s	Yes
Periodic 10 s	10 s	Yes
Periodic 900 s	900 s	Yes

### A1159 A.8.5.3 Burst Size

A1160 Every topic has an assigned *burst size*, allowing sample updates to exceed the rate limit for  
A1161 short periods of time, while still averaging to stay within the rate limit over longer periods  
A1162 of time. The burst size is specified in the timing table for each data flow, and it is not shown  
A1163 in the data flow diagrams.

A1164 Some operator inputs, transitions, and alarms have short periods of high activity between  
A1165 long periods of little or no activity. Periodic sampling at a high rate would be inefficient,  
A1166 and sampling at a low rate would introduce unacceptable delay. It is generally preferred for  
A1167 sample updates to stay within the rate class, but rapid activities may justify faster updates as  
A1168 long as resources are not overwhelmed. For example, an operator may change a configuration  
A1169 setting multiple times before finding the right value. As another example, a power device  
A1170 may sequence through multiple states to complete a request or respond to a power event.

A1171 Publishers may send messages in bursts. Subscribers may respond to bursts by not fully  
A1172 responding to each sample update, effectively treating newer samples as higher priority than  
A1173 older samples. For example, successive requests to stop and then start may result in a device  
A1174 staying on rather than stopping briefly. Subscribers should reply to every request in a burst.  
A1175 Further requirements may be added by procurement documents.

A1176 The burst size and rate limit interact in the following manner. This description uses a  
A1177 timer and burst count. Other implementations are permitted.

- A1178 1. Initialize the burst count to the specified burst size.
- A1179 2. When a sample update would exceed the rate limit, check that the burst count is  
A1180 greater than zero, then count down by one.
- A1181 3. When a sample update is skipped, that is two nominal rate time intervals elapse without  
A1182 a sample update in the middle, then count up by one but not exceeding the burst size.

A1183 Two burst sizes are defined.

- A1184 1. **One (1).** A single message may exceed the rate limit.
- A1185 2. **Ten (10).** Ten messages may exceed the rate limit.

A1186 A.8.5.4 Quality of Service (QoS)

A1187 *Editorial note: Work in progress.*

A1188 Every topic has an assigned *QoS profile*, each profile specifies settings for the various QoS  
A1189 options, ...

A1190 There are many aspects to QoS and thus many options. A *QoS profile* selects one option  
A1191 for each aspect. These profiles are designed to meet the timing requirements of Groups of  
A1192 related selections are called *profiles*. Each TMS topic has an assigned profile.

History			Liveliness	
QoS Profile	Kind	Depth	Kind	Duration
Publish Last	Keep Last	1	Automatic	Infinite

A1194 All QoS profiles have the history set to keep the last 1 sample. All QoS profiles have the  
A1195 liveliness set to automatic, infinite.

A1196 A.8.6 Quality of Service Options

A1197 *Quality of Service* is used to specify how messages are sent in each data flow. TMS defines  
A1198 the following QoS...

A1199 A.8.6.1 Ordering

A1200 Note that Related Status identifies the status information that is specific to the  
A1201 request/response pair of the data flow.

A1202 There are times when a device role is handling multiple messages

A1203 In the actual message structure, part of the response value is also part of the request  
A1204 value. The request has an ID and the response contains a copy of the request ID. The  
A1205 status contains a copy of other parts of the request value. When the request is accepted,  
A1206 the response is acknowledged or denied immediately but it may take time to execute so the  
A1207 related status is sent after the response has been executed.

A1208 This standard does not require specific timing or order of delivery across multiple topics.  
A1209 In particular, sample updates may be sent to multiple topics in one order and received in  
A1210 another order. Application software shall not rely on ordered delivery of samples between  
A1211 topics. When ordered execution of commands is required across topics, application software  
A1212 shall wait for responses to earlier commands before sending later commands.

A1213 A.8.6.2 Coherency

A1214 By default, all message delivery is handled in parallel, without a defined order of delivery  
A1215 across devices. Values of a single object shall be sent “in order”, but values across multiple  
A1216 objects may be delivered “out of order”. If a message sequence is marked as *coherent*, then  
A1217 the subscriber must not process any objects in the set until all have been received. If a  
A1218 late-joining subscriber, communications error, or early-leaving subscriber prevents delivery  
A1219 of the full sequence, then none of the objects shall be processed.

A1220 A.8.6.3 Durability

A1221 The durability policy controls the fate of objects sent before a subscriber connects to the  
A1222 publisher. *Volatile* objects are sent to the current subscriber list, and no attempt is made to  
A1223 send them to late-joining subscribers. *Transient* objects are sent to all current and future  
A1224 subscribers that match with the original publisher. All objects are volatile unless otherwise  
A1225 specified.

A1226 By default, all object values are lost when a device turns off. The data model may specify  
A1227 that some values are *persistent*, meaning that the subscriber shall store the most recent value  
A1228 such that it is preserved across restarts.

A1229 A.8.6.4 History

A1230 Unless a history depth specified, only the most recent value of an object is sent. When a  
A1231 new value becomes available, any previous values that were not sent are silently discarded.  
A1232 This affects both late subscribers and messages that are delayed due to network congestion  
A1233 or loss.

A1234 This definition is similar to the HISTORY QoS policy in DDS.

A1235 History may be implemented by creating an explicit array or ring buffer of the specified  
A1236 depth.

A1237 A.8.6.5 Liveliness

A1238 In the tactical microgrid, liveliness is the property of being connected and actively  
A1239 communicating with other devices. The mechanism for detecting liveliness is specific to  
A1240 the communications profile. The communications layer may assert that a message is live  
A1241 without having to re-send the values.

A1242 A.8.6.6 Real-Time

A1243 This standard does not define any message flows that would require the use of real-time  
A1244 networking protocols. Instead, tasks with hard real-time constraints, such as synchronizing  
A1245 phases or opening a protective breaker, are defined so that they can be implemented within  
A1246 a single device.

A1247 A.8.6.7 Reliability

A1248 A *best effort* delivery will attempt to transmit a data object, but not confirm that it  
A1249 was received. A *reliable* delivery mechanism will maintain a copy of the data object and  
A1250 periodically resend it until the receiver acknowledges delivery. All data flows are best effort  
A1251 unless otherwise specified.

A1252 A.8.7 Optional Topics

A1253 Some customers may want to exclude features to reduce cost, simplify operation, reduce  
A1254 cybersecurity risk, or meet some other objective. The data model provides some flexibility

A1255 to meet these needs by not requiring all topics to be implemented. The data model does  
A1256 impose some constraints on how topics are implemented to avoid a combinatorial explosion  
A1257 of complexity. Each device's **ProductInfo** indicates which topics it implements.

A1258 All topics have been assigned to one of the following three categories.

A1259 **Core** Required of all TMS devices.

A1260 **Conditional** Required of TMS devices that meet specific conditions as documented in this  
A1261 standard. Most conditions are in **ProductInfo**.

A1262 **Optional** May be required by acquisition documents / customers. Listed in **ProductInfo**.

A1263 Within each data flow, the status topic is a precondition for the other topics, and the  
A1264 request/response topics must be implemented as a pair.

A1265 Most of these requirements are documented from the perspective of the publisher.  
A1266 Conformance testing will verify that publishers and subscribers participate in topics  
A1267 consistent with their **ProductInfo**. Publishers must update samples according to the specified  
A1268 timing. Subscribers must behave according to the samples received.

A1269 *Editorial note: This section will describe how optional topics are indicated in the data*  
A1270 *flows.*

A1271 *Editorial note: This section will describe the general expected behavior when one device*  
A1272 *does not implement a topic. Each data flow will describe any flow-specific expectations.*

A1273 *Editorial note: The acquisition guidance section will provide a summary of what*  
A1274 *subsetting options exist and recommendations for selecting thresholds and objectives.*

## A1275 A.9 Data Types

A1276 The TMS data model standard is designed to be platform-independent, allowing flexibility  
A1277 in system design and implementation.

A1278 All data types in the data model represent platform-independent logical types. Platform-  
A1279 specific details such as data APIs and network serialization are defined in the communications  
A1280 profile that is selected to implement the data model. This section describes the generic  
A1281 primitive and constructed data types that are used for the TMS data model. The domain-  
A1282 specific data types are discussed in Chapter B.

A1283 The data model messages are pre-defined objects, each having a name, type, and value.  
A1284 The name and type are constant, as defined by TMS. The type identifies possible values for  
A1285 the messages where each message may have a different value.

### A1286 A.9.1 Type Hierarchy

A1287 This section defines the types available for use and the relationship between them. Three  
A1288 categories of types are defined in the data model.

- A1289 1. **Primitive Types.** Building blocks for individual pieces of information.
- A1290 2. **Constructed Types.** Building blocks for relationships between information.
- A1291 3. **Domain-Specific Types.** Primitive or constructed types, used as building blocks or  
A1292 complete data samples, and defined in Chapter B.

### A1293 A.9.2 Type Names

A1294 Every data type is defined with a unique *name*.

A1295 When creating data type names for TMS, the names must:

- A1296 1. Only contain alphabet ('a'-'z' and 'A'-'Z'), digit ('0'-'9'), and underscore ('\_')  
A1297 characters
- A1298 2. Begin with an alphabet character
- A1299 3. Always be used with the same capitalization (uppercase and lowercase alphabet) as  
A1300 their definition
- A1301 4. Be unique beyond their capitalization

A1302 For example, the names `abc`, `Abc123`, and `a_b_c` are valid. The names `123Abc` and `_abc`  
A1303 are invalid. Both names `abc` and `ABC` are valid, but it is invalid to use both capitalizations  
A1304 within the same type or namespace.

A1305 This data model does not define any reserved words. To avoid software development  
A1306 issues, this data model avoids any names that are reserved words in the C++, Java, OMG  
A1307 IDL, or Structured Text programming languages.



### A1308 A.9.3 Type Namespaces

A1309 Every type name exists within a *namespace* that defines the scope of the type definition.  
A1310 Namespace names use the same character rules as type names. Namespaces may contain  
A1311 other, nested namespaces. The top-level, “unnamed” namespace is the *global namespace*. In  
A1312 a single namespace, every definition has a unique name. In multiple namespaces, the same  
A1313 name may be used for different definitions.

A1314 When a type name is referenced, the namespace may be explicit (*qualified name*) or  
A1315 implicit (*unqualified name*). A qualified name is written as a concatenation of namespaces,  
A1316 from outer to inner, each separated by a dot (‘.’), and ending with the type name. An  
A1317 unqualified name is simply the type name.

A1318 For example, the qualified name `tms.Heartbeat` refers to the `Heartbeat` type within  
A1319 the `tms` namespace. Similarly, the qualified name `tms.dc.PowerMeasurement` refers to the  
A1320 `PowerMeasurement` type, within the `dc` namespace, within the `tms` namespace.

A1321 Unqualified names are *resolved* by searching for the nearest matching namespace. This  
A1322 search first checks for a match in the namespace where the unqualified name is used, and  
A1323 then checks containing namespaces. If no match is found, then there is an error in the name  
A1324 resolution. Qualified name are resolved by simply traversing the given namespace path,  
A1325 without searching. All qualified names in TMS are fully qualified, and it is an error to only  
A1326 include part of the namespace path.

A1327 For example, in both the `tms` and `tms.dc` namespaces, the unqualified name `Heartbeat`  
A1328 resolves to `tms.Heartbeat`. The unqualified name `PowerMeasurement` resolves to either  
A1329 `tms.PowerMeasurement` or `tms.dc.PowerMeasurement`, depending on where it is used.  
A1330 The partially qualified name `dc.PowerMeasurement` is invalid since it is missing the `tms`  
A1331 namespace.

A1332 The notation for qualified names may also be used to reference attributes within data  
A1333 types. In this usage, the attribute name is dropped before performing name resolution for  
A1334 the type.

A1335 All data types defined by TMS exist in the `tms` namespace or a nested namespace.  
A1336 All names in the `tms` namespace are reserved for definition by TMS. Table A.5 lists all  
A1337 namespaces defined by TMS.

Table A.5: Namespaces defined by TMS.

Namespace	Purpose
<code>tms</code>	TMS core, including AC power
<code>tms.dc</code>	DC power
<code>tms.esu</code>	Storage devices

A1338 TMS uses a naming convention to help distinguish between the different kinds of names.  
A1339 General-purpose data types use short lowercase words. Domain-specific data types use longer  
A1340 “CamelCase” words. Named values use “UPPERCASE\_NAMES”. Namespaces use lowercase  
A1341 abbreviations.

#### A.9.4 Units of Measure

Machine-to-machine communications shall always transmit values using the units of measure specified by the data type. Units of measure are specified by adding a *units* annotation to the data type. All measurements are encoded using metric units. Measurements use the appropriate base or derived unit of measure, without metric prefixes or other scaling.

User interfaces may use more convenient terminology for presentation and operator input, converting to/from the specified units as required. For example, one kilowatt is communicated as 1000 watts, one hour is communicated as 3600 seconds, and one kilowatt-hour is communicated as 3,600,000 joules.

This policy is intended to minimize the risk of conversion errors. It is also in accordance with the guidelines presented in IEEE Std 945.

The following units of measure are used in TMS:

Table A.6: Units of Measure

Abbreviation	Name	Description
A	ampere	Electrical Current
C	Celsius	Temperature
°	degree	Plane and phase angle
Hz	hertz	Frequency
J	joule	Energy, work, amount of heat
L	litre	Volume
Pa	pascal	Pressure, stress
rad	radian	Plane Angle
s	second	Time
S	siemens	Electrical conductance
V	volt	Electrical Voltage
VA	volt ampere	Power (apparent)
var	volt ampere reactive	Power (reactive)
W	watt	Power (real)

In addition to raw units of measure, normalized or “per unit” measurements are used in situations where the relative value is more important than the absolute value. For example, a gauge may show when a resource is empty or full, or load sharing may give each source an equal fraction of the load relative to its rated capacity. Per-unit measurements generally range from 0 to 1, or from -1 to +1 where negative values make sense. While per-unit measurements are effectively unitless, the original unit of measure is often indicated for clarity. For example, “p. u. watts” indicates a power measurement that has been normalized.

### A.9.5 Primitive Data Types

Primitive data types, also known as atomic, scalar, or simple data types, hold values that have a single assignable part. Primitive data types represent individual pieces of information and are generally used as building blocks in other types. An overview of how primitives are generally used is given in Table A.7.

Table A.7: Overview of Primitive Data Types and their General Usage

Data Type	General Usage
Boolean	True/false values
Character	Human-readable text
Octet	Container for uninterpreted values
Integer Number	Countable, discrete quantities and indices
Floating-Point Number	Measurable, continuous quantities
Enumeration	Named categories and options

#### A.9.5.1 Boolean

A boolean, or bool for short, is a value that is either true or false.

The data model represents all Booleans as separate, discrete values. Implementations may pack multiple booleans as individual bits within a larger data type. For example, eight separate booleans may be packed together as individual ones and zeros within a single 8-bit integer.

For an example boolean, see the `hasRealtimeClock` attribute in `tms.ControlHardwareInfo`.

#### A.9.5.2 Character

A character, or char for short, is an 8-bit value that represents a UTF-8 code point in the Universal Coded Character Set (UCS, Unicode), as defined in ISO/IEC 10646.

To maximize portability, publishers should only use characters in the Portable Character Set, as defined in POSIX.1 / IEEE Std 1003.1. Control characters other than U000A (`'\n'`, newline) should not be used. For convenience, the portable code points are listed in Table A.8. Devices may render characters outside this range with a “missing symbol” character such as a white square (U25A1).

Table A.8: UCS Portable Character Set

UCS	Char	UCS	Char	UCS	Char	UCS	Char
U0000	Null	U0037	7	U004F	O	U0067	g
U000A	Newline	U0038	8	U0050	P	U0068	h
U0020	Space	U0039	9	U0051	Q	U0069	i
U0021	!	U003A	:	U0052	R	U006A	j
U0022	"	U003B	;	U0053	S	U006B	k
U0023	#	U003C	<	U0054	T	U006C	l
U0024	\$	U003D	=	U0055	U	U006D	m
U0026	&	U003E	>	U0056	V	U006E	n
U0027	'	U003F	?	U0057	W	U006F	o
U0028	(	U0040	@	U0058	X	U0070	p
U0029	)	U0041	A	U0059	Y	U0071	q
U002A	*	U0042	B	U005A	Z	U0072	r
U002B	+	U0043	C	U005B	[	U0073	s
U002C	,	U0044	D	U005C	\	U0074	t
U002D	-	U0045	E	U005D	]	U0075	u
U002E	.	U0046	F	U005E	^	U0076	v
U002F	/	U0047	G	U005F	_	U0077	w
U0030	0	U0048	H	U0060	`	U0078	x
U0031	1	U0049	I	U0061	a	U0079	y
U0032	2	U004A	J	U0062	b	U007A	z
U0033	3	U004B	K	U0063	c	U007B	{
U0034	4	U004C	L	U0064	d	U007C	
U0035	5	U004D	M	U0065	e	U007D	}
U0036	6	U004E	N	U0066	f	U007E	~

A1381 For an example character, see the [tms.NatoStockNumber](#).

#### A1382 A.9.5.3 Octet

A1383 An **octet** is an opaque container for an 8-bit value. Communications between devices are  
A1384 guaranteed to preserve the order and value of all eight bits. The value sent always equals  
A1385 the value received. No other interpretation is defined for the contents. In particular, octets  
A1386 are not interpreted as numbers, so there is no notion of “signed” or “unsigned” octets.

A1387 For an example octet, see the [tms.Fingerprint](#).

#### A1388 A.9.5.4 Integer Numbers

A1389 An integer number, or **int** for short, is a whole number with zero fractional value. Signed  
A1390 integers can have positive or negative values. Unsigned integers can only store zero and  
A1391 positive values.

A1392 Integers are specified using the fixed-width type names defined in POSIX.1 / IEEE Std  
A1393 1003.1. The “\_t” suffix is dropped from these type names for brevity (it was merely added  
A1394 to avoid naming conflicts with legacy source code).

The following integer types are required:

Table A.9: Required Integer Types

Size	Signed	Unsigned
8	int8	uint8
16	int16	uint16
32	int32	uint32
64	int64	uint64

For example integers, see the `numNetworkPorts` attribute in `tms.TopicInfo` and both attributes in `tms.ClockMonotonic`.

#### A.9.5.5 Floating-Point Numbers

A floating-point number, or float for short, is a number that may have a nonzero fractional value. Floating-point numbers are signed values that may be positive or negative. Floating-point numbers also have special “not a number” (NaN) values.

Floating-point numbers are defined in IEEE Std 754 / IEC 60559. The only required type is float32, corresponding to “single precision” or binary32 in IEEE Std 754.

Publishers may use a NaN value to indicate that a measurement is not available, whether due to design or fault.

For example floats, see the attributes in `tms.ControlHardwareInfo` and `tms.ControlHardwareState`.

#### A.9.5.6 Enumeration

An enumeration, or enum for short, is a set of named values and each value is unique. Every instance of the enumerated type contains exactly one of these named values.

The maximum number of values in an enumeration is equal to the number of values that can be stored in an int32. Any mapping between named values and integers should not be relied upon in application software, as it is an implementation detail that may not be portable or interoperable. Implementations should preserve the relative ordering of values as specified in the data model. For example, different implementations may represent enumerations as integers counting from 0, integers counting from 1, or characters.

The following naming convention is used to give each enumerated value a unique name. Enumeration type names start with a capital letter and each new word is capitalized. Enumeration value names start with a capitalized abbreviation of the type name. Each word is capitalized and prefixed with an underscore ('\_'). For example, ExampleEnum may contain EE\_VALUE1 and EE\_VALUE2.

For an example enumeration, see `tms.DeviceRole`.

#### A.9.6 Constructed Data Types

Constructed data types, also known as aggregate, compound, or complex data types, hold values that may have multiple, separately assignable parts. Constructed data types represent

A1426 relationships between information and are generally used to provide richer semantics. Some  
A1427 constructed types are used as building blocks for larger types. All TMS messages are  
A1428 constructed types.

#### A1429 A.9.6.1 Attributes

A1430 Constructed types may have multiple pre-defined attributes that are unique to the  
A1431 constructed type and provide further definition of the possible values for that type.  
A1432 Attributes follow consistent patterns, such as request/response messaging, but each attribute  
A1433 is specific to the constructed type in which it is used.

A1434 The value of a constructed data type is the collective value of all of the type's attributes.  
A1435 For example, a Date type could have attributes for the year (integer), month (enumeration  
A1436 or 1-12), and day (1-31). The value of this Date type is the year, month, and day as a  
A1437 combined unit.

#### A1438 A.9.6.2 Constants

A1439 Each domain-specific type may include definitions for one or more constants. Constants are  
A1440 named values that have special semantics when associated with a domain-specific data type.  
A1441 Constants are defined Each constant is given a unique name. Multiple constants may have  
A1442 the same value, each with its own semantics.

A1443 For an example use of constants, see [tms.ReplyStatus](#).

A1444 The data model also defines special constants for every constructed types that has a  
A1445 specified length. These are listed in Table A.10.

Table A.10: Naming convention for pre-defined length constants.

Constant Name	Constant Description
<i>TypeName_LEN</i>	Length of array defined as typedef.
<i>AttributeName_LEN</i>	Length of array defined in structure.
<i>TypeName_MINLEN</i>	Minimum length of sequence.
<i>TypeName_MAXLEN</i>	Maximum length of sequence.

#### A1446 A.9.6.3 Structure

A1447 A structure, or struct for short, is a logical collection of attributes that are transmitted as  
A1448 a unit. Each attribute has a name and a data type. Attribute names are lexically-scoped  
A1449 within the structure type. Attribute names must be unique within the structure but may be  
A1450 re-used with in other structures.

A1451 A naming convention has been established for structures. Structure type names start  
A1452 with a capital letter and each new word is capitalized. Structure attribute names start with  
A1453 a lowercase letter and each new word is capitalized. For example, **ExampleStructure** may  
A1454 contain **exampleAttribute1** and **exampleAttribute2**.

A1455 For an example structure, see [tms.ReplyStatus](#).

A1456 A.9.6.4 Array

A1457 An array is a fixed length, ordered list of elements, all having the same type.

A1458 Each array has two attributes, the data type and the length. Arrays may be nested to  
A1459 create multi-dimensional values.

A1460 The notation for a single-dimensional array is `type[length]`. The notation for a multi-  
A1461 dimensional array is `type[length1][length2]...[lengthN]`.

A1462 Individual elements in an array may be specified by an array index. Ordering must be  
A1463 preserved but indices are not specified.

A1464 A.9.6.5 Sequence

A1465 A sequence is a variable length, ordered list of elements, all having the same type. The name  
A1466 “sequence” is consistent with data schemas such as OMG IDL and W3C XSD. Programming  
A1467 languages may use other names for this data type such as array list, dynamic array, ordered  
A1468 collection, and vector.

A1469 The notation for a sequence is `sequence<T,minOccurs,maxOccurs>`. The three  
A1470 parameters inside the `<` and `>` type template are defined as follows.

- A1471 • The `T` is the data type of all elements in the sequence.
- A1472 • The `minOccurs` is the minimum number of elements required by the protocol.
- A1473 • The `maxOccurs` is the maximum number of elements allowed by the protocol.

A1474 Publishers shall set the sequence length between `minOccurs` and `maxOccurs`. Subscribers  
A1475 should verify that the length is between these bounds.

A1476 Both `minOccurs` and `maxOccurs` should be checked by sending and receiving devices.  
A1477 Devices may also use `maxOccurs` to reserve memory for messaging. A sequence value that  
A1478 has zero length is called “empty”.

A1479 The `ExampleSequenceEncoding` shows a common memory representation for sequences.

<u>example.ExampleSequenceEncoding</u>	
PURPOSE: Illustrate a common binary encoding for sequences.	
PATTERN: Structure	
ATTRIBUTES:	
Name	Type and Description
<b>length</b>	uint32 Counter indicating the number of buffer entries actually used.
<b>data</b>	T[maxOccurs] Buffer allocated to hold sequence values. Entries are used in order from the first index.
CONSTANTS:	
Name	Type [Units], Value, and Description
<b>minOccurs</b>	uint32, 0 Minimum array length
<b>maxOccurs</b>	uint32, 5 Maximum array length
DESCRIPTION:	
The attributes in the example above represent how the elements of the sequence <b>sequence</b> <T,minOccurs=0,maxOccurs=5> may be stored in memory. Other representations may be used. For message transmission, the <b>length</b> is sent first, followed by the contents of the <b>data</b> buffers that are in use. Note: In this example, the minOccurs and maxOccurs values are fixed by the definition of the sequence so they do not need to be sent.	

Figure A.10: Example Encoding of a Sequence as a Structure with Constants

#### A1480 A.9.6.6 String

A1481 The string type is specialized sequence of characters.

A1482 The notation for a string is **string**<minLength,maxLength>. This is nearly equivalent  
A1483 to **sequence**<char,minLength,maxLength>.

A1484 There are two important differences. The **string** cannot contain a null (0-valued)  
A1485 character. The **string** will be terminated by an additional null character as appropriate  
A1486 for the native programming language.

A1487 For example string values, see the uses of [tms.String32](#) in [tms.ProductInfo](#).



#### A.9.6.7 Optional

The optional type is a container for a single value. The notation is `optional<T>`, where `T` is the type of the contained value. This value may be present or absent. When present, it's referred to as "full" and when absent, it's "empty."

The optional type may be implemented as a `sequence<T,0,1>`, a structure containing a Boolean flag, a value of type `T`, or some other representation. The functional requirement is that presence and absence of the contained value can be detected without restricting the range of `T`. The data model does not specify whether space must be reserved for absent values.

For example optional values, see the `controlHardware` and `powerHardware` attributes in [tms.DeviceInfo](#).

#### A.9.6.8 Typedef

A typedef creates an alias name for another datatype. The type represented by the alias name contains the same values as the original datatype. This alias name is used to indicate a special meaning for uses of the type. In TMS, the typedef can be used both as an individual variable in a another type or as a wholly-defined type, where it becomes a constant of that defined type.

For instance, if a sequence type called "SerNum" is defined with a set of attributes, a typedef could be made called "PublicKey" that has SerNum as one of its attributes. Or if a structure called "Date" is defined with attributes of "Month" and "Year" then a typedef is defined called "TestDate" that uses the Date structure. The specific instance of TestDate calls the Date type and binds the Date attributes to it.

This is a useful type for building common domain-specific constructed types that can then be called out in other domain-specific constructed types.

For example, it may be used to specify units for primitive types or to specify fixed attribute values (type and length) for array and sequence types.

For an example typedef, see [tms.PowerPortNumber](#).

### A.9.7 Domain-Specific Data Types

As explained in Section [A.9.1](#), primitive and constructed data types are building blocks used to define domain-specific data types. These domain-specific data types specify the actual values that are communicated between devices.

Domain-specific data types are defined in Chapter [B](#) using one of the following three patterns.

1. **Enumeration.** Define a new enumeration (Section [A.9.5.6](#)) and its named values.
2. **Structure.** Define a new structure (Section [A.9.6.3](#)) and the attributes it contains.
3. **Typedef.** Use a typedef (Section [A.9.6.8](#)) to define new semantics for an existing type.

A1524 A.9.8 Default Values

A1525 Publishers shall provide all values in every sample update. Unless otherwise specified, the  
A1526 default values specified in Table A.11 may be used.

Table A.11: Default Values for all Data Types	
Data Type	Default Value
Boolean	False
Character	U0000 (Null)
Octet	0x00 (Zero)
Integer	0 (Zero)
Float	Not a Number (NaN)
Enumeration	UNKNOWN named value
Structure	Set each attribute to default value
Array	Set each element to default value
Sequence	Zero-length sequence
String	Zero-length string
Optional	Empty (no contained value)
Typedef	See original data type

A1527 A.9.9 Annotations

A1528 A.9.9.1 Units

A1529 A.9.9.2 Extensibility

A1530 In consideration of future TMS development, some domain-specific data types may evolve  
A1531 with further functionality, such as new fields in a structure or increased sequence size.  
A1532 To ensure that such development does not break interoperability with previously-deployed  
A1533 systems, future development will follow the three kinds of extensibility types defined in this  
A1534 section.

- A1535 • **FINAL** - indicates that no future changes may be initiated.
- A1536 • **APPENDABLE** - indicates that additional range, values, or attributes may be added.  
A1537 All changes must be contained within all elements/members of the existing type and  
A1538 appended to the end whereby existing systems can ignore or cut-off such additional  
A1539 appended information without impact to operations.
- A1540 • **MUTABLE** - indicates that the type may differ from existing systems in regards to  
A1541 attributes of the type that may be added, removed, or renamed.

A1542 In general, most TMS types are final, some are appendable, and very few are mutable.  
A1543 Table A.12 summarizes the appendable values of the extensibility type.

Table A.12: Appendable Extensibility for all Data Types

<b>Data Type</b>	<b>Appendable Values</b>
Boolean	N/A
Character	N/A
Octet	N/A
Integer Numbers	Smaller min, larger max
Floating-Point Number	Smaller min, larger max
Enumeration	Add new named value(s) to the end
Structure	Add new attributes to the end
Array	Increase length / add new elements to the end
Sequence	Increase maximum length
String	Increase maximum length
Optional	N/A
Typedef	See original data type

A1544 A.10 Resource Allocation

A1545 Various features of the data model can be used together to estimate the resources required  
A1546 for microgrid operation.

A1547 1. Publishers and subscribers define the communications end points.

A1548 2. Topics and instances define the number of samples.

A1549 3. Timing defines the rate of each sample.

A1550 4. Data types define the size of each sample.

A1551 These features should be used to derive hardware and software resource allocation  
A1552 requirements for individual devices. These include but are not limited to the following.

A1553 1. Network bandwidth.

A1554 2. Processor speed.

A1555 3. Memory capacity.

A1556 4. Log file storage.

A1557 Estimates may be made using static design information and verified using dynamic  
A1558 operating information. Additional resources should be reserved for future upgrades, including  
A1559 new microgrid features and cybersecurity protections.

A1560 Procurement documents will require the number of publishers and subscribers that each  
A1561 device must support. For example, see Table [A.13](#).

Table A.13: Example quantities for a small microgrid.

Device Role	Count
MC	2
MD	2
SRC	6
DIST	10
STOR	2
LOAD	10
CONV	2

## A.11 Data Model Extensions

Vendors may choose to extend the data model subject to the following constraints. A device does not conform to the present version of this standard if any of these constraints are not met.

### 1. Protocols.

- (a) New protocols must use different network ports than those used in this standard. Network utilization must be documented for non-interference.
- (b) Protocol changes must be backwards-compatible with the version used in this standard and should be approved by the appropriate governing body.

### 2. Data Flows.

- (a) Device roles shall not be granted new publication rights to existing command topics. This would violate the intentional separation of concerns.
- (b) New data flows are encouraged to re-use existing topics, subject to the previous constraint, as long as the timing and data types do not change.
- (c) New topics should be in a different domain, to avoid conflict with future versions of this standard.
- (d) Assigned data types should not be modified. Any new types must define and demonstrate interoperability with the standard types.
- (e) Timing and QoS should not be modified. Any such changes must be compatible with the full range of standard timing.
- (f) Data topics may be left unused, subject to the constraints given in Section [A.8.7](#)

### 3. Data Types.

- (a) New data types should be defined outside the “tms” namespace unless they are proposed for addition to this standard. Any proposed types may be modified or replaced by future versions of this standard.

A1587 (b) Existing data types should not be modified. Such changes may affect resource  
A1588 allocation and could conflict with future versions of this standard. The  
A1589 extensibility annotations are reserved solely for use by this standard. Any  
A1590 modification must be consistent with the existing annotations.

A1591 (c) New annotations may only be added if they interoperate with participants that  
A1592 do not have them.

A1593 Any device that extends the data model has the following additional requirements.

A1594 1. A simple on-device visual indicator when any extensions are enabled.

A1595 2. A simple on-device control to disable all extensions.

A1596 3. Publish a warning in ActiveDiagnostics. Details TBD.

A1597 4. Design and test artifacts that demonstrate non-interference with standard-conformant  
A1598 devices when the extensions are enabled.

A1599 5. Design artifacts explaining how extensions could be added to or interoperate with  
A1600 future versions of this standard. Any constraints imposed by the extension must be  
A1601 clearly documented.

A1602 Conformance testing will verify the accuracy of enabled indicator, effectiveness of disabled  
A1603 control, and the ability to fully meet all threshold requirements with the extensions disabled.  
A1604 Additional conformance testing may also be required to demonstrate interoperability with  
A1605 other devices that do not have the extensions.

A1606 B. Detailed Data Model Requirements

A1607 Domain-specific data types are defined as data type requirements focused on a specific  
A1608 functionality or expected behavior of the TMS. These are constructed data types of which  
A1609 the majority are either structure or typedef patterns.

A1610 The domain-specific data type requirements are grouped into units of functionality. Each  
A1611 functional group describes all of the device roles that may participate. See Section [B.30](#) for  
A1612 cross-references that organize this information by device role, topic name, and data type  
A1613 name.

A1614 Data flows from multiple functional groups may be used together to achieve higher-  
A1615 level system behaviors. See Section TBD of the Handbook for more information on system  
A1616 behaviors.

A1617 Table [B.1](#) is a summary of the topics defined in this data model. The data types described  
A1618 in this section execute the data topics.

Table B.1: Overview of All Data Model Topics

Topic	Participants							Section
	MD	MC	SRC	STOR	DIST	LOAD	CONV	
Heartbeat	$P_S$	$P_S$	$P_S$	$P_S$	$P_S$	$P_S$	$P_S$	B.4.1
DeviceAnnouncement	$P_S$	$P_S$	$P_S$	$P_S$	$P_S$	$P_S$	$P_S$	B.5.1
DeviceIcon	$P_S$	$P$	$P$	$P$	$P$	$P$	$P$	B.7.1
FingerprintNickname	$P_S$	$P$	$P$	$P$	$P$	$P$	$P$	B.6.1
FingerprintNicknameRequest	$P_S$	$S$	$S$	$S$	$S$	$S$	$S$	B.6.1
OperatorConnectionList	$P$	$S$						B.22.1
DiscoveredConnectionList		$S$	$P$	$P$	$P$	$P$	$P$	B.22.1
MicrogridConnectionList	$S$	$P$						B.22.1
ActiveDiagnostics	$P_S$	$P_S$	$P$	$P$	$P$	$P$	$P$	B.12.1
DeviceClockStatus	$P_S$	$P_S$	$P$	$P$	$P$	$P$	$P$	B.26.1
StandardConfigMaster	$S$	$S$	$P$	$P$	$P$	$P$	$P$	B.13.1
DevicePowerMeasurementList	$S$	$S$	$P$	$P$	$P$	$P$	$P$	B.20.1
DevicePowerStatusList	$S$	$S$	$P$	$P$	$P$	$P$	$P$	B.19.1
PowerSwitchRequest		$P$			$S$		$S$	B.19.1
SourceTransitionState	$S$	$S$	$P$	$P$				B.18.1
SourceTransitionRequest		$P$	$S$	$S$				B.18.1
LoadSharingStatus	$S$	$S$	$P$	$P$			$P$	B.23.1
LoadSharingRequest		$P$	$S$	$S$			$S$	B.23.1
ControlHardwareStatus	$P_S$	$P_S$	$P$	$P$	$P$	$P$	$P$	B.8.1
PowerHardwareStatus	$S$	$S$	$P$	$P$	$P$	$P$	$P$	B.9.1
StorageState	$S$	$S$		$P$				B.10.1
StorageControlStatus	$S$	$S$		$P$				B.29.1
StorageControlRequest		$P$		$S$				B.29.1
Reply	$P_S$	$P_S$	$P$	$P$	$P$	$P$	$P$	B.29.1
ControlParameterStatus	$S$	$S$	$P$	$P$			$P$	B.16.1
ControlParameterRequest		$P$	$S$	$S$			$S$	B.16.1
MetricParameterStatus	$P_S$	$S$	$P$	$P$			$P$	B.16.1
AuthorizationToEnergizeResponse	$P_S$		$S$	$S$	$S$			B.27.1
AuthorizationToEnergizeRequest	$S$		$P$	$P$	$P$		$P$	B.27.1
AuthorizationToEnergizeOutcome	$S$		$P$	$P$	$P$		$P$	B.27.1
OperatorIntentState	$P_S$	$P$						B.11.1
OperatorIntentRequest	$P$	$S$						B.11.1
GetConfigContentsResponse		$S$	$P$	$P$	$P$	$P$	$P$	B.14.1
CopyConfigRequest		$P$	$S$	$S$	$S$	$S$	$S$	B.15.1
GetConfigContentsRequest		$P$	$S$	$S$	$S$	$S$	$S$	B.14.1
MicrogridMembershipRequest	$S$	$S$	$P$	$P$	$P$	$P$	$P$	B.28.1
MicrogridMembershipOutcome		$P$	$S$	$S$	$S$	$S$	$S$	B.28.1
DeviceGroundingStatus	$S$	$S$	$P$	$P$	$P$	$P$	$P$	B.17.1
GroundingCircuitRequest		$P$	$S$	$S$	$S$	$S$	$S$	B.17.1
DcDevicePowerMeasurementList	$S$	$S$	$P$	$P$	$P$	$P$	$P$	B.21.1
DcLoadSharingStatus	$S$	$S$	$P$	$P$			$P$	B.24.1
DcLoadSharingRequest		$P$	$S$	$S$			$S$	B.24.1

## A1619 B.1 Device Role Schematics

A1620 As described in Section A.6, each TMS device is assigned a single specific device role in the  
A1621 microgrid, which then defines the messaging services of the device.

A1622 The diagrams and tables in this section describe how the device roles are represented in  
A1623 the data model.

### A1624 B.1.1 Source Device Role

A1625 Figure B.1 shows the major hardware components in the Source (SRC) device role. This  
A1626 schematic is intended to name components and show how they are usually connected. It is  
A1627 not intended to require any particular implementation. In particular, not all components  
A1628 are required and substitutions of equivalent components are allowed.

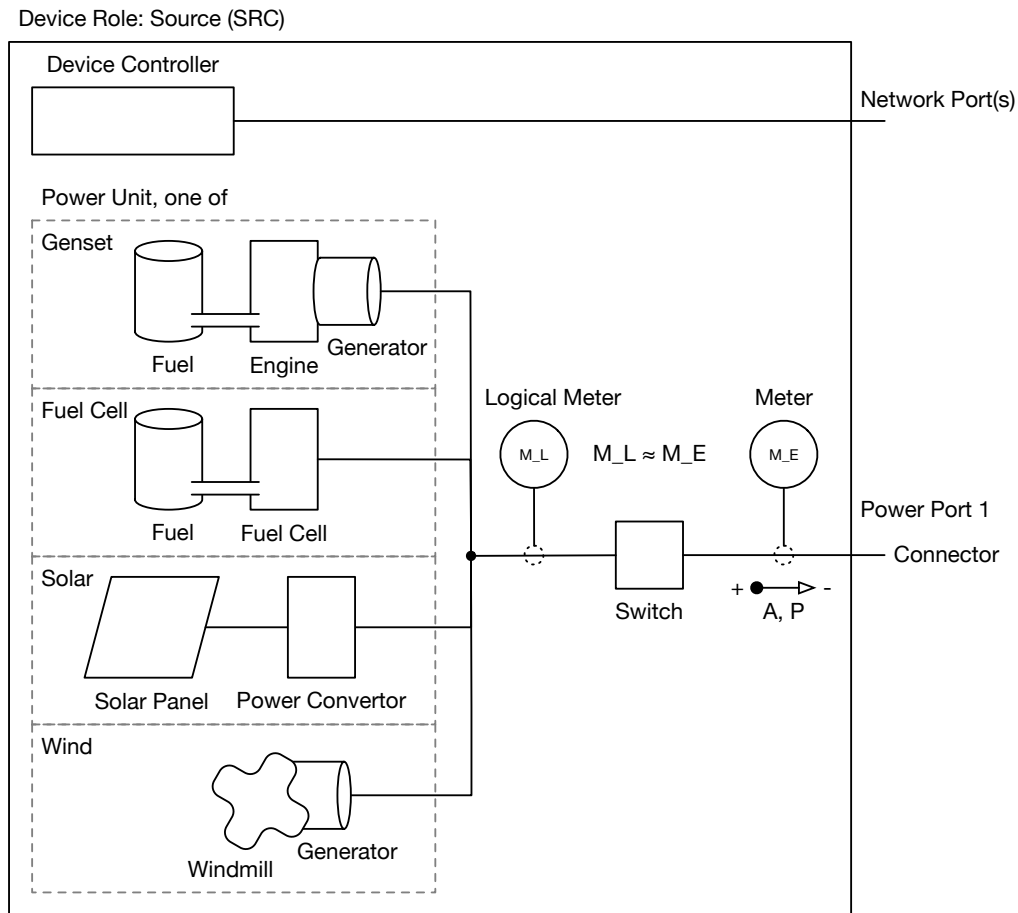


Figure B.1: Schematic diagram of the Source device role.

A1629 Every source device shall have a Device Controller that connects to the TMS network.  
A1630 This is the entry point to the data model. The Device Controller's function is to send and  
A1631 receive messages, translating as necessary between external messages and internal signals.



Every source device shall have at least one network port. More network ports may be required to allow for connections to multiple TMS devices.

Every source device shall have some form of power unit, usually a generator set (genset), fuel cell, solar, or wind power source. At present, the data model provides the most detailed support for gensets. More support for other power units may be added in future versions of this standard. The data model provides support for monitoring the energy source (fuel level) and also for starting, stopping, and regulating the amount of power produced.

Every source device shall have a single power port with an external connector for other devices. If additional power ports are required, then the device controller should present both the source and distribution device roles, as described in Section [A.6.3](#). The data model provides support for monitoring the connections between devices.

The power port may have a switch that isolates the internal components from the external connector. Power measurements are made on the external side of this switch, and optionally, also on the internal side. The data model provides support for opening and closing the switch, measuring power, and synchronizing waveforms. Not shown, every source device may have a grounding circuit, with its associated metering and switching capabilities.

Table [B.2](#) identifies sections of the data model that pertain to each component of the source device role.

Table B.2: Data Model Sections for Source Role Components

Source Component	Data Model Sections
Device Controller	Identity (B.3) Heartbeat (B.4) Device Announcement (B.5) Nickname (B.6) Device Icon (B.7) Control Hardware Status (B.8) Microgrid Controller Selection (B.13) Read Configuration Settings (B.14) Copy Configuration Settings (B.15) Device Parameters (B.16) Clock Status (B.26) Black Start (B.27) Microgrid Membership (B.28)
Power Unit	Power Hardware Status (B.9) Diagnostics Reporting (B.12) Grounding Circuits (B.17) Remote Start/Stop (B.18) AC Load Sharing (B.23) DC Load Sharing (B.24)
Meters	AC Measurements (B.20) DC Measurements (B.21)
Switch & Connector	Power Switches and Connectors (B.19) Power Topology (B.22)

A1650 B.1.2 Distribution Device Role

A1651 Figure B.2 shows the major hardware components in the Distribution (DIST) device role.  
A1652 This schematic is intended to name components and show how they are usually connected. It  
A1653 is not intended to require any particular implementation. In particular, not all components  
A1654 are required, and substitutions of equivalent components are allowed.

A1655 *Editorial Note: This section is work in progress. See previous Source Device Role, Section*  
A1656 *B.1.1 for an example.*

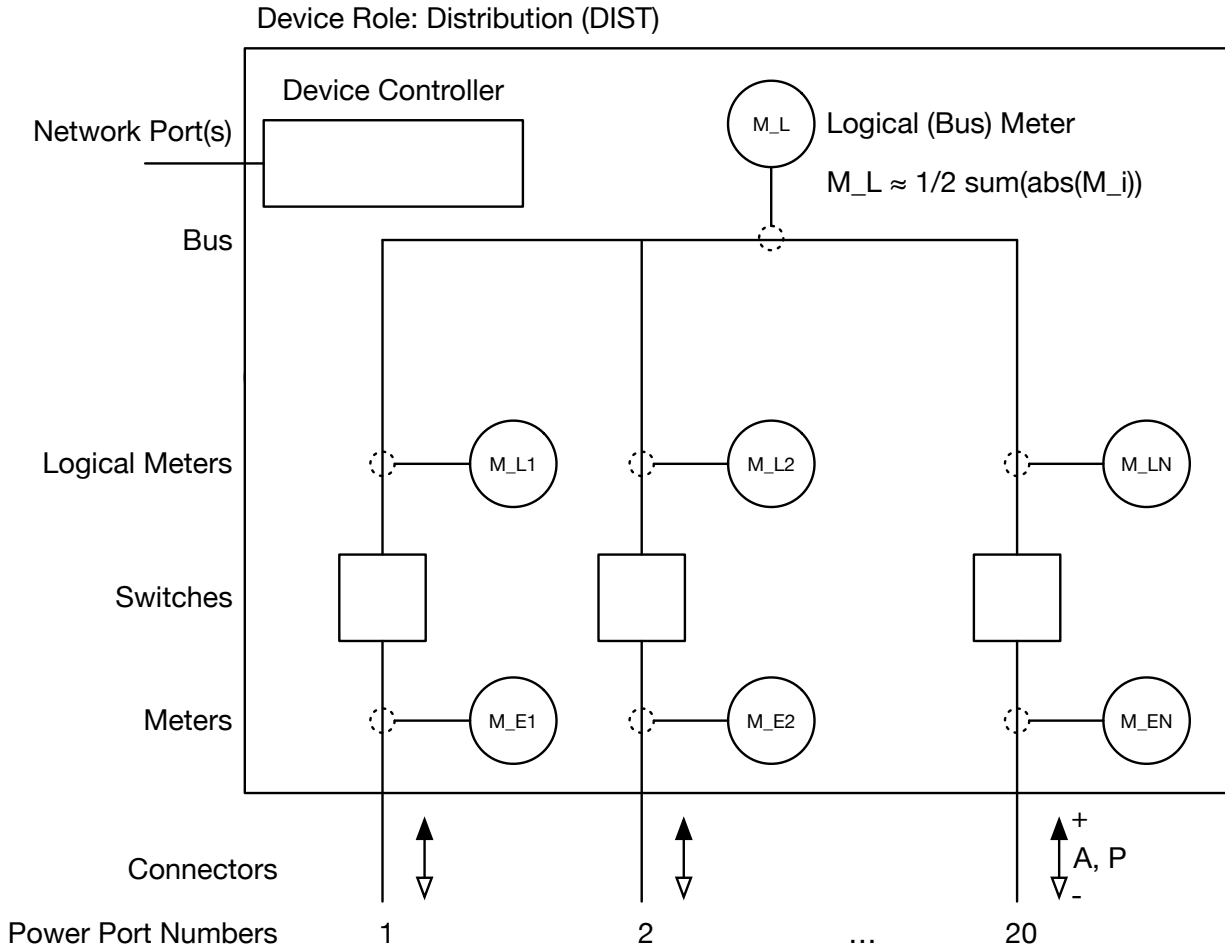


Figure B.2: Schematic diagram of the Distribution device role.

### A1657 B.1.3 Storage Device Role

A1658 Figure B.3 shows the major hardware components in the Storage (STOR) device role. This  
A1659 schematic is intended to name components and show how they are usually connected. It is  
A1660 not intended to require any particular implementation. In particular, not all components  
A1661 are required, and substitutions of equivalent components are allowed.

A1662 *Editorial Note: This section is work in progress. See previous Source Device Role, Section*  
A1663 *B.1.1 for an example.*

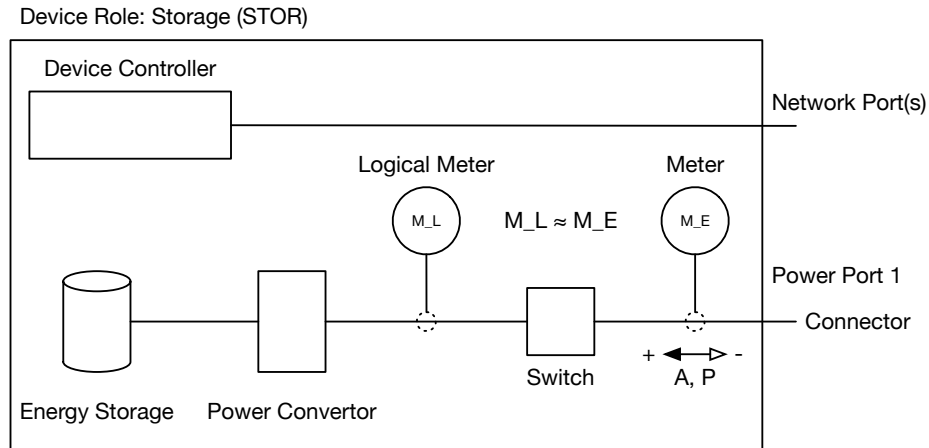


Figure B.3: Schematic diagram of the Storage device role.

#### B.1.4 Load Device Role

Figure B.4 shows the major hardware components in the Load (LOAD) device role. This schematic is intended to name components and show how they are usually connected. It is not intended to require any particular implementation. In particular, not all components are required, and substitutions of equivalent components are allowed.

*Editorial Note: This section is work in progress. See previous Source Device Role, Section B.1.1 for an example.*

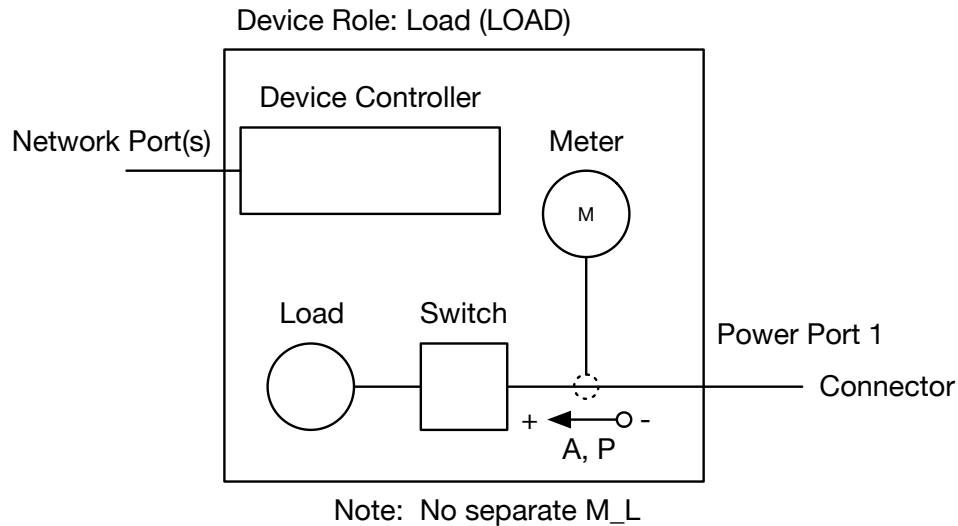


Figure B.4: Schematic diagram of the Load device role.

#### B.1.5 Conversion Device Role

Figure B.5 shows the major hardware components in the Conversion (CONV) device role. This schematic is intended to name components and show how they are usually connected. It



4. Status. As the request is executed, send event-based or periodic updates of its progress. Most updates are via the related status topic. Some updates may use **ActiveDiagnostics** or another topic.

Every request is addressed to a specific key value instance, as described in Section A.8.4, and it contains a **RequestSequence** that increments for each request. Taken together, the request key value and sequence number uniquely identify each request and identify the order in which they were issued.

Some requests are for immediate action and others are for deferred action. The time when the request should be processed is indicated by its **ConfigId**.

Configuration ID:

- **CONFIG\_DEFAULTS**. Manufacturer default settings. Read-only. A factory reset copies these values over any other saved requests.
- **CONFIG\_ACTIVE**. Request having immediate effect. Often called a command. Most may be saved in volatile memory. Requests that must be saved in persistent memory are clearly marked.
- **CONFIG\_ON\_REBOOT**. Request for effect after the device restarts or reboots and before communications are established. Must be saved in persistent memory.
- **CONFIG\_ON\_COMMS\_LOSS**. Request for effect after communications loss timeout. May be saved in volatile memory.
- **CONFIG\_ON\_COPY**. Requests for effect after they are copied to another request type. This temporary location may be used to queue up and issue multiple requests simultaneously or to validate a request without modifying the state of the target device.

### B.2.1 Data Flows

There are no data flows specific to this pattern. The data flow shown below is a template for all request/response data flows. The wildcard **\*Request** represents any request topic and the **Reply** represents the default reply topic.

Figure B.6 illustrates all the topics in the data flow.

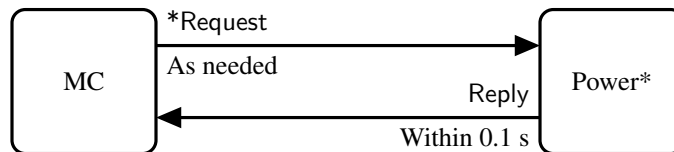


Figure B.6: Request/Response Pattern data flow

Most requests are from the control services to the power devices and no type-specific return value is expected. All such requests share the same default **RequestResponse**. For requests that have other needs, another response topic is specified in the data flow.

A1722 Table B.3 provides an overview of how each topic is used.

Table B.3: Description of the Request/Response Pattern topics.

Topic	Description
*Request	
Reply	The reply to a request.

A1723 Table B.4 specifies the publishers and subscribers for each topic as well as usage  
A1724 requirements as described in Section A.8.7.

Table B.4: Participants of the Request/Response Pattern topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
*Request									
Reply	REQUIRED	Pub Sub	Pub Sub	Pub Sub	Pub	Pub	Pub	Pub	Pub

A1725 Table B.5 specifies the timing for when sample updates are sent to each topic. These  
A1726 parameters are defined in Section A.8.5.

Table B.5: Timing of the Request/Response Pattern topics.

Topic	Data Trigger	Rate Class	Burst Size	QoS Profile
*Request				
Reply	Within 0.1 s	All	1	Response

## A1727 B.2.2 Data Types

A1728 Table B.6 specifies the data types that are used as sample values for each topic.

Table B.6: Data types of the Request/Response Pattern topics.

Topic	Data Type
*Request	
Reply	<code>tms.Reply</code>

A1729 Topic-level data types are defined in the order shown in Table B.6. Then nested data  
A1730 types are defined in the order of appearance. Cross-references are provided for data types  
A1731 that have already been defined.

A1732 The data types are listed in the order shown in the specifications table above. Nested  
A1733 data types are displayed in the order of appearance. Data types that have been described

in previous sections are referenced to the section containing the data type description.

Some requests require a source-specific return value. Those requests have custom `< Type >Request` and `< Type >Response` messages defined for the request `< Type >`. Many other requests trigger a change in state that is reported using another data object. This response structure is used for such requests. In effect, the request triggers two messages to be sent: the updated data value and a `RequestResponse`.

#### B.2.2.1 `tms.Reply`

**PURPOSE:** The outcome of a request. Supported requests include `GridRequest`, `DeviceRequest`, `DeviceConfigRequest`, and `PortConfigRequest`.

**TOPIC USAGE:** Reply

**EXTENSIBILITY:** extensibility(APPENDABLE)

**PATTERN:** Structure

**ATTRIBUTES:**

Name	Type and Description
<code>requestingDeviceId</code>	<p><b>Fingerprint</b> Identity of the device that sent this request. Annotations: <i>keyval</i></p>
<code>targetDeviceId</code>	<p><b>Fingerprint</b> Identity of the device that received the request. Default to NULL Fingerprint. Annotations: <i>keyval</i></p>
<code>config</code>	<p><b>ConfigId</b> Power device configuration that this request belongs to. Default to 0. Annotations: <i>keyval</i></p>
<code>portNumber</code>	<p><b>PowerPortNumber</b> Power port number that should change state. Default to 0. Annotations: <i>keyval</i></p>
<code>requestSequenceId</code>	<p><b>RequestSequence</b> Copy of the RequestSequence data from the processed request.</p>
<code>status</code>	<p><b>ReplyStatus</b> Indication of success or failure.</p>



A1750 B.2.2.2 tms.Fingerprint

A1751 PURPOSE: Short representation of PublicKey, suitable for uniquely identifying a device,  
A1752 person, platform, or other entity. A value of all 0s represents an invalid fingerprint

A1753 EXTENSIBILITY: extensibility(APPENDABLE)

A1754 PATTERN: Typedef

A1755 ORIGINAL TYPE: `octet[32]`

A1756 From highest to lowest priority:

- A1757 1. Mechanism defined in the procurement specification.
- A1758 2. Cryptographic hash of the device's public key (X.509 certificate). Format is "...".
- A1759 3. MAC address of the primary Ethernet interface. Format is "mac:...".

A1760 B.2.2.3 tms.ConfigId

A1761 PURPOSE: Identifies the time when configuration settings are to be used.

A1762 EXTENSIBILITY: extensibility(APPENDABLE)

A1763 PATTERN: Enumeration

A1764 VALUES:

Name	Description
<hr/>	
CONFIG_DEFAULTS	Read-only manufacturer defaults.
CONFIG_ACTIVE	Settings for use at the present time.
A1765 CONFIG_ON_REBOOT	Settings for use after a reboot / restart.
CONFIG_ON_COMMS_LOSS	Settings for use after a communications loss timeout.
CONFIG_ON_COPY	Settings for use after a CopyConfigRequest.
<hr/>	

A1767 B.2.2.4 tms.PowerPortNumber

A1768 PURPOSE: Identifies a port within a device.

A1769 EXTENSIBILITY: extensibility(APPENDABLE)

A1770 PATTERN: Typedef

A1771 ORIGINAL TYPE: uint32

A1772 CONSTANTS:

Name	Type [Units], Value, and Description
<hr/> MAX_PORTS	
	uint32, 64
A1773	Maximum number of power ports that a device may contain.
ONLY_PORT	
	uint32, 0
A1774	Number to use for a single-port device with no listed number.

A1775 B.2.2.5 tms.RequestSequence

A1776 PURPOSE: Allows Request / Reply topics to be paired together.

A1777 TOPIC USAGE: Nested

A1778 EXTENSIBILITY: extensibility(APPENDABLE)

A1779 PATTERN: Structure

A1780 ATTRIBUTES:

Name	Type and Description
<hr/> sequenceNumber	
A1781	uint64
A1782	Requester assigned sequence number sent in a request and returned in a reply.

A1783 B.2.2.6 tms.ReplyStatus

A1784 PURPOSE: Indicate the type of success or failure of a request.

A1785 TOPIC USAGE: Nested

A1786 EXTENSIBILITY: extensibility(APPENDABLE)

A1787 PATTERN: Structure

A1788 ATTRIBUTES:

Name	Type and Description
------	----------------------

---

<b>code</b>	
-------------	--

<b>code</b>	<b>uint32</b>
-------------	---------------

<b>code</b>	Indicator of success or failure. Intended to support automatic handling. Values from 100 to 599 are as defined in the IETF RFCs. Values outside that range are reserved for future versions of this standard. Selected values are defined as constants.
-------------	---

<b>reason</b>	
---------------	--

<b>reason</b>	<b>String32</b>
---------------	-----------------

<b>reason</b>	Short textual description of the status code intended for human operators.
---------------	--

---

A1790 CONSTANTS:

*TMS Data Model*  
*DRAFT 19 Apr 2021*

Name	Type [Units], Value, and Description
------	--------------------------------------

---

REPLY_OK	
----------	--

	uint32, 200
--	-------------

	The request has succeeded.
--	----------------------------

REPLY_BAD_REQUEST	
-------------------	--

	uint32, 400
--	-------------

	Some value in the request was invalid.
--	--

REPLY_METHOD_NOT_ALLOWED	
--------------------------	--

	uint32, 405
--	-------------

REPLY_CONFLICT	
----------------	--

	uint32, 409
--	-------------

REPLY_GONE	
------------	--

	uint32, 410
--	-------------

REPLY_PRECONDITION_FAILED	
---------------------------	--

	uint32, 412
--	-------------

REPLY_REQUEST_ENTITY_TOO_LARGE	
--------------------------------	--

	uint32, 413
--	-------------

REPLY_INTERNAL_SERVER_ERROR	
-----------------------------	--

	uint32, 500
--	-------------

REPLY_NOT_IMPLEMENTED	
-----------------------	--

	uint32, 501
--	-------------

REPLY_SERVICE_UNAVAILABLE	
---------------------------	--

	uint32, 503
--	-------------

REPLY_PENDING_AUTHORIZATION	
-----------------------------	--

	uint32, 600
--	-------------

	Request is valid, and authorization is required before processing.
--	--

---

DESCRIPTION:

Based on the "Status Code and Reason Phrase" defined in RFC2616 and updated as the "Status Line" in RFC 7230.

A1796 B.2.2.7 tms.String32

A1797 PURPOSE: A variable length string.

A1798 EXTENSIBILITY: extensibility(FINAL)

A1799 PATTERN: Typedef

A1800 ORIGINAL TYPE: `string<0,32>`

A1801 B.3 Identity

A1802 Each TMS device has a unique device identity, as discussed in Section [A.5](#), which is bound to  
A1803 a single specific device role with a defined set of requirements and permissions. In addition,  
A1804 each device role has its own unique **Fingerprint** to differentiate multiple devices assigned  
A1805 to the same device role.

A1806 Identity is also used to establish trust for all devices. Trust is a precondition of  
A1807 communications in the tactical microgrid, built upon the concepts of device identity and  
A1808 approval by an authorized operator. Each TMS device requires an authorized operator to  
A1809 assign the device identity, allowing devices to uniquely specify the source or destination of a  
A1810 message.

A1811 Trust is established when devices provide evidence that their identity has been approved  
A1812 by an authorized operator. This evidence is provided through defined protocols that prevent  
A1813 unauthorized devices from forging approval. Once trust is established, authorized operators  
A1814 then share credentials with other devices to approve them. Authorized operators have  
A1815 privileged credentials and physical access that allows them to manage trust between devices.

A1816 Operator roles are used to control network access to TMS devices. Operators may be  
A1817 granted access to specific devices or groups of devices on the grid. The operator's ability to  
A1818 interact with TMS devices is constrained by the assigned access. An *authorized operator* is  
A1819 granted access to control all normal operating modes of the device(s). Additional operator  
A1820 roles may include an *Administrator*, granted elevated device privileges to control access rights  
A1821 and operating bounds of other users, and *Technician*, granted limited device access to enable  
A1822 diagnostics and other functionality in support of device maintenance and repair.

A1823 Note: The messages in this section rely on lower-level protocol to provide secure  
A1824 communications using standard protocols, such as SSH and TLS. Full cryptographic protocol  
A1825 is beyond the scope of this document.

A1826 There are no data flows associated with this pattern.

A1827

A1828 B.3.1 Data Types

A1829 The following data type is used to implement this topic.

A1830

A1831 B.3.1.1 tms.Fingerprint

A1832 See definition in Section [B.2.2.2](#).

## A1833 B.4 Heartbeat

A1834 The heartbeat provides a periodic announcement that identifies the device as connected and  
A1835 functional, asserting basic functionality only. The intent is to provide notice of the device's  
A1836 awareness and connection.

A1837

### A1838 B.4.1 Data Flows

A1839 Figure B.7 illustrates all the topics in the data flow.

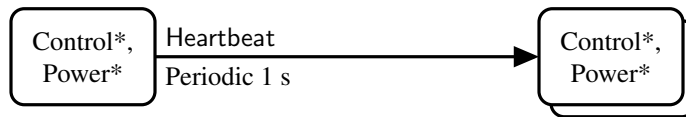


Figure B.7: Heartbeat data flow

A1840 Each device publishes a new **Heartbeat** once per second.

A1841

A1842 Table B.7 provides an overview of how each topic is used.

Table B.7: Description of the Heartbeat topics.

Topic	Description
Heartbeat	Liveliness heartbeat for all devices.

A1843 Table B.8 specifies the publishers and subscribers for each topic as well as usage  
A1844 requirements as described in Section A.8.7.

Table B.8: Participants of the Heartbeat topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
Heartbeat	REQUIRED	Pub Sub	Pub Sub	Pub Sub	Pub Sub	Pub Sub	Pub Sub	Pub Sub	Pub Sub

A1845 Table B.9 specifies the timing for when sample updates are sent to each topic. These  
A1846 parameters are defined in Section A.8.5.

Table B.9: Timing of the Heartbeat topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
Heartbeat	Periodic 1 s	1 s	1			Medium

## B.4.2 Data Types

Table B.10 specifies the data types that are used as sample values for each topic.

Table B.10: Data types of the Heartbeat topics.

Topic	Data Type
Heartbeat	<code>tms.Heartbeat</code>

Topic-level data types are defined in the order shown in Table B.10. Then nested data types are defined in the order of appearance. Cross-references are provided for data types that have already been defined.

### B.4.2.1 `tms.Heartbeat`

PURPOSE: Periodic indication of device availability.

TOPIC USAGE: Heartbeat

EXTENSIBILITY: extensibility(APPENDABLE)

PATTERN: Structure

ATTRIBUTES:

Name	Type and Description
<code>deviceId</code>	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
<code>sequenceNumber</code>	<b>uint32</b> A counter that starts at 0 and increments by 1 for each new heartbeat.

DESCRIPTION:

Network communications generally provides a low-level indication whether two devices are connected. The periodic update of `Heartbeat` with an incrementing `sequenceNumber` confirms availability of the application software.

### B.4.2.2 `tms.Fingerprint`

See definition in Section B.2.2.2.

## B.5 Device Announcement

Device Announcements is the discovery process of establishing communications and announcing the availability of a device. This process communicates the identity, type,

A1869 ratings, interfaces, and liveliness of the device. This information enables the rapid, ad-  
A1870 hoc deployment of trusted TMS devices. The discovery sequence starts when the device is  
A1871 powered on, connected to the network, and authorized by the operator.

### A1872 B.5.1 Data Flows

A1873 The device announcement message acts as a digital faceplate, providing information about  
A1874 the device to allow the microgrid controller to interact with it. The message is sent during  
A1875 startup, after trusted communication with the TMS is established, and describes the device's  
A1876 identity, type, ratings, and interfaces available. Essentially, it is an announcement that the  
A1877 device is available to the network. For the human operator, the device announcement also  
A1878 confirms that the digital faceplate matches the physical device in the microgrid. Device  
A1879 Announcements also remain available for other devices that connect at a later time.

A1880

A1881 Figure B.8 illustrates all the topics in the data flow.

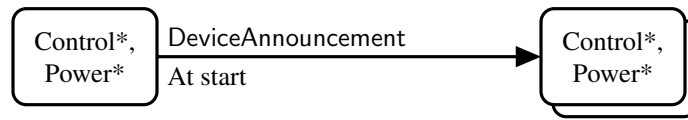


Figure B.8: Device Announcement data flow

A1882 When each device starts, it publishes a **DeviceAnnouncement** so it can be discovered by  
A1883 neighboring devices. This registers it with the MC and causes it to appear on the MD. Power  
A1884 devices use this to find MCs to follow.

A1885 For information on the additional messages sent during startup that are not covered in  
A1886 this section, refer to the Device Power Port List and Device Grounding data types.

A1887

A1888 Table B.11 provides an overview of how each topic is used.

Table B.11: Description of the Device Announcement topics.

Topic	Description
DeviceAnnouncement	Product information for the device. Used to discover TMS compliant devices.

A1889 Table B.12 specifies the publishers and subscribers for each topic as well as usage  
A1890 requirements as described in Section A.8.7.

Table B.12: Participants of the Device Announcement topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DeviceAnnouncement	REQUIRED	Pub Sub	Pub Sub	Sub	Pub Sub	Pub Sub	Pub Sub	Pub Sub	Pub Sub



A1891 Table B.13 specifies the timing for when sample updates are sent to each topic. These  
A1892 parameters are defined in Section A.8.5.

Table B.13: Timing of the Device Announcement topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
		Class	Size	
DeviceAnnouncement	At Start	Once	1	PublishLast

### A1893 B.5.2 Data Types

A1894 Table B.14 specifies the data types that are used as sample values for each topic.

Table B.14: Data types of the Device Announcement topics.

Topic	Data Type
DeviceAnnouncement	tms.DeviceInfo

A1895 Topic-level data types are defined in the order shown in Table B.14. Then nested data  
A1896 types are defined in the order of appearance. Cross-references are provided for data types  
A1897 that have already been defined.

A1898 B.5.2.1 tms.DeviceInfo

A1899 PURPOSE: Design information for the device.

A1900 TOPIC USAGE: DeviceAnnouncement

A1901 EXTENSIBILITY: extensibility(APPENDABLE)

A1902 PATTERN: Structure

A1903 ATTRIBUTES:

Name	Type and Description
<code>deviceId</code>	<p><b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i></p>
<code>platformId</code>	<p><b>Fingerprint</b> A platform is a group of TMS devices that are assembled into a larger physical package. Platforms may integrate multiple device roles. Set to FINGERPRINT_INVALID if this device is not permanently mounted on a platform.</p>
<code>role</code>	<p><b>DeviceRole</b> TMS role implemented by this device.</p>
<code>product</code>	<p><b>ProductInfo</b> Product information.</p>
<code>topics</code>	<p><b>TopicInfo</b> communications interfaces implemented by this device</p>
<code>controlHardware</code>	<p><b>optional&lt;ControlHardwareInfo&gt;</b> Information about the device controller. May apply to both control services and power devices.</p>
<code>powerHardware</code>	<p><b>optional&lt;PowerHardwareInfo&gt;</b> Information about the power hardware, if applicable.</p>
<code>controlParameters</code>	<p><b>ParameterMetadataSequence</b> A full list of read-write control parameters available on the ControlParameterStatus topic.</p>
<code>metricParameters</code>	<p><b>ParameterMetadataSequence</b> A full list of read-only metric parameters available on the MetricParameterStatus topic.</p>
<code>controlService</code>	<p><b>optional&lt;ControlServiceInfo&gt;</b> Control role-specific information.</p>
<code>powerDevice</code>	<p><b>optional&lt;PowerDeviceInfo&gt;</b> Power device role-specific information.</p>

#### B.5.2.2 tms.Fingerprint

See definition in Section [B.2.2.2](#).

A1908 B.5.2.3 tms.DeviceRole

A1909 PURPOSE: Identify the role filled by a device.

A1910 EXTENSIBILITY: extensibility(APPENDABLE)

A1911 PATTERN: Enumeration

A1912 VALUES:

Name	Description
ROLE_MICROGRID_CONTROLLER	Microgrid controller, a control service.
ROLE_SOURCE	Source power device.
ROLE_LOAD	Load power device.
ROLE_STORAGE	Storage power device.
ROLE_DISTRIBUTION	Distribution power device.
ROLE_MICROGRID_DASHBOARD	Microgrid dashboard, a control service.
ROLE_CONVERSION	Conversion power device.
ROLE_MONITOR	Monitor microgrid, read-only access.

A1915 B.5.2.4 tms.ProductInfo

A1916 PURPOSE: Design information for the device.

A1917 TOPIC USAGE: Nested

A1918 EXTENSIBILITY: extensibility(APPENDABLE)

A1919 PATTERN: Structure

A1920 ATTRIBUTES:

Name	Type and Description
nsn	NatoStockNumber Supply type of this device.
gtin	GlobalTradeItemNumber Product type of this device.
manufacturerName	String32 Name of the device manufacturer.
modelName	String32 Name of the device model. Optional.
modelNumber	String32 Manufacturer number for the device model. Optional.
serialNumber	String32 Unique production number for a specific device. Large enough to contain a variety of formats, including the GS1 serial number (AI 21). Optional.
softwareVersion	String32 Software Version.

A1923 DESCRIPTION:

A1924 This information is intended to aid in logistics, operator interaction, maintenance, logging,  
A1925 and other tasks. Some of this information should be presented to the operator, so they  
A1926 can confirm the microgrid configuration. Some of this information should be used to  
A1927 automatically validate the values in other messages.

A1928 B.5.2.5 tms.NatoStockNumber

A1929 PURPOSE: Encodes a NATO Stock Number (NSN), aka National Stock Number.

A1930 EXTENSIBILITY: extensibility(APPENDABLE)

A1931 PATTERN: Typedef

A1932 ORIGINAL TYPE: `char[13]`

A1933 DESCRIPTION:

A1934 A NATO Stock Number (NSN) identifies the supply type (form, fit, and function) of a device.  
A1935 NSN codes are managed according to the NATO Standardization Agreements. The NSN  
A1936 is stored without dashes or other delimiters and is encoded as characters. The NSN value  
A1937 000000000000 is reserved as invalid/not available.

A1938 Note: These numbers are designed for human interpretation so do not follow normal algebraic  
A1939 conventions and in some cases, extensions may use characters instead of numbers.

A1940 B.5.2.6 tms.GlobalTradeItemNumber

A1941 PURPOSE: Encodes a Global Trade Item Number (GTIN).

A1942 EXTENSIBILITY: extensibility(APPENDABLE)

A1943 PATTERN: Typedef

A1944 ORIGINAL TYPE: `char[14]`

A1945 DESCRIPTION:

A1946 A Global Trade Item Number (GTIN) identifies the actual product type of a device and are  
A1947 managed by [GS1](#).

A1948 As specified by GS1, shorter GTINs, such as the 12-character UPC code, must be prefixed  
A1949 by leading zero digits. If a GTIN has not been assigned, then Local Assigned Codes (LACs)  
A1950 must be used, for example, by using a seven-zero prefix. All characters in the GTIN must  
A1951 be a digit, 0 to 9. See [GS1 General Specifications](#) for more details.

A1952 B.5.2.7 tms.String32

A1953 See definition in Section [B.2.2.7](#).

A1954 B.5.2.8 tms.TopicInfo

A1955 PURPOSE: Topic information. Defines which communications interfaces a given device  
A1956 implements.

A1957 TOPIC USAGE: Nested

A1958 EXTENSIBILITY: extensibility(APPENDABLE)

A1959 PATTERN: Structure

A1960 ATTRIBUTES:

Name	Type and Description
<hr/>	
dataModelVersion	
String32	
	Release version of data topics and types.
conditionalTopics	
TopicConditionSequence	
A1961	List of conditional topics.
optionalTopics	
TopicConditionSequence	
	List of optional topic names that are implemented by this device.
numNetworkPorts	
uint16	
A1962	Number of network port connectors.
<hr/>	

A1963 B.5.2.9 tms.TopicConditionSequence

A1964 PURPOSE: A sequence of TopicCondition.

A1965 EXTENSIBILITY: extensibility(FINAL)

A1966 PATTERN: Typedef

A1967 ORIGINAL TYPE: `sequence<String1_32,0,5>`

A1968 B.5.2.10 tms.String1-32

A1969 PURPOSE: A variable length string with a fixed minimum.

A1970 EXTENSIBILITY: extensibility(FINAL)

A1971 PATTERN: Typedef

A1972 ORIGINAL TYPE: `string<1,32>`

A1973 B.5.2.11 tms.ControlHardwareInfo

A1974 PURPOSE: Design information for control hardware.

A1975 TOPIC USAGE: Nested

A1976 EXTENSIBILITY: extensibility(APPENDABLE)

A1977 PATTERN: Structure

A1978 ATTRIBUTES:

Name	Type and Description
<code>hasRealtimeClock</code>	<code>boolean</code> True indicates that the device has a real time clock. False otherwise.
<code>minTemperature</code>	<code>float32</code> Minimum rated operating temperature. Annotations: <i>min</i> =-273, <i>max</i> =10000, <i>units</i> =Celsius
<code>maxTemperature</code>	<code>float32</code> Maximum rated operating temperature. Annotations: <i>min</i> =-273, <i>max</i> =10000, <i>units</i> =Celsius

A1979 Minimum rated operating temperature.

Annotations: *min*=-273,*max*=10000,*units*=Celsius

Maximum rated operating temperature.

Annotations: *min*=-273,*max*=10000,*units*=Celsius

A1980



A1981 B.5.2.12 tms.PowerHardwareInfo

A1982 PURPOSE: Design information for power hardware. Power hardware refers to accessories  
A1983 or supporting components of a power device, such as an engine, control boards, etc.

A1984 TOPIC USAGE: Nested

A1985 EXTENSIBILITY: extensibility(APPENDABLE)

A1986 PATTERN: Structure

A1987 ATTRIBUTES:

Name	Type and Description
engine	optional<EngineInfo> Engine information.
fuel	optional<FuelInfo> Fuel information.
generator	optional<GeneratorInfo> Generator information.
energyStorage	optional<EnergyStorageInfo> Energy storage information.
powerElectronics	optional<PowerElectronicsInfo> Power electronics information.
transformer	optional<TransformerInfo> Transformer information.
thermal	optional<ThermalInfo> Thermal zone information.

A1990 B.5.2.13 tms.EngineInfo

A1991 PURPOSE: Design information for an engine, such as on a generator set (genset).

A1992 TOPIC USAGE: Nested

A1993 EXTENSIBILITY: extensibility(APPENDABLE)

A1994 PATTERN: Structure

A1995 ATTRIBUTES:

Name	Type and Description
minOilPressure	float32 Minimum oil pressure for the operation of the engine that prevents engine damage. Annotations: <i>units</i> =pascal
maxOilPressure	float32 Maximum oil pressure for the operation of the engine that prevents engine damage. Annotations: <i>units</i> =pascal
minCoolantTemperature	float32 Minimum coolant temperature for the operation of the engine that prevents engine damage. Annotations: <i>units</i> =Celsius
maxCoolantTemperature	float32 Maximum coolant temperature for the operation of the engine that prevents engine damage. Annotations: <i>units</i> =Celsius
minEngineSpeed	float32 Stall warning speed of the engine. Speeds below this threshold may cause a malfunction or protective shutdown. Annotations: <i>units</i> =radian per second
maxEngineSpeed	float32 Maximum engine speed. I.e. the red line Annotations: <i>units</i> =radian per second
minWetStackPreventionLoad	float32 Minimum load required to prevent wet stack build up. Annotations: <i>units</i> =watt
minWetStackMitigationLoad	float32 Minimum load required to clear wet stack build up. Annotations: <i>units</i> =watt

A1997

A1998 B.5.2.14 tms.FuelInfo

A1999 PURPOSE: Design information for the fuel associated with a power device.

A2000 TOPIC USAGE: Nested

A2001 EXTENSIBILITY: extensibility(APPENDABLE)

A2002 PATTERN: Structure

A2003 ATTRIBUTES:

	Name	Type and Description
	<b>maxFuelLevel</b>	
		float32
		The rated fuel capacity of the device.
A2004		Annotations: <i>min=0,units=liter</i>
	<b>lowFuelLevelCutoff</b>	
		float32
		Level where engine automatically shuts down.
A2005		Annotations: <i>min=0,max=1,units=per unit</i>

A2006 B.5.2.15 tms.GeneratorInfo

A2007 PURPOSE: Design information for a power generator.

A2008 TOPIC USAGE: Nested

A2009 EXTENSIBILITY: extensibility(APPENDABLE)

A2010 PATTERN: Structure

A2011 ATTRIBUTES:

	Name	Type and Description
	<b>minFieldCurrent</b>	
		float32
		Minimum field current. Indicates field collapse?
		Annotations: <i>units=Ampere</i>
	<b>maxFieldCurrent</b>	
A2012		float32
		Maximum field current. Indicates saturation.
		Annotations: <i>units=Ampere</i>
	<b>maxStatorTemperature</b>	
		float32
		Maximum stator temperature. Indicates overheating.
A2013		Annotations: <i>min=-273,max=10000,units=Celsius</i>

A2014 B.5.2.16 tms.EnergyStorageInfo

A2015 PURPOSE: Design information for an energy storage device or unit.

A2016 TOPIC USAGE: Nested

A2017 EXTENSIBILITY: extensibility(APPENDABLE)

A2018 PATTERN: Structure

A2019 ATTRIBUTES:

Name	Type and Description
highStateOfCharge	float32 The high state of charge during continuous operations, only to be exceeded for limited-duration operation. Increased wear when charging above this point Annotations: <i>min</i> =0, <i>max</i> =1, <i>units</i> =per unit
lowStateOfCharge	float32 The low state of charge during continuous operations, only to go below for limited-duration operation. Increased wear when discharging below this point. Annotations: <i>min</i> =0, <i>max</i> =1, <i>units</i> =per unit
maxChargeEnergy	float32 Rated energy at full state of charge. Annotations: <i>units</i> =joule
minTemperature	float32 Minimum rated operating temperature. Annotations: <i>min</i> =-273, <i>max</i> =10000, <i>units</i> =Celsius
nomTemperature	float32 Most efficient operating temperature. Annotations: <i>min</i> =-273, <i>max</i> =10000, <i>units</i> =Celsius
maxTemperature	float32 Maximum rated operating temperature. Annotations: <i>min</i> =-273, <i>max</i> =10000, <i>units</i> =Celsius

A2021

A2022 B.5.2.17 tms.PowerElectronicsInfo

A2023 PURPOSE: Design information for power electronics.

A2024 TOPIC USAGE: Nested

A2025 EXTENSIBILITY: extensibility(APPENDABLE)

A2026 PATTERN: Structure

A2027 ATTRIBUTES:

Name	Type and Description
minTemperature	float32 Minimum rated operating temperature. Annotations: <i>min</i> =-273, <i>max</i> =10000, <i>units</i> =Celsius
maxTemperature	float32 Maximum rated operating temperature. Annotations: <i>min</i> =-273, <i>max</i> =10000, <i>units</i> =Celsius
minPower	float32 Minimum (greatest output) rated operating power. Annotations: <i>units</i> =watt
maxPower	float32 Minimum (maximum input) rated operating power. Annotations: <i>units</i> =watt
maxApparentPower	float32 Maximum (in/out) rated apparent power. Annotations: <i>units</i> =volt-ampere
shortCircuitTolerance	float32 Rated short circuit to not cause damage to the device. Annotations: <i>units</i> =Ampere

A2029

A2030 B.5.2.18 tms.TransformerInfo

A2031 PURPOSE: Design information for power transformers.

A2032 TOPIC USAGE: Nested

A2033 EXTENSIBILITY: extensibility(APPENDABLE)

A2034 PATTERN: Structure

A2035 ATTRIBUTES:

Name	Type and Description
------	----------------------

placeholder	
-------------	--

boolean	
---------	--

TODO feedback requested.	
--------------------------	--

A2038 B.5.2.19 tms.ThermalInfo

A2039 PURPOSE: Design information for thermal management.

A2040 TOPIC USAGE: Nested

A2041 EXTENSIBILITY: extensibility(APPENDABLE)

A2042 PATTERN: Structure

A2043 ATTRIBUTES:

Name	Type and Description
------	----------------------

thermalZone	
-------------	--

ThermalZoneSequence	
---------------------	--

Name of each thermal zone.	
----------------------------	--

A2046 B.5.2.20 tms.ThermalZoneSequence

A2047 PURPOSE: A sequence of String1\_32.

A2048 EXTENSIBILITY: extensibility(FINAL)

A2049 PATTERN: Typedef

A2050 ORIGINAL TYPE: `sequence<String1_32,0,5>`

A2051 B.5.2.21 tms.ParameterMetadataSequence

A2052 PURPOSE: A sequence of ParameterMetadata.

A2053 EXTENSIBILITY: extensibility(FINAL)

A2054 PATTERN: Typedef

A2055 ORIGINAL TYPE: `sequence<ParameterMetadata,0,128>`

A2056 B.5.2.22 tms.ParameterMetadata

A2057 PURPOSE: Describe an adjustable parameter.

A2058 TOPIC USAGE: Nested

A2059 EXTENSIBILITY: extensibility(APPENDABLE)

A2060 PATTERN: Structure

A2061 ATTRIBUTES:

Name	Type and Description
<code>name</code>	<code>String1_32</code> Name of this parameter.
<code>units</code>	<code>String16</code> Units of measure. (V, A, Hz, V/Hz, ...).
<code>nominalMinValue</code>	<code>float32</code> Recommended lower bound. Set to -infinity if no limit, otherwise $\text{hardMinValue} \leq \text{nominalMinValue} \leq \text{nominalMaxValue} \leq \text{hardMaxValue}$ .
<code>nominalMaxValue</code>	<code>float32</code> Recommended upper bound. Set to +infinity if no limit, otherwise $\text{hardMinValue} \leq \text{nominalMinValue} \leq \text{nominalMaxValue} \leq \text{hardMaxValue}$ .
<code>hardMinValue</code>	<code>float32</code> Absolute lower bound. Lower values will be rejected. Set to -infinity if no limit, otherwise $\text{hardMinValue} \leq \text{nominalMinValue} \leq \text{nominalMaxValue} \leq \text{hardMaxValue}$ .
<code>hardMaxValue</code>	<code>float32</code> Absolute upper bound. Higher values will be rejected. Set to +infinity if no limit, otherwise $\text{hardMinValue} \leq \text{nominalMinValue} \leq \text{nominalMaxValue} \leq \text{hardMaxValue}$ .
<code>resolution</code>	<code>float32</code> Smallest change $\geq 0$ that may have an effect. Set to 0 if no limit. Annotations: <i>min</i> =0
<code>enumLabels</code>	<code>EnumLabelsSequence</code> Enumeration labels. Each value must be at least 1 character in length. No duplicates are allowed per ParameterMetadata.

A2063 CONSTANTS:



	Name	Type [Units], Value, and Description
A2064	MAX_PARAMETERS	uint32, 128
A2065		Maximum number of parameters that a device may contain.
A2066	B.5.2.23 <u>tms.String16</u>	
A2067	PURPOSE:	A variable length string.
A2068	EXTENSIBILITY:	extensibility(FINAL)
A2069	PATTERN:	Typedef
A2070	ORIGINAL TYPE:	string<0,16>
A2071	B.5.2.24 <u>tms.EnumLabelsSequence</u>	
A2072	PURPOSE:	A sequence of String1_32.
A2073	EXTENSIBILITY:	extensibility(FINAL)
A2074	PATTERN:	Typedef
A2075	ORIGINAL TYPE:	sequence<String1_32,0,100>
A2076	B.5.2.25 <u>tms.ControlServiceInfo</u>	
A2077	PURPOSE:	Design information for control services on microgrid controllers and dashboards.
A2078	TOPIC USAGE:	Nested
A2079	EXTENSIBILITY:	extensibility(APPENDABLE)
A2080	PATTERN:	Structure
A2081	ATTRIBUTES:	
	Name	Type and Description
	mc	optional<MicrogridControllerInfo>
A2082		Microgrid Controller information.
	md	optional<MicrogridDashboardInfo>
A2083		Microgrid Dashboard information.

A2084 B.5.2.26 tms.MicrogridControllerInfo

A2085 PURPOSE: Design information for a microgrid controller.

A2086 TOPIC USAGE: Nested

A2087 EXTENSIBILITY: extensibility(APPENDABLE)

A2088 PATTERN: Structure

A2089 ATTRIBUTES:

Name	Type and Description
<b>features</b>	<b>MicrogridControllerFeatureSequence</b> A set of features this device supports.

A2092 B.5.2.27 tms.MicrogridControllerFeatureSequence

A2093 PURPOSE: A sequence of MicrogridControllerFeature.

A2094 EXTENSIBILITY: extensibility(FINAL)

A2095 PATTERN: Typedef

A2096 ORIGINAL TYPE: `sequence<MicrogridControllerFeature,1,64>`

A2097 B.5.2.28 tms.MicrogridControllerFeature

A2098 PURPOSE: Describes a feature on a specific microgrid controller.

A2099 EXTENSIBILITY: extensibility(APPENDABLE)

A2100 PATTERN: Enumeration

A2101 VALUES:

Name	Description
<b>MCF_UNKNOWN</b>	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
<b>MCF_FIXED</b>	special-purpose microgrid controller designed for a single platform or limited microgrid configuration.
<b>MCF_GENERAL</b>	general-purpose microgrid controller.

A2104 B.5.2.29 tms.MicrogridDashboardInfo

A2105 PURPOSE: Design information for a microgrid dashboard.

A2106 TOPIC USAGE: Nested

A2107 EXTENSIBILITY: extensibility(APPENDABLE)

A2108 PATTERN: Structure

A2109 ATTRIBUTES:

Name	Type and Description
<b>features</b>	<b>MicrogridDashboardFeatureSequence</b> A set of features this device supports.

A2112 B.5.2.30 tms.MicrogridDashboardFeatureSequence

A2113 PURPOSE: A sequence of MicrogridDashboardFeature.

A2114 EXTENSIBILITY: extensibility(FINAL)

A2115 PATTERN: Typedef

A2116 ORIGINAL TYPE: `sequence<MicrogridDashboardFeature,1,2>`

A2117 B.5.2.31 tms.MicrogridDashboardFeature

A2118 PURPOSE: Describes a feature on a specific microgrid dashboard.

A2119 EXTENSIBILITY: extensibility(APPENDABLE)

A2120 PATTERN: Enumeration

A2121 VALUES:

Name	Description
MDF_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
MDF_DISPLAY	Can show microgrid status.
MDF_CONTROL	Can control microgrid status.

A2124 B.5.2.32 tms.PowerDeviceInfo

A2125 PURPOSE: Design information for a power device which is any TMS device role that  
A2126 generates, distributes or consumes power. SRC, STOR, LOAD, DIST and CONV.

A2127 TOPIC USAGE: Nested

A2128 EXTENSIBILITY: extensibility(APPENDABLE)

A2129 PATTERN: Structure

A2130 ATTRIBUTES:

Name	Type and Description
<b>powerPorts</b>	<b>PowerPortInfoSequence</b> 1 for SRC and STOR and LOAD, 2+ for DIST and CONV
<b>grounds</b>	<b>GroundingInfoSequence</b> Grounding port information.
<b>conversion</b>	<b>optional&lt;ConversionInfo&gt;</b> Conversion information.
<b>distribution</b>	<b>optional&lt;DistributionInfo&gt;</b> Distribution device information.
<b>source</b>	<b>optional&lt;SourceInfo&gt;</b> Power source device information.
<b>storage</b>	<b>optional&lt;StorageInfo&gt;</b> Power storage device information.
<b>load</b>	<b>optional&lt;LoadInfo&gt;</b> Power load information.

A2133 B.5.2.33 tms.PowerPortInfoSequence

A2134 PURPOSE: A sequence of PowerPortInfo.

A2135 EXTENSIBILITY: extensibility(FINAL)

A2136 PATTERN: Typedef

A2137 ORIGINAL TYPE: `sequence<PowerPortInfo,0,MAX_PORTS>`

A2138 B.5.2.34 tms.PowerPortInfo

A2139 PURPOSE: Design information for the connector, rated circuit, and switching capabilities  
A2140 of a power port.

A2141 TOPIC USAGE: Nested

A2142 EXTENSIBILITY: extensibility(APPENDABLE)

A2143 PATTERN: Structure

A2144 ATTRIBUTES:

Name	Type and Description
portNumber	PowerPortNumber Number shown on the device exterior to represent this port.
directionality	PowerPortDirectionality Anticipated direction of power flow.
connectorType	PowerConnectorType Type of connector built in to the port.
polarity	PowerConnectorPolarity Polarity of the connector built in to the port.
phases	PowerConnectorPhases Type of circuit supported by this connector.
hasSwitch	boolean Indicate whether this power port has a switch.
hasExternalMeter	boolean Indicate whether this power port has an external meter.
hasInternalMeter	boolean Indicate whether this power port has an internal meter.
minAmperage	float32 minimum (greatest output) rated operating current. Annotations: <i>units</i> =Ampere
maxAmperage	float32 minimum (greatest output) rated operating current. Annotations: <i>units</i> =Ampere
interruptAmperage	optional<float32> maximum (in/out) interrupt current. Not present if no switch. Required if switch exists. Annotations: <i>units</i> =Ampere
minVoltage	float32 minimum rated operating voltage. Annotations: <i>units</i> =Volt
maxVoltage	float32 maximum rated operating voltage. Annotations: <i>units</i> =Volt
minFrequency	optional<float32> minimum rated operating frequency, not present for DC, required for AC

A2145

A2147 B.5.2.35 tms.PowerPortNumber

A2148 See definition in Section [B.2.2.4](#).

A2149 B.5.2.36 tms.PowerPortDirectionality

A2150 PURPOSE: Anticipated direction of power flow.

A2151 EXTENSIBILITY: extensibility(APPENDABLE)

A2152 PATTERN: Enumeration

A2153 VALUES:

Name	Description
PPD_UNKNOWN	not initialized.
PPD_NONE	directionality not applicable (e.g., for a logical port, expect zero power).
PPD_IN	Port intended to receive power.
PPD_OUT	Port intended to send power .
PPD_IN_OUT	Port intended to send and receive power.



A2156 B.5.2.37 tms.PowerConnectorType

A2157 PURPOSE: Indicate the standard defining the physical shape and form of a power connector.

A2158 EXTENSIBILITY: extensibility(APPENDABLE)

A2159 PATTERN: Enumeration

A2160 VALUES:

Name	Description
CONNECTOR_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
CONNECTOR_TERMINAL_BLOCK	Terminal block.
CONNECTOR_MILSTD	Generic MIL spec connector. MIL-STD-1651, MIL-DTL-22992, and MIL-DTL-53126, as used by PDISE.
CONNECTOR_NEMA5	For convenience power ports, etc.
CONNECTOR_CAMLOCK	As used by MEPDIS-R.
CONNECTOR_POWERLOCK	ITT VEAM PowerLock
CONNECTOR_IEC60309	As used by MEPDIS-R.
CONNECTOR_J1772	Electric vehicle charging cable.
CONNECTOR_POWERLOK	Amphenol PowerLok.
CONNECTOR_MILSTD1651	MIL-STD-1651.
CONNECTOR_MILDTL22992	MIL-DTL-22992.
CONNECTOR_MILDTL53126	MIL-DTL-53126.
CONNECTOR_OTHER	Other connector type, not listed elsewhere.
CONNECTOR_LOGICAL	Not a physical connector. Used to represent bus measurements or synchronizing voltage.

A2162

A2163 B.5.2.38 tms.PowerConnectorPolarity

A2164 PURPOSE: Indicate the polarity of a power connector.

A2165 EXTENSIBILITY: extensibility(APPENDABLE)

A2166 PATTERN: Enumeration

A2167 VALUES:

Name	Description
------	-------------

---

POLARITY_UNKNOWN	
------------------	--

	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
--	---

POLARITY_PIN	
--------------	--

A2168 

	This connector contains pins to insert in sockets.
--	--

POLARITY_SOCKET	
-----------------	--

	This connector contains sockets to receive pins. Includes terminal blocks.
--	--

POLARITY_UNIVERSAL	
--------------------	--

	This connector is symmetric.
--	------------------------------

A2169

---

A2170 B.5.2.39 tms.PowerConnectorPhases

A2171 PURPOSE: Indicate how many phases a power connector supports.

A2172 EXTENSIBILITY: extensibility(APPENDABLE)

A2173 PATTERN: Enumeration

A2174 VALUES:

Name	Description
------	-------------

---

PHASE_UNKNOWN	
---------------	--

	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
--	---

PHASE_SINGLE	
--------------	--

	Single-phase circuit.
--	-----------------------

PHASE_SPLIT	
-------------	--

A2175 

	Split-phase circuit.
--	----------------------

PHASE_3WYE	
------------	--

	3-phase circuit, wye configuration.
--	-------------------------------------

PHASE_3DELTA	
--------------	--

	3-phase circuit, delta configuration.
--	---------------------------------------

PHASE_DC	
----------	--

	DC circuit.
--	-------------

A2176

---

A2177 B.5.2.40 tms.SwitchFeatureSequence

A2178 PURPOSE: A sequence of SwitchFeature.

A2179 EXTENSIBILITY: extensibility(FINAL)

A2180 PATTERN: Typedef

A2181 ORIGINAL TYPE: `sequence<SwitchFeature,0,9>`

A2182 B.5.2.41 tms.SwitchFeature

A2183 PURPOSE: Describes a feature on a switch.

A2184 EXTENSIBILITY: extensibility(APPENDABLE)

A2185 PATTERN: Enumeration

A2186 VALUES:

Name	Description
<hr/>	
SWITCH_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
SWITCH_MANUAL	Manual operation.
SWITCH_AUTO	Automatic (internal) logic such as load shed.
SWITCH_REMOTE	Remote operation (PowerSwitchRequest).
SWITCH_RECLOSER	Able to close without manual operation (automatic or remote).
SWITCH_BREAKER	Over current / thermal protection.
SWITCH_GFI	Ground fault protection.
SWITCH_ARC_FLASH	Arc flash protection.
SWITCH_SYNCHRONIZER	Synchronize before close.
SWITCH_SURGE	Over voltage protection.

A2188

A2189 B.5.2.42 tms.ConnectorFeatureSequence

A2190 PURPOSE: A sequence of ConnectorFeature.

A2191 EXTENSIBILITY: extensibility(FINAL)

A2192 PATTERN: Typedef

A2193 ORIGINAL TYPE: `sequence<ConnectorFeature,0,5>`

A2194 B.5.2.43 tms.ConnectorFeature

A2195 PURPOSE: Indicate how a cable can be detected.

A2196 EXTENSIBILITY: extensibility(APPENDABLE)

A2197 PATTERN: Enumeration

A2198 VALUES:

Name	Description
------	-------------

---

CF_UNKNOWN	
------------	--

	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
--	---

CF_CABLE_SENSE	
----------------	--

	A continuity circuit or similar feature detects when a cable is connected.
--	--

CF_CABLE_ID_READER	
--------------------	--

A2199                      A cable ID reader is present.

CF_CABLE_MEASUREMENT	
----------------------	--

	Voltage and current monitoring can establish the presence of a cable.
--	---

CF_COMMUNICATION	
------------------	--

	Contains network interface.
--	-----------------------------

CF_TOPOLOGY_DISCOVERY	
-----------------------	--

A2200                      Ability to discover connectivity to remote device power ports.

---

A2201 B.5.2.44 tms.GroundingInfoSequence

A2202 PURPOSE: A sequence of GroundingInfo.

A2203 EXTENSIBILITY: extensibility(FINAL)

A2204 PATTERN: Typedef

A2205 ORIGINAL TYPE: `sequence<GroundingInfo,0,MAX_PORTS>`

A2206 B.5.2.45 tms.GroundingInfo

A2207 PURPOSE: Design information describing a grounding circuit in a device.

A2208 TOPIC USAGE: Nested

A2209 EXTENSIBILITY: extensibility(APPENDABLE)

A2210 PATTERN: Structure

A2211 ATTRIBUTES:

Name	Type and Description
groundNumber	GroundingCircuitNumber Number used to represent this circuit.
groundType	GroundingDesignType Type of grounding provided by this circuit.
protectedPorts	PowerPortNumberSequence List of ports protected by this circuit.
controlSwitchFeatures	SwitchFeatureSequence Not present if no control switch. Required if control switch exists. Length 0 indicates no features.
pulseSwitchFeatures	SwitchFeatureSequence Not present if no pulse switch. Required if pulse switch exists. Length 0 indicates no features.

A2213

A2214 B.5.2.46 tms.GroundingCircuitNumber

A2215 PURPOSE: Identifies a grounding circuit within a device.

A2216 EXTENSIBILITY: extensibility(APPENDABLE)

A2217 PATTERN: Typedef

A2218 ORIGINAL TYPE: uint32

A2219 CONSTANTS:

Name	Type [Units], Value, and Description
MAX_GROUNDS	uint32, 20 Maximum number of grounding circuits that a device may contain.

A2220

A2221

A2222 B.5.2.47 tms.GroundingDesignType

A2223 PURPOSE: Describe the design type of a grounding circuit.

A2224 EXTENSIBILITY: extensibility(APPENDABLE)

A2225 PATTERN: Enumeration

A2226 VALUES:

Name	Description
GROUNDING_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
GROUNDING_UNGROUNDED	No grounding.
GROUNDING_SOLID	Solidly grounded.
A2227 GROUNDING_HIGH_RESISTANCE	High-resistance grounded system. Generally designed for 1-10 A ground fault current.
GROUNDING_LOW_RESISTANCE	Low-resistance grounded system. Generally designed to limit equipment damage.
GROUNDING_REACTANCE	Reactance grounded system. Generally limits ground fault current to rated line current.

A2228

A2229 B.5.2.48 tms.PowerPortNumberSequence

A2230 PURPOSE: A sequence of PowerPortNumber.

A2231 EXTENSIBILITY: extensibility(FINAL)

A2232 PATTERN: Typedef

A2233 ORIGINAL TYPE: `sequence<PowerPortNumber,0,MAX_PORTS>`

A2234 B.5.2.49 tms.ConversionInfo

A2235 PURPOSE: Design information for power conversion devices.

A2236 TOPIC USAGE: Nested

A2237 EXTENSIBILITY: extensibility(APPENDABLE)

A2238 PATTERN: Structure

A2239 ATTRIBUTES:

Name	Type and Description
features	
ConversionFeatureSequence	
A set of features this device supports.	
activeConversion	
optional<ActiveConversionInfo>	
Power regulation for active devices.	
passiveConversion	
optional<PassiveConversionInfo>	
Power regulation for passive devices.	

A2242 B.5.2.50 tms.ConversionFeatureSequence

A2243 PURPOSE: A sequence of ConversionFeature.

A2244 EXTENSIBILITY: extensibility(FINAL)

A2245 PATTERN: Typedef

A2246 ORIGINAL TYPE: sequence<ConversionFeature,0,2>

A2247 B.5.2.51 tms.ConversionFeature

A2248 PURPOSE: Describes a feature on a specific energy converter.

A2249 EXTENSIBILITY: extensibility(APPENDABLE)

A2250 PATTERN: Enumeration

A2251 VALUES:

Name	Description
CONVF_UNKNOWN	
Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.	
CONVF_ACTIVE	
Uses power electronics or other active components (programmable).	
CONVF_PASSIVE	
Uses transformers or other passive components (fixed function).	

A2254 B.5.2.52 tms.ActiveConversionInfo

A2255 PURPOSE: Design information for active conversion load sharing devices.

A2256 TOPIC USAGE: Nested

A2257 EXTENSIBILITY: extensibility(APPENDABLE)

A2258 PATTERN: Structure

A2259 ATTRIBUTES:

Name	Type and Description
loadSharing	LoadSharingInfoSequence
	Load sharing capabilities of each power port.

A2262 B.5.2.53 tms.LoadSharingInfoSequence

A2263 PURPOSE: A sequence of LoadSharingInfo.

A2264 EXTENSIBILITY: extensibility(FINAL)

A2265 PATTERN: Typedef

A2266 ORIGINAL TYPE: `sequence<LoadSharingInfo,1,MAX_PORTS>`



A2267 B.5.2.54 tms.LoadSharingInfo

A2268 PURPOSE: Design information for a device capable of load sharing.

A2269 TOPIC USAGE: Nested

A2270 EXTENSIBILITY: extensibility(APPENDABLE)

A2271 PATTERN: Structure

A2272 ATTRIBUTES:

*TMS Data Model*  
*DRAFT 19 Apr 2021*

Name	Type and Description
portNumber	optional<PowerPortNumber> The power port number.
supportsDroopCurve	boolean Indicates that the device can allow the frequency (AC) and voltage (AC and DC) to sag as load increases.
supportsNonlinearDroopCurve	boolean Supports droop curves having multiple piecewise linear segments.
inertia	float32 TBD: units and number of measurements
minP	float32 Minimum (greatest output) rated real power (without overload) Annotations: <i>units</i> =watt
maxP	float32 Maximum (greatest input) rated real power (without overload) Annotations: <i>units</i> =watt
minOverloadP	float32 Greatest overload output real power (short duration or under ideal conditions) Annotations: <i>units</i> =watt
minQ	float32 Minimum Q value. Annotations: <i>units</i> =volt ampere reactive
maxQ	float32 Maximum Q value. Annotations: <i>units</i> =volt ampere reactive
maxVA	float32 Rated apparent power
powerFactor	float32 Rated power factor, usually 0.8

A2273

A2274

A2275 B.5.2.55 tms.PassiveConversionInfo

A2276 PURPOSE: Design information for a passive energy converter.

A2277 TOPIC USAGE: Nested

A2278 EXTENSIBILITY: extensibility(APPENDABLE)

A2279 PATTERN: Structure

A2280 ATTRIBUTES:

Name	Type and Description
<b>portConversion</b>	<b>PowerPortConversionInfoSequence</b> Information for each port on the device.

A2283 B.5.2.56 tms.PowerPortConversionInfoSequence

A2284 PURPOSE: A sequence of PowerPortConversionInfo.

A2285 EXTENSIBILITY: extensibility(FINAL)

A2286 PATTERN: Typedef

A2287 ORIGINAL TYPE: `sequence<PowerPortConversionInfo,1,MAX_PORTS>`

A2288 B.5.2.57 tms.PowerPortConversionInfo

A2289 PURPOSE: Design information for power port conversation devices.

A2290 TOPIC USAGE: Nested

A2291 EXTENSIBILITY: extensibility(APPENDABLE)

A2292 PATTERN: Structure

A2293 ATTRIBUTES:

Name	Type and Description
<b>portNumber</b>	<b>PowerPortNumber</b> The power port number this structure represents.
<b>conversionTaps</b>	<b>ConversionTapInfoSequence</b> Sequence to support tap changers
<b>changeUnderLoad</b>	<b>boolean</b> False requires (or causes) circuit interrupt to change, true can change under rated load.

A2296 B.5.2.58 tms.ConversionTapInfoSequence

A2297 PURPOSE: A sequence of ConversionTapInfo.

A2298 EXTENSIBILITY: extensibility(FINAL)

A2299 PATTERN: Typedef

A2300 ORIGINAL TYPE: `sequence<ConversionTapInfo,1,MAX_TAPS>`

A2301 B.5.2.59 tms.ConversionTapInfo

A2302 PURPOSE: Design information for conversion taps.

A2303 TOPIC USAGE: Nested

A2304 EXTENSIBILITY: extensibility(APPENDABLE)

A2305 PATTERN: Structure

A2306 ATTRIBUTES:

Name	Type and Description
<code>tapId</code>	
	<code>TapNumber</code>
	Tap changer position.
<code>phaseShift</code>	
	<code>float32</code>
	Phase shift between this port and the internal bus.
<code>voltageRatio</code>	
	<code>float32</code>
	Voltage ratio between this port and the internal bus.

A2309 B.5.2.60 tms.TapNumber

A2310 PURPOSE: Tap changer index in a passive conversion device. Defaults to 0 when no changer  
A2311 is present.

A2312 EXTENSIBILITY: extensibility(APPENDABLE)

A2313 PATTERN: Typedef

A2314 ORIGINAL TYPE: `int32`

A2315 CONSTANTS:

Name	Type [Units], Value, and Description
<code>MAX_TAPS</code>	
	<code>int32, 64</code>
	Maximum number of positions in a tap changer. Many tap changers have 33
	positions.

A2318 B.5.2.61 tms.DistributionInfo

A2319 PURPOSE: Design information for the distribution device role.

A2320 TOPIC USAGE: Nested

A2321 EXTENSIBILITY: extensibility(APPENDABLE)

A2322 PATTERN: Structure

A2323 ATTRIBUTES:

Name	Type and Description
features	DistributionFeatureSequence A set of features this device supports.

A2326 B.5.2.62 tms.DistributionFeatureSequence

A2327 PURPOSE: A sequence of DistributionFeature.

A2328 EXTENSIBILITY: extensibility(FINAL)

A2329 PATTERN: Typedef

A2330 ORIGINAL TYPE: sequence<DistributionFeature,0,5>

A2331 B.5.2.63 tms.DistributionFeature

A2332 PURPOSE: Describes a feature on a specific energy distribution device.

A2333 EXTENSIBILITY: extensibility(APPENDABLE)

A2334 PATTERN: Enumeration

A2335 VALUES:

Name	Description
DISTF_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
DISTF_CLAMP_METER	clamp-on power meter, two logical power ports.
DISTF_TAP_METER	pass-through power meter, two physical power ports.
DISTF_PCC	has additional capabilities for PCC interconnect with another grid.
DISTF_FEEDER	intended for supplying power to other DIST devices.
DISTF_DISTRIBUTION	intended for supplying power to LOAD devices.

A2338 B.5.2.64 tms.SourceInfo

A2339 PURPOSE: Design information for power sources.

A2340 TOPIC USAGE: Nested

A2341 EXTENSIBILITY: extensibility(APPENDABLE)

A2342 PATTERN: Structure

A2343 ATTRIBUTES:

Name	Type and Description
<b>features</b>	<b>SourceFeatureSequence</b>
	A set of features this device supports.
<b>loadSharing</b>	<b>LoadSharingInfo</b>
	Load sharing capabilities of this device.

A2346 B.5.2.65 tms.SourceFeatureSequence

A2347 PURPOSE: A sequence of SourceFeature.

A2348 EXTENSIBILITY: extensibility(FINAL)

A2349 PATTERN: Typedef

A2350 ORIGINAL TYPE: **sequence<SourceFeature,0,5>**

A2351 B.5.2.66 tms.SourceFeature

A2352 PURPOSE: Describes a feature on a specific energy source device.

A2353 EXTENSIBILITY: extensibility(APPENDABLE)

A2354 PATTERN: Enumeration

A2355 VALUES:

Name	Description
SRCF_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
SRCF_GENSET	Genset source.
SRCF_FUEL_CELL	Fuel cell source.
SRCF_SOLAR	Solar source.
SRCF_WIND	Wind source.
SRCF_VEHICLE	Source shared with a vehicle power train.

A2358 B.5.2.67 tms.StorageInfo

A2359 PURPOSE: Design information for power storage devices.

A2360 TOPIC USAGE: Nested

A2361 EXTENSIBILITY: extensibility(APPENDABLE)

A2362 PATTERN: Structure

A2363 ATTRIBUTES:

Name	Type and Description
features	StorageFeatureSequence
	A set of features this device supports.
loadSharing	LoadSharingInfo
	Load sharing capabilities of this device.

A2366 B.5.2.68 tms.StorageFeatureSequence

A2367 PURPOSE: A sequence of StorageFeature.

A2368 EXTENSIBILITY: extensibility(FINAL)

A2369 PATTERN: Typedef

A2370 ORIGINAL TYPE: `sequence<StorageFeature,0,4>`

A2371 B.5.2.69 tms.StorageFeature

A2372 PURPOSE: Describes a feature on a specific energy distribution device.

A2373 EXTENSIBILITY: extensibility(APPENDABLE)

A2374 PATTERN: Enumeration

A2375 VALUES:

Name	Description
------	-------------

---

STORF_UNKNOWN	
---------------	--

	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
--	---

STORF_GRID	
------------	--

	grid attached storage, does not require pass-through from source to load.
--	---

STORF_SUBCYCLE_UPS	
--------------------	--

	can provide fast voltage support to recover from a grid fault condition.
--	--

STORF_CHARGING	
----------------	--

	charging station.
--	-------------------

STORF_VEHICLE	
---------------	--

	storage shared with a vehicle power train.
--	--

---



A2378 B.5.2.70 tms.LoadInfo

A2379 PURPOSE: Design information for load devices.

A2380 TOPIC USAGE: Nested

A2381 EXTENSIBILITY: extensibility(APPENDABLE)

A2382 PATTERN: Structure

A2383 ATTRIBUTES:

Name	Type and Description
<b>features</b>	<b>LoadFeatureSequence</b> A set of features this device supports.
<b>loadChangeThreshold</b>	<b>float32</b> Minimum step size for a LoadChangeRequest. Set to NaN if not supported. Annotations: <i>units</i> =watt
<b>loadEstimateDeadband</b>	<b>float32</b> Minimum step size for a LoadEstimate update. Annotations: <i>units</i> =watt

A2386 B.5.2.71 tms.LoadFeatureSequence

A2387 PURPOSE: A sequence of LoadFeature.

A2388 EXTENSIBILITY: extensibility(FINAL)

A2389 PATTERN: Typedef

A2390 ORIGINAL TYPE: `sequence<LoadFeature,0,3>`

A2391 B.5.2.72 tms.LoadFeature

A2392 PURPOSE: Describes a feature on a specific energy load.

A2393 EXTENSIBILITY: extensibility(APPENDABLE)

A2394 PATTERN: Enumeration

A2395 VALUES:

Name	Description
LOADF_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
LOADF_DEMAND_RESPONSE	Device can shed load to provide grid support.
LOADF_CHANGE_NOTIFICATION	Device can notify the microgrid before causing a large load transient.
LOADF_SOFT_START	Device can smoothly ramp up and down to minimize load transients.

A2398 B.6 Nickname

A2399 Every TMS device is identified by a machine-readable **Fingerprint**. This **Fingerprint**  
A2400 may appear as an extremely long number with no clear meaning to the operator. The  
A2401 **FingerprintNickname** is an alternate identity, containing human-readable text intended for  
A2402 display on the device and MD. The **FingerprintNickname** may be configured on the device  
A2403 or MD.

A2404 B.6.1 Data Flows

A2405 Figure B.9 illustrates all the topics in the data flow.

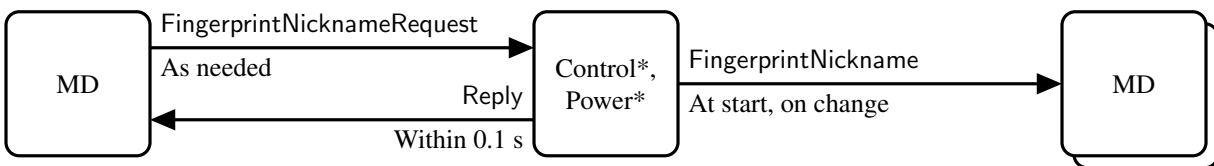


Figure B.9: Nickname data flow

A2406 The **FingerprintNickname** is only for operator convenience. It does not impact the  
A2407 operation of the microgrid because all TMS device communications use the full cryptographic  
A2408 identity. The **FingerprintNickname** shall not be used for automatic control.

A2409 There are three ways that a nickname can be assigned to the device:

- A2410 1. Stored in memory so it can be announced at startup.

- A2411 2. Directly assigned by an operator. The device will send a message after the nickname  
A2412 is assigned.
- A2413 3. Issued a new name by the MD. If the MD request is successful, the device nickname is  
A2414 updated and the message set after the update.
- A2415 Table B.15 provides an overview of how each topic is used.

Table B.15: Description of the Nickname topics.

Topic	Description
FingerprintNickname	Device and Platform Fingerprint Nickname.
FingerprintNicknameRequest	Request to change a Device or Platform Fingerprint Nickname.
Reply	The reply to a request.

- A2416 Table B.16 specifies the publishers and subscribers for each topic as well as usage  
A2417 requirements as described in Section A.8.7.

Table B.16: Participants of the Nickname topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
FingerprintNickname	OPTIONAL	Pub	Pub		Pub	Pub	Pub	Pub	Pub
		Sub		Sub					
FingerprintNicknameRequest	OPTIONAL	Pub							
		Sub	Sub	Sub	Sub	Sub	Sub	Sub	Sub
Reply	REQUIRED	Pub	Pub		Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

- A2418 Table B.17 specifies the timing for when sample updates are sent to each topic. These  
A2419 parameters are defined in Section A.8.5.

Table B.17: Timing of the Nickname topics.

Topic	Data Trigger	Rate Class	Burst Size	QoS Profile
FingerprintNickname	At Start, On Change	10 s	1	PublishLast
FingerprintNicknameRequest	As Needed	10 s	1	Command
Reply	Within 0.1 s	All	1	Response

## A2420 B.6.2 Data Types

- A2421 Table B.18 specifies the data types that are used as sample values for each topic.

Table B.18: Data types of the Nickname topics.

Topic	Data Type
FingerprintNickname	<code>tms.FingerprintNickname</code>
FingerprintNicknameRequest	<code>tms.ChangeNicknameRequest</code>
Reply	<code>tms.Reply</code>

A2422 Topic-level data types are defined in the order shown in Table [B.18](#). Then nested data  
A2423 types are defined in the order of appearance. Cross-references are provided for data types  
A2424 that have already been defined.

#### A2425 B.6.2.1 `tms.FingerprintNickname`

A2426 PURPOSE: Provide a short, human-readable name for a Fingerprint.

A2427 TOPIC USAGE: FingerprintNickname

A2428 EXTENSIBILITY: extensibility(APPENDABLE)

A2429 PATTERN: Structure

A2430 ATTRIBUTES:

Name	Type and Description
<code>id</code>	<b>Fingerprint</b> The Fingerprint described by this structure. Annotations: <i>keyval</i>
<code>nickname</code>	<b>String32</b> Nickname given to this Fingerprint.

A2433 B.6.2.2 tms.ChangeNicknameRequest

A2434 PURPOSE: Change the nickname associated with a Fingerprint.

A2435 TOPIC USAGE: FingerprintNicknameRequest

A2436 EXTENSIBILITY: extensibility(APPENDABLE)

A2437 PATTERN: Structure

A2438 ATTRIBUTES:

Name	Type and Description
<code>requestId</code>	<code>DeviceRequest</code> Identity of this request. Annotations: <i>keyval</i>
<code>sequenceId</code>	<code>tms::RequestSequence</code> Request sequence data used to associate a request and returning reply.
<code>id</code>	<code>Fingerprint</code> Fingerprint that should store a new nickname.
<code>nickname</code>	<code>String32</code> New nickname to be used for this Fingerprint.

A2441 B.6.2.3 tms.Reply

A2442 See definition in Section [B.2.2.1](#).

A2443 B.6.2.4 tms.Fingerprint

A2444 See definition in Section [B.2.2.2](#).

A2445 B.6.2.5 tms.String32

A2446 See definition in Section [B.2.2.7](#).

A2447 B.6.2.6 tms.DeviceRequest

A2448 PURPOSE: Key value for a request that targets one device.

A2449 TOPIC USAGE: Nested

A2450 EXTENSIBILITY: extensibility(APPENDABLE)

A2451 PATTERN: Structure

A2452 ATTRIBUTES:

Name	Type and Description
<u>requestingDeviceId</u>	Fingerprint
	Identity of the device that sent this request.
<u>targetDeviceId</u>	Fingerprint
	Identity of the device that should receive this request.

A2455 B.6.2.7 tms.RequestSequence

A2456 See definition in Section [B.2.2.5](#).

A2457 B.6.2.8 tms.ConfigId

A2458 See definition in Section [B.2.2.3](#).

A2459 B.6.2.9 tms.PowerPortNumber

A2460 See definition in Section [B.2.2.4](#).

A2461 B.6.2.10 tms.ReplyStatus

A2462 See definition in Section [B.2.2.6](#).

A2463 B.7 Device Icon

A2464 The Device Icon identifies an image that can be embedded into a device and shown on an  
A2465 operator GUI. The image is tied to a device type, not a specific device or its operational  
A2466 state.

A2467 B.7.1 Data Flows

A2468 Figure [B.10](#) illustrates all the topics in the data flow.

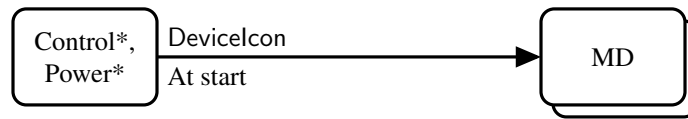


Figure B.10: Device Icon data flow

A2469 The GUI may receive an image from one device of a given type and use it for all devices  
A2470 of the same type.

A2471

A2472 Table B.19 provides an overview of how each topic is used.

Table B.19: Description of the Device Icon topics.

Topic	Description
DeviceIcon	Device image as an icon.

A2473 Table B.20 specifies the publishers and subscribers for each topic as well as usage  
A2474 requirements as described in Section A.8.7.

Table B.20: Participants of the Device Icon topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DeviceIcon	OPTIONAL	Pub Sub	Pub	Sub	Pub	Pub	Pub	Pub	Pub

A2475 Table B.21 specifies the timing for when sample updates are sent to each topic. These  
A2476 parameters are defined in Section A.8.5.

Table B.21: Timing of the Device Icon topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
DeviceIcon	At Start	Once	1			PublishLast

## A2477 B.7.2 Data Types

A2478 Table B.22 specifies the data types that are used as sample values for each topic.

Table B.22: Data types of the Device Icon topics.

Topic	Data Type
DeviceIcon	tms.DeviceIcon

A2479 Topic-level data types are defined in the order shown in Table B.22. Then nested data  
A2480 types are defined in the order of appearance. Cross-references are provided for data types  
A2481 that have already been defined.

A2482 B.7.2.1 tms.DeviceIcon

A2483 PURPOSE: Small image representing a device.

A2484 TOPIC USAGE: DeviceIcon

A2485 EXTENSIBILITY: extensibility(APPENDABLE)

A2486 PATTERN: Structure

A2487 ATTRIBUTES:

Name	Type and Description
<b>deviceId</b>	<b>Fingerprint</b> The device illustrated by this image. Annotations: <i>keyval</i>
<b>contentType</b>	<b>String32</b> Type of the image format. Currently, the only supported type is 'image/png'. See the IANA Media Type list - <a href="http://www.iana.org/assignments/media-types/media-types.xhtml">http://www.iana.org/assignments/media-types/media-types.xhtml</a> .
<b>data</b>	<b>OctetSequence</b> Contents of the image with a resolution of 64x64 and transparent background. The image should be clearly visible on a white or black background.

A2490 B.7.2.2 tms.Fingerprint

A2491 See definition in Section [B.2.2.2](#).

A2492 B.7.2.3 tms.String32

A2493 See definition in Section [B.2.2.7](#).



A2494 B.7.2.4 tms.OctetSequence

A2495 PURPOSE: A sequence of Octet (byte).

A2496 EXTENSIBILITY: extensibility(FINAL)

A2497 PATTERN: Typedef

A2498 ORIGINAL TYPE: `sequence<octet,1,32768>`

A2499 B.8 Control Hardware Status

A2500 B.8.1 Data Flows

A2501 Figure B.11 illustrates all the topics in the data flow.

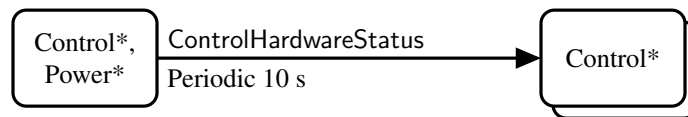


Figure B.11: Control Hardware Status data flow

A2502 Table B.23 provides an overview of how each topic is used.

Table B.23: Description of the Control Hardware Status topics.

Topic	Description
ControlHardwareStatus	Reports dynamic state changes of control device hardware.

A2503 Table B.24 specifies the publishers and subscribers for each topic as well as usage  
A2504 requirements as described in Section A.8.7.

Table B.24: Participants of the Control Hardware Status topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
ControlHardwareStatus	OPTIONAL	Pub Sub	Pub Sub	Pub Sub	Pub	Pub	Pub	Pub	Pub

A2505 Table B.25 specifies the timing for when sample updates are sent to each topic. These  
A2506 parameters are defined in Section A.8.5.

Table B.25: Timing of the Control Hardware Status topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
ControlHardwareStatus	Periodic 10 s	10 s	1			Slow

A2507 B.8.2 Data Types

A2508 Table [B.26](#) specifies the data types that are used as sample values for each topic.

Table B.26: Data types of the Control Hardware Status topics.

Topic	Data Type
ControlHardwareStatus	<code>tms.ControlHardwareState</code>

A2509 Topic-level data types are defined in the order shown in Table [B.26](#). Then nested data  
A2510 types are defined in the order of appearance. Cross-references are provided for data types  
A2511 that have already been defined.

A2512 B.8.2.1 tms.ControlHardwareState

A2513 PURPOSE: Operating state for the control hardware in this device.

A2514 TOPIC USAGE: ControlHardwareStatus

A2515 EXTENSIBILITY: extensibility(APPENDABLE)

A2516 PATTERN: Structure

A2517 ATTRIBUTES:

Name	Type and Description
deviceId	
	Fingerprint
	The device described by this structure.
	Annotations: <i>keyval</i>
timestamp	
	ClockMonotonic
	Time of these measurements.
processorUsage	
	float32
	Central Processing Unit (CPU) usage, as a per unit measure of the whole capacity.
	Annotations: <i>min=0,max=1,units=per unit</i>
memoryUsage	
	float32
	Memory utilization of the hardware, as a per unit measure of the whole capacity.
	Annotations: <i>min=0,max=1,units=per unit</i>
networkUsage	
	float32
	Network utilization of the hardware, as a per unit measure of the whole capacity.
	If the device has multiple network interfaces, this field will describe the network interface utilized by TMS communication protocols.
	Annotations: <i>min=0,max=1,units=per unit</i>
storageUsage	
	float32
	Disk storage utilization of the hardware, as a per unit measure of the whole capacity.
	Annotations: <i>min=0,max=1,units=per unit</i>
temperature	
	float32
	Temperature measurement of the control hardware.
	Annotations: <i>min=-273,max=10000,units=Celsius</i>

A2520 B.8.2.2 tms.Fingerprint

A2521 See definition in Section [B.2.2.2](#).

A2522 B.8.2.3 tms.ClockMonotonic

A2523 PURPOSE: Local oscillator timestamp, not affected by clock jumps.

A2524 TOPIC USAGE: Nested

A2525 EXTENSIBILITY: extensibility(APPENDABLE)

A2526 PATTERN: Structure

A2527 ATTRIBUTES:

	Name	Type and Description
	<b>seconds</b>	
	uint32	
A2528		integer portion of the timestamp.
	<b>nanoseconds</b>	
	uint32	
A2529		fractional portion of the timestamp.

A2530 DESCRIPTION:

A2531 This clock is used to measure the relative time between events local to a device. The start  
A2532 time,  $t=0.0$ , of this clock is unspecified. This clock must be available for all messages,  
A2533 forward progress must track a faithful estimate of passing time, and negative jumps are  
A2534 prohibited.

A2535

A2536 For more information on the intended usage of this clock, see IEEE 1003.1-2012, Section  
A2537 3.227 Monotonic Clock. See also IEEE 1003.13-2003, Section 6.6.1.13 Clocks and Timers.

A2538 B.9 Power Hardware Status

A2539 B.9.1 Data Flows

A2540 Figure B.12 illustrates all the topics in the data flow.

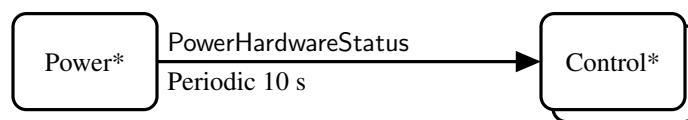


Figure B.12: Power Hardware Status data flow

A2541 Table B.27 provides an overview of how each topic is used.

Table B.27: Description of the Power Hardware Status topics.

Topic	Description
PowerHardwareStatus	Reports dynamic state of power hardware components.

A2542 Table B.28 specifies the publishers and subscribers for each topic as well as usage  
A2543 requirements as described in Section A.8.7.

Table B.28: Participants of the Power Hardware Status topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
PowerHardwareStatus	OPTIONAL	Pub	Pub	Pub	Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A2544 Table B.29 specifies the timing for when sample updates are sent to each topic. These  
A2545 parameters are defined in Section A.8.5.

Table B.29: Timing of the Power Hardware Status topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
PowerHardwareStatus	Periodic 10 s	10 s	1			Slow

## A2546 B.9.2 Data Types

A2547 Table B.30 specifies the data types that are used as sample values for each topic.

Table B.30: Data types of the Power Hardware Status topics.

Topic	Data Type
PowerHardwareStatus	<code>tms.PowerHardwareState</code>

A2548 Topic-level data types are defined in the order shown in Table B.30. Then nested data  
A2549 types are defined in the order of appearance. Cross-references are provided for data types  
A2550 that have already been defined.

A2551 B.9.2.1 tms.PowerHardwareState

A2552 PURPOSE: Operating state of the power hardware.

A2553 TOPIC USAGE: PowerHardwareStatus

A2554 EXTENSIBILITY: extensibility(APPENDABLE)

A2555 PATTERN: Structure

A2556 ATTRIBUTES:

Name	Type and Description
deviceId	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
timestamp	<b>ClockMonotonic</b> Time of these measurements.
engine	<b>optional&lt;EngineState&gt;</b> Operating state of the engine if present.
fuel	<b>optional&lt;FuelState&gt;</b> Fuel state, if present.
generator	<b>optional&lt;GeneratorState&gt;</b> Operating state of a generator, if present.
energyStorage	<b>optional&lt;EnergyStorageState&gt;</b> Operating state of energy storage if present.
powerElectronics	<b>optional&lt;PowerElectronicsState&gt;</b> Operating state of active components including inverters and rectifiers, if present.
transformer	<b>optional&lt;TransformerState&gt;</b> Operating state of passive components, if present.
thermal	<b>optional&lt;ThermalState&gt;</b> Thermal operating state including heating and cooling components, if present.

A2559 B.9.2.2 tms.Fingerprint

A2560 See definition in Section [B.2.2.2](#).

A2561 B.9.2.3 tms.ClockMonotonic

A2562 See definition in Section [B.8.2.3](#).

A2563 B.9.2.4 tms.EngineState

A2564 PURPOSE: Report key parameters about the state of the engine.

A2565 TOPIC USAGE: Nested

A2566 EXTENSIBILITY: extensibility(APPENDABLE)

A2567 PATTERN: Structure

A2568 ATTRIBUTES:

Name	Type and Description
oilPressure	float32 Oil pressure. Annotations: <i>units</i> =pascal
coolantTemperature	float32 Coolant temperature. Annotations: <i>min</i> =-273.15, <i>max</i> =10000, <i>units</i> =Celsius
speed	float32 Speed. Annotations: <i>min</i> =0, <i>units</i> =radian per second
wetStack	optional<float32> wet stack per unit, mitigation required or indicated at 1. Annotations: <i>min</i> =0, <i>max</i> =1, <i>units</i> =per unit
engineRuntime	float32 Engine runtime hours. Annotations: <i>min</i> =0, <i>units</i> =hour
fuelConsumptionRate	optional<float32> Fuel consumption rate at the present power level. Annotations: <i>min</i> =0, <i>units</i> =liters per second

A2571 B.9.2.5 tms.FuelState

A2572 PURPOSE: Provides dynamic state information regarding fuel levels and pumps.

A2573 TOPIC USAGE: Nested

A2574 EXTENSIBILITY: extensibility(APPENDABLE)

A2575 PATTERN: Structure

A2576 ATTRIBUTES:

Name	Type and Description
<b>fuelLevel</b>	<b>float32</b> The current fuel level of the device. Annotations: <i>units</i> =liter
<b>fuelPumpRunning</b>	<b>boolean</b> True indicates that fuel is being pulled from external tank (fuel pump is on). False otherwise or if the device does not have a fuel pump.

A2579 B.9.2.6 tms.GeneratorState

A2580 PURPOSE: Report information on AVR and generator condition.

A2581 TOPIC USAGE: Nested

A2582 EXTENSIBILITY: extensibility(APPENDABLE)

A2583 PATTERN: Structure

A2584 ATTRIBUTES:

Name	Type and Description
<b>fieldCurrent</b>	<b>float32</b> field current, as controlled by AVR. Annotations: <i>units</i> =Ampere
<b>statorTemperature</b>	<b>float32</b> stator temperature. Annotations: <i>units</i> =Celsius



A2587 B.9.2.7 tms.EnergyStorageState

A2588 PURPOSE: Operating state of the energy storage device or unit.

A2589 TOPIC USAGE: Nested

A2590 EXTENSIBILITY: extensibility(APPENDABLE)

A2591 PATTERN: Structure

A2592 ATTRIBUTES:

Name	Type and Description
<b>stateOfCharge</b>	<b>float32</b> Estimated energy stored. Annotations: <i>units=joule</i>
<b>temperature</b>	<b>float32</b> Present temperature of the energy storage components. Annotations: <i>min=-273.15,max=10000,units=Celsius</i>

A2595 B.9.2.8 tms.PowerElectronicsState

A2596 PURPOSE: Operating state of the power electronics hardware.

A2597 TOPIC USAGE: Nested

A2598 EXTENSIBILITY: extensibility(APPENDABLE)

A2599 PATTERN: Structure

A2600 ATTRIBUTES:

Name	Type and Description
<b>temperature</b>	<b>float32</b> Present temperature. Annotations: <i>min=-273,max=10000,units=Celsius</i>

A2603 B.9.2.9 tms.TransformerState

A2604 PURPOSE: Provides operating information about about power transformer devices.

A2605 TOPIC USAGE: Nested

A2606 EXTENSIBILITY: extensibility(APPENDABLE)

A2607 PATTERN: Structure

A2608 ATTRIBUTES:

Name	Type and Description
placeholder	
boolean	
TODO feedback requested.	

A2611 B.9.2.10 tms.ThermalState

A2612 PURPOSE: Provides operating information of the thermal management in this device.

A2613 TOPIC USAGE: Nested

A2614 EXTENSIBILITY: extensibility(APPENDABLE)

A2615 PATTERN: Structure

A2616 ATTRIBUTES:

Name	Type and Description
thermalLoad	
ThermalLoadSequence	
Load for each thermalZone, in same order published in tms.ThermalInfo.	

A2619 B.9.2.11 tms.ThermalLoadSequence

A2620 PURPOSE: A sequence of float32.

A2621 EXTENSIBILITY: extensibility(FINAL)

A2622 PATTERN: Typedef

A2623 ORIGINAL TYPE: `sequence<float32,0,5>`

A2624 B.10 Storage State

A2625 B.10.1 Data Flows

A2626 Figure B.13 illustrates all the topics in the data flow.

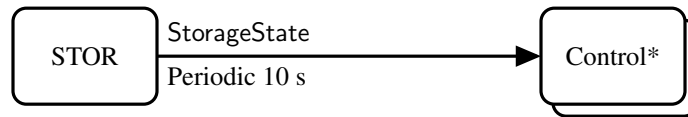


Figure B.13: Storage State data flow

A2627 Table B.31 provides an overview of how each topic is used.

Table B.31: Description of the Storage State topics.

Topic	Description
StorageState	The dynamic state of a Storage device.

A2628 Table B.32 specifies the publishers and subscribers for each topic as well as usage  
 A2629 requirements as described in Section A.8.7.

Table B.32: Participants of the Storage State topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
StorageState	REQUIRED					Pub			
		Sub	Sub	Sub					

A2630 Table B.33 specifies the timing for when sample updates are sent to each topic. These  
 A2631 parameters are defined in Section A.8.5.

Table B.33: Timing of the Storage State topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
StorageState	Periodic 10 s	10 s	1			Slow

A2632 B.10.2 Data Types

A2633 Table [B.34](#) specifies the data types that are used as sample values for each topic.

Table B.34: Data types of the Storage State topics.

Topic	Data Type
StorageState	<code>tms.stor.StorageState</code>

A2634 Topic-level data types are defined in the order shown in Table [B.34](#). Then nested data  
A2635 types are defined in the order of appearance. Cross-references are provided for data types  
A2636 that have already been defined.

A2637 B.10.2.1 tms.stor.StorageState

A2638 PURPOSE: Report key parameters about the state of the ESU.

A2639 TOPIC USAGE: StorageState

A2640 EXTENSIBILITY: extensibility(APPENDABLE)

A2641 PATTERN: Structure

A2642 ATTRIBUTES:

*TMS Data Model*  
*DRAFT 19 Apr 2021*

Name	Type and Description
deviceId	<p>tms::Fingerprint</p> <p>The device described by this structure.</p> <p>Annotations: <i>keyval</i></p>
operatingPower	<p>float32</p> <p>How much power the ESU is consuming for its present internal operation, whether charging, discharging, or idle. Equals the difference between the external ESU power output and the internal energy storage power output.</p> <p>Annotations: <i>units=watt</i></p>
runTime	<p>float32</p> <p>Cumulative time that the ESU has been actively charging or discharging. Usually displayed as an hour meter.</p> <p>Annotations: <i>units=second</i></p>
cycleCounter	<p>float32</p> <p>Cumulative estimate of the energy storage wear, measured as the number of operating charge/discharge cycles.</p>
internalVoltage	<p>float32</p> <p>Measurement indicating the DC bus voltage between the energy storage and power converter.</p> <p>Annotations: <i>units=Volt</i></p>
stateOfCharge	<p>float32</p> <p>Estimate of present charge as a fraction of rated capacity. 0=empty and 1=full. TODO: show how reserve thresholds interact, note that &gt;1 may be possible, do not saturate.</p> <p>Annotations: <i>min=0,max=1</i></p>
holdTime	<p>float32</p> <p>Estimate of how long the ESU can operate at the present power rate.</p> <p>Annotations: <i>units=second</i></p>
A2643 chargeTime	<p>tms::Curve2D</p> <p>Estimate of how long the ESU can charge at different power levels, without exceeding the highStateOfCharge.</p> <p>Annotations: <i>units=watt, second</i></p>
dischargeTime	<p>tms::Curve2D</p> <p>Estimate of how long the ESU can discharge at different power levels, without exceeding the lowStateOfCharge.</p> <p>Annotations: <i>units=watt, second</i></p>
maxChargeTime	<p>tms::Curve2D</p> <p>Estimate of how long the ESU can charge at different power levels, with battle short enabled. Must be greater than or equal to the chargeTime.</p>

A2645 B.10.2.2 tms.Fingerprint

A2646 See definition in Section [B.2.2.2](#).

A2647 B.10.2.3 tms.Curve2D

A2648 PURPOSE: Represent a curve in a two-dimensional space.

A2649 TOPIC USAGE: Nested

A2650 EXTENSIBILITY: extensibility(APPENDABLE)

A2651 PATTERN: Structure

A2652 ATTRIBUTES:

Name	Type and Description
<b>points</b>	<b>Point2DSequence</b>
	Vertices on the curve.

A2655 DESCRIPTION:

A2656 The Curve2D specifies a single vertical value, y, for each horizontal value, x. Coordinates on  
A2657 a Curve2D are written as (x, y) and the corresponding units are written as [unit x, unit y].

A2658 Curve2D defines a set of straight lines connecting between specified points. The points shall  
A2659 be “in order” with points[i].x < points[i+1].x. It is an error for two or more points to have  
A2660 the same x value. To evaluate the Curve2D extrapolation for any point along the curve, the  
A2661 identified points in the sequence are used.

A2662

A2663 To calculate the vertical y values:

- A2664 1. If x is one of the points in the sequence, the value of y is given in the sequence.
- A2665 2. If x is between two points in the sequence, then the y value is a straight-line  
A2666 interpolation between the two points.
- A2667 3. If x is before the point in the sequence, then the y value is a straight line projected  
A2668 through the first two points of the sequence.
- A2669 4. If x is after the last point in the sequence, then the y value is a straight line projected  
A2670 through the last two points of the sequence.

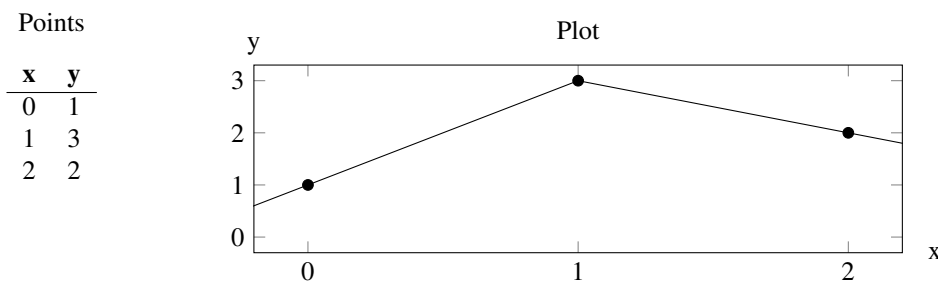


Figure B.14: Example Curve2D with three points.



A2671 B.10.2.4 tms.Point2DSequence

A2672 PURPOSE: A sequence of Point2D.

A2673 EXTENSIBILITY: extensibility(FINAL)

A2674 PATTERN: Typedef

A2675 ORIGINAL TYPE: `sequence<Point2D,0,21>`

A2676 B.10.2.5 tms.Point2D

A2677 PURPOSE: Represent a point in a two-dimensional space.

A2678 TOPIC USAGE: Nested

A2679 EXTENSIBILITY: extensibility(APPENDABLE)

A2680 PATTERN: Structure

A2681 ATTRIBUTES:

	Name	Type and Description
	x	
		float32
A2682		Horizontal coordinate.
	y	
		float32
A2683		Vertical coordinate.

A2684 B.10.2.6 tms.stor.StorageCellStateSequence

A2685 PURPOSE: A sequence of StorageCellState.

A2686 EXTENSIBILITY: extensibility(FINAL)

A2687 PATTERN: Typedef

A2688 ORIGINAL TYPE: `sequence<StorageCellState,1,999>`

A2689 B.10.2.7 tms.stor.StorageCellState

A2690 PURPOSE: Power measurement for direct current devices.

A2691 TOPIC USAGE: Nested

A2692 EXTENSIBILITY: extensibility(APPENDABLE)

A2693 PATTERN: Structure

A2694 ATTRIBUTES:

Name	Type and Description
<b>cellNumber</b>	<b>StorageCellNumber</b> Identifies the energy storage cell being reported on within a device.
<b>cycleCounter</b>	<b>float32</b> Cumulative estimate of the energy storage wear, measured as the number of operating charge/discharge cycles. Set to NaN if not available.
<b>stateOfCharge</b>	<b>float32</b> Estimate of present charge as a fraction of rated capacity. 0=empty and 1=full. Set to NaN if not available. Annotations: <i>min</i> =0, <i>max</i> =1, <i>units</i> =per unit
<b>temperature</b>	<b>float32</b> Temperature measurement of the individual storage cell. Set to NaN if not available. Annotations: <i>min</i> =-273.15, <i>max</i> =10000, <i>units</i> =Celsius
<b>voltage</b>	<b>float32</b> DC bus voltage across the energy storage cell terminals. Set to NaN if not available. Annotations: <i>units</i> =Volt

A2697 B.10.2.8 tms.stor.StorageCellNumber

A2698 PURPOSE: Identifies an energy storage cell within a device.

A2699 EXTENSIBILITY: extensibility(APPENDABLE)

A2700 PATTERN: Typedef

A2701 ORIGINAL TYPE: uint16

A2702 CONSTANTS:

Name	Type [Units], Value, and Description
<hr/>	
ONLY_CELL	
	StorageCellNumber, 0
	Number to use for a single-cell device.
MAXLEN_storageCells	
	StorageCellNumber, 50
	Maximum number of cells that a device may report.

A2705 B.11 Operator Input

A2706 Operator Input data type specifies how the operator can influence MD and MC control over  
A2707 the microgrid. These commands are used when the operator requires that a TMS device be  
A2708 in a specific state, such as off-line to perform maintenance, or to specify how a TMS device  
A2709 handles its functions; for instance, an operator might enable load curtailment by assigning  
A2710 priorities to the LOAD devices and setting the minimum priority a device must have before  
A2711 it can be considered for load curtailment.

A2712

A2713 B.11.1 Data Flows

A2714 Figure B.15 illustrates all the topics in the data flow.

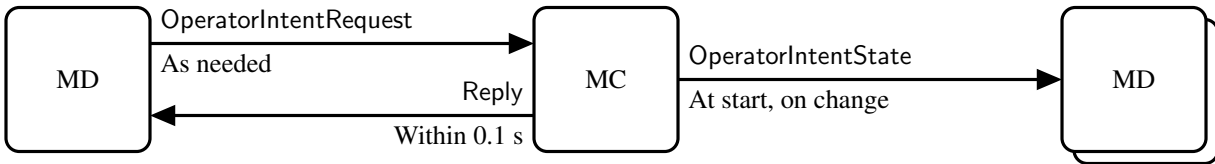


Figure B.15: Operator Input data flow

A2715 The MD and MC responsibilities for the Operator Input topic are:

- A2716 • MD (Microgrid Dashboard)
  - A2717 – Allow the operator to define and activate the `OperatorIntent` data type
  - A2718 – Only a single operator intent can be active at a time

- A2719      • MC (Microgrid Controller)
- A2720            – Transition from the current `OperatorIntent` to a new `OperatorIntent`
- A2721            – Set the default `OperatorIntent` if the MD has not set one
- A2722            – Persist the MD-activated `OperatorIntent` for use on re-start
- A2723      The operator input is specified at three control levels with different impact at each level,  
A2724 as follows:
- A2725      • Microgrid Level
- A2726            – Operational Mode
- A2727            – Spinning Reserve Thresholds
- A2728            – Load Shedding Settings
- A2729      • Device Level
- A2730            – Battle short state of the device
- A2731            – For SRC and STOR devices, device selection priority used by the MC for including  
A2732 this device as a power provider
- A2733            – For LOAD devices, the load shedding priority used by the MC for selecting a  
A2734 LOAD device to curtail when there is not enough available power in the microgrid
- A2735      • Device Power Port Level
- A2736            – Circuit Shedding Priority - this only applies to DIST device roles and controls  
A2737 the load shedding accomplished by opening the power port of a distribution box  
A2738 using the `PowerSwitchCommand`
- A2739      The MD sends the new operator input request to the MC, which sends a confirmation  
A2740 response back to the MD. The MC then sends the operator input state command to all  
A2741 MDs, including the original requester. This data flow is only initiated by operator action.  
A2742
- A2743      Table [B.35](#) provides an overview of how each topic is used.

Table B.35: Description of the Operator Input topics.

Topic	Description
<code>OperatorIntentState</code>	The last published <code>OperatorIntent</code> received by a Microgrid Controller
<code>OperatorIntentRequest</code>	Operator defined microgrid control directives that influence how the microgrid behaves.
<code>Reply</code>	The reply to a request.

- A2744      Table [B.36](#) specifies the publishers and subscribers for each topic as well as usage  
A2745 requirements as described in Section [A.8.7](#).

Table B.36: Participants of the Operator Input topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
OperatorIntentState	OPTIONAL		Pub						
		Sub		Sub					
OperatorIntentRequest	OPTIONAL	Pub							
			Sub	Sub					
Reply	REQUIRED	Pub	Pub		Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A2746 Table B.37 specifies the timing for when sample updates are sent to each topic. These  
A2747 parameters are defined in Section A.8.5.

Table B.37: Timing of the Operator Input topics.

Topic	Data Trigger	Rate Class	Burst Size	QoS Profile
OperatorIntentState	At Start, On Change	10 s	10	PublishLast
OperatorIntentRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response

#### A2748 B.11.2 Data Types

A2749 Table B.38 specifies the data types that are used as sample values for each topic.

Table B.38: Data types of the Operator Input topics.

Topic	Data Type
OperatorIntentState	<code>tms.OperatorIntentState</code>
OperatorIntentRequest	<code>tms.OperatorIntentRequest</code>
Reply	<code>tms.Reply</code>

A2750 Topic-level data types are defined in the order shown in Table B.38. Then nested data  
A2751 types are defined in the order of appearance. Cross-references are provided for data types  
A2752 that have already been defined.

A2753 B.11.2.1 tms.OperatorIntentState

A2754 PURPOSE: The active OperatorIntent.

A2755 TOPIC USAGE: OperatorIntentState

A2756 EXTENSIBILITY: extensibility(APPENDABLE)

A2757 PATTERN: Structure

A2758 ATTRIBUTES:

Name	Type and Description
<b>deviceId</b>	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
<b>activeOperatorIntent</b>	<b>OperatorIntent</b> The complete active OperatorIntent.

A2761 B.11.2.2 tms.OperatorIntentRequest

A2762 PURPOSE: Used to request activation of an OperatorIntent.

A2763 TOPIC USAGE: OperatorIntentRequest

A2764 EXTENSIBILITY: extensibility(APPENDABLE)

A2765 PATTERN: Structure

A2766 ATTRIBUTES:

Name	Type and Description
requestId	<b>GridRequest</b> The device sending this request. Annotations: <i>keyval</i>
sequenceId	<b>tms::RequestSequence</b> Request sequence data used to associate a request and returning reply.
desiredOperatorIntent	<b>OperatorIntent</b> The OperatorIntent to activate.

A2769 B.11.2.3 tms.Reply

A2770 See definition in Section [B.2.2.1](#).

A2771 B.11.2.4 tms.Fingerprint

A2772 See definition in Section [B.2.2.2](#).

A2773 B.11.2.5 tms.OperatorIntent

A2774 PURPOSE: Specifies a complete set of operator directives used by the MC to control the  
A2775 microgrid.

A2776 TOPIC USAGE: Nested

A2777 EXTENSIBILITY: extensibility(APPENDABLE)

A2778 PATTERN: Structure

A2779 ATTRIBUTES:

Name	Type and Description
<b>requestId</b>	<b>GridRequest</b> The device that defined this structure. Annotations: <i>keyval</i>
<b>intentType</b>	<b>OperatorIntentType</b> The operator intent type.
<b>microgrid</b>	<b>MicrogridIntent</b> The microgrid level directives.
<b>devices</b>	<b>DeviceIntentSequence</b> The device level directives.
<b>powerPorts</b>	<b>PowerPortIntentSequence</b> The power port level directives.

A2782 B.11.2.6 tms.GridRequest

A2783 PURPOSE: Key value for a request that targets all devices in the grid.

A2784 TOPIC USAGE: Nested

A2785 EXTENSIBILITY: extensibility(APPENDABLE)

A2786 PATTERN: Structure

A2787 ATTRIBUTES:

Name	Type and Description
<b>requestingDeviceId</b>	<b>Fingerprint</b> Identity of the device that sent this request.



A2790 B.11.2.7 tms.OperatorIntentType

A2791 PURPOSE: Specify the type of Operator Intent.

A2792 EXTENSIBILITY: extensibility(APPENDABLE)

A2793 PATTERN: Enumeration

A2794 VALUES:

Name	Description
OIT_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
OIT_DEFAULT_INTENT	Microgrid Controller defined default operator intent.
OIT_OPERATOR_DEFINED	Microgrid Dashboard activated operator intent.

A2797 B.11.2.8 tms.MicrogridIntent

A2798 PURPOSE: Specify the microgrid level operator intent.

A2799 TOPIC USAGE: Nested

A2800 EXTENSIBILITY: extensibility(APPENDABLE)

A2801 PATTERN: Structure

A2802 ATTRIBUTES:

Name	Type and Description
operatingModeValue	OperatingMode Overall operating mode for the microgrid.
loadSharingIntentValue	LoadSharingIntent Load sharing Microgrid Controller directives.
storageIntentValue	StorageIntent Storage device Microgrid Controller directives.

A2805 B.11.2.9 tms.OperatingMode

A2806 PURPOSE: Indicate the microgrid level operating mode

A2807 EXTENSIBILITY: extensibility(APPENDABLE)

A2808 PATTERN: Enumeration

A2809 VALUES:

Name	Description
------	-------------

---

OPM_UNKNOWN	
-------------	--

	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
--	---

OPM_NORMAL	
------------	--

	Normal operating mode.
--	------------------------

OPM_EMERGENCY	
---------------	--

	Emergency operating mode.
--	---------------------------

OPM_SILENT_WATCH	
------------------	--

	Silent watch mode.
--	--------------------

---

A2812 B.11.2.10 tms.LoadSharingIntent

A2813 PURPOSE: Specify the microgrid level load sharing intent.

A2814 TOPIC USAGE: Nested

A2815 EXTENSIBILITY: extensibility(APPENDABLE)

A2816 PATTERN: Structure

A2817 ATTRIBUTES:

Name	Type and Description
increaseCapacityThreshold	float32 Per unit of used capacity that triggers the MC to add additional power source to the microgrid. Per unit is in the range 0 - 1. Annotations: <i>min</i> =0, <i>max</i> =1, <i>units</i> =per unit
decreaseCapacityThreshold	float32 Per unit of used capacity that triggers the MC to remove a power source from the microgrid. Per unit is in the range 0 - 1 and must be less than increaseCapacityThreshold. Annotations: <i>min</i> =0, <i>max</i> =1, <i>units</i> =per unit
decreaseCapacityDuration	Duration The time duration the decreaseCapacityThreshold must be met before the Microgrid Controller removes a power source from the microgrid. This duration prevents power sources from being added and removed in rapid succession.
priorityLoadCurtailementAllowed	OperatorPriorityType The priority above which the Microgrid Controller is authorized to trigger load curtailment. Load curtailment is accomplished by stopped a load device. A value of OPT_ALWAYS_OPERATE disables load curtailment.
priorityPowerPortSheddingAllowed	OperatorPriorityType The priority above which the MC is authorized to trigger port power load shedding. Port power load shedding is accomplished by opening a power port on a distribution device. A value of OPT_ALWAYS_OPERATE disables circuit load shedding.

A2819

A2820 B.11.2.11 tms.Duration

A2821 PURPOSE: Specify a time interval. The total duration is the sum of seconds and  
A2822 nanoseconds.

A2823 TOPIC USAGE: Nested

A2824 EXTENSIBILITY: extensibility(APPENDABLE)

A2825 PATTERN: Structure

A2826 ATTRIBUTES:

Name	Type and Description
<b>second</b>	uint64 Time in seconds.
<b>nanosecond</b>	uint64 Time in nanoseconds.

A2829 B.11.2.12 tms.OperatorPriorityType

A2830 PURPOSE: Specify the type of Operator priority.

A2831 EXTENSIBILITY: extensibility(APPENDABLE)

A2832 PATTERN: Enumeration

A2833 VALUES:

Name	Description
OPT_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
OPT_NEVER_OPERATE	Transition the device to SS_ON or open the power port.
OPT_ALWAYS_OPERATE	Transition the device to SS_SOURCING or close the power port.
OPT_NUMERIC_RANK	Priority based on OperatorPriority.numericRank.

A2836 B.11.2.13 tms.StorageIntent

A2837 PURPOSE: Specify the storage device intent.

A2838 TOPIC USAGE: Nested

A2839 EXTENSIBILITY: extensibility(APPENDABLE)

A2840 PATTERN: Structure

A2841 ATTRIBUTES:

Name	Type and Description
reservedStateOfCharge	float32 The desired total state of charge to hold in reserve. Annotations: <i>min</i> =0, <i>max</i> =1, <i>units</i> =per unit
timeTillReservedSoc	ClockMonotonic Time till the reservedStateOfCharge is reached and maintained.
priorityLoadSheddingAllowed	OperatorPriorityType The priority above which the MC is authorized to trigger load shedding. Load shedding is accomplished by stopped a smart load device. A value of OPT_ALWAYS_OPERATE disables load curtailment.
priorityPowerPortSheddingAllowed	OperatorPriorityType The priority above which the MC is authorized to trigger power port load curtailment. Power port shedding is accomplished by opening a power port on a distribution device. A value of OPT_ALWAYS_OPERATE disables circuit load shedding.
reservedDeviceIds	DeviceFingerprintSequence The list of storage devices that should reach the reservedStateOfCharge. At empty list indicates all available storage devices.

A2843

A2844 B.11.2.14 tms.ClockMonotonic

A2845 See definition in Section [B.8.2.3](#).

A2846 B.11.2.15 tms.DeviceFingerprintSequence

A2847 PURPOSE: A sequence of DeviceFingerprint.

A2848 EXTENSIBILITY: extensibility(FINAL)

A2849 PATTERN: Typedef

A2850 ORIGINAL TYPE: `sequence<DeviceFingerprint,0,MAX_PORTS>`

A2851 B.11.2.16 tms.DeviceFingerprint

A2852 PURPOSE: Periodic indication of device availability.

A2853 TOPIC USAGE: Nested

A2854 EXTENSIBILITY: extensibility(APPENDABLE)

A2855 PATTERN: Structure

A2856 ATTRIBUTES:

Name	Type and Description
------	----------------------

<code>deviceId</code>	
-----------------------	--

	<b>Fingerprint</b>
--	--------------------

	The device described by this structure.
--	---

A2859 B.11.2.17 tms.DeviceIntentSequence

A2860 PURPOSE: A sequence of DeviceIntent.

A2861 EXTENSIBILITY: extensibility(FINAL)

A2862 PATTERN: Typedef

A2863 ORIGINAL TYPE: `sequence<DeviceIntent,0,100>`

A2864 B.11.2.18 tms.DeviceIntent

A2865 PURPOSE: Specify the device level operator intent.

A2866 TOPIC USAGE: Nested

A2867 EXTENSIBILITY: extensibility(APPENDABLE)

A2868 PATTERN: Structure

A2869 ATTRIBUTES:

Name	Type and Description
deviceId	
	Fingerprint
	The device described by this structure.
battleShort	
A2870	boolean
	The device Battle Short state.
priority	
	OperatorPriority
A2871	The operator assigned priority of the device.

A2872 B.11.2.19 tms.OperatorPriority

A2873 PURPOSE: Specifies the operator priority for a single device or power port.

A2874 TOPIC USAGE: Nested

A2875 EXTENSIBILITY: extensibility(APPENDABLE)

A2876 PATTERN: Structure

A2877 ATTRIBUTES:

Name	Type and Description
priorityType	
	OperatorPriorityType
	The operator priority type.
A2878	numericRank
	int16
	Rank values indicate priority order, where 0 is the highest priority and increasing values are lower priority. The numericRank is used when priorityType is OPT_NUMERIC_RANK.
A2879	

A2880 B.11.2.20 tms.PowerPortIntentSequence

A2881 PURPOSE: A sequence of PowerPortIntent.

A2882 EXTENSIBILITY: extensibility(FINAL)

A2883 PATTERN: Typedef

A2884 ORIGINAL TYPE: `sequence<PowerPortIntent,0,4000>`

A2885 B.11.2.21 tms.PowerPortIntent

A2886 PURPOSE: Specify the device power port level operator intent.

A2887 TOPIC USAGE: Nested

A2888 EXTENSIBILITY: extensibility(APPENDABLE)

A2889 PATTERN: Structure

A2890 ATTRIBUTES:

Name	Type and Description
deviceId	
	Fingerprint
	The device described by this structure.
portNumber	
A2891	PowerPortNumber
	Number shown on the device exterior to represent this port.
priority	
	OperatorPriority
A2892	The operator assigned priority of this power port.

A2893 B.11.2.22 tms.PowerPortNumber

A2894 See definition in Section [B.2.2.4](#).

A2895 B.11.2.23 tms.RequestSequence

A2896 See definition in Section [B.2.2.5](#).

A2897 B.11.2.24 tms.ConfigId

A2898 See definition in Section [B.2.2.3](#).

A2899 B.11.2.25 tms.ReplyStatus

A2900 See definition in Section [B.2.2.6](#).



A2901 B.11.2.26 tms.String32

A2902 See definition in Section [B.2.2.7](#).

A2903 B.12 Diagnostics Reporting

A2904 TMS has active diagnostics reporting that provide notifications when fault conditions or  
A2905 other defined diagnostic reporting is required from any TMS device. This can include  
A2906 informational notes, warnings, and errors. Error messaging should be released only to  
A2907 authorized TMS personnel. Consideration should be given to how fault notifications are  
A2908 handled such that information is conveyed for corrective actions but without revealing  
A2909 anything in the TMS that could be exploited by adversaries.

A2910 Any TMS device that experiences a fault initiates an active diagnostics error message to  
A2911 the MCs and MDs in the grid.

A2912

A2913 B.12.1 Data Flows

A2914 Figure [B.16](#) illustrates all the topics in the data flow.

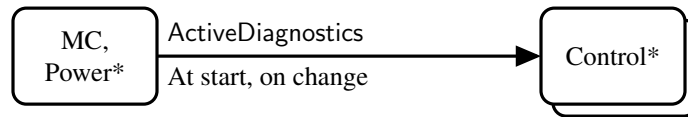


Figure B.16: Diagnostics Reporting data flow

A2915 The data flow does not have a controller response because TMS does not define the active  
A2916 diagnostics response hierarchy. TMS only identifies the message communication must occur.

A2917 For example, a TMS generator sends a diagnostics message that the engine overheating.  
A2918 The MC can choose to take the following actions:

- A2919 • No action - the fault will clear itself (load goes away when the generator shuts down)
- A2920 • or potential equipment damage acceptable to keep it active (battle conditions)
- A2921 • Send a command to a different device - start another generator or shed a load
- A2922 • Shutdown - send command to shutdown the device that is overheating

A2923 Note: The data model does not presently include support for diagnostics troubleshooting  
A2924 so this will be performed by other interfaces.

A2925

A2926 Table [B.39](#) provides an overview of how each topic is used.

Table B.39: Description of the Diagnostics Reporting topics.

Topic	Description
ActiveDiagnostics	Reports a device diagnostics state including information and alarm states.

A2927 Table B.40 specifies the publishers and subscribers for each topic as well as usage  
A2928 requirements as described in Section A.8.7.

Table B.40: Participants of the Diagnostics Reporting topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
ActiveDiagnostics	REQUIRED		Pub		Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A2929 Table B.41 specifies the timing for when sample updates are sent to each topic. These  
A2930 parameters are defined in Section A.8.5.

Table B.41: Timing of the Diagnostics Reporting topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
ActiveDiagnostics	At Start, On Change	10 s	10			PublishLast

## A2931 B.12.2 Data Types

A2932 The message patterns in this section are based on SAE J1939-73 Diagnostic Trouble Codes  
A2933 (DTCs). The DTCs have been extended to address extensibility issues for TMS. Since these  
A2934 objects are based on J1939, that family of standards may be referenced to supplement this  
A2935 standard. Other aspects of that standard, including wire formats and timing requirements,  
A2936 do not apply.

A2937 Table B.42 specifies the data types that are used as sample values for each topic.

Table B.42: Data types of the Diagnostics Reporting topics.

Topic	Data Type
ActiveDiagnostics	<code>tms.ActiveDiagnosticMessages</code>

A2938 Topic-level data types are defined in the order shown in Table B.42. Then nested data  
A2939 types are defined in the order of appearance. Cross-references are provided for data types  
A2940 that have already been defined.

A2941 B.12.2.1 tms.ActiveDiagnosticMessages

A2942 PURPOSE: ActiveDiagnosticMessages

A2943 TOPIC USAGE: ActiveDiagnostics

A2944 EXTENSIBILITY: extensibility(APPENDABLE)

A2945 PATTERN: Structure

A2946 ATTRIBUTES:

Name	Type and Description
deviceId	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
codes	<b>DiagnosticTroubleCodeSequence</b> Active trouble codes. A given suspect parameter can only appear once in this sequence.
overflow	<b>uint32</b> Count of trouble codes that were truncated from the codes because the maximum sequence length was exceeded. A count of 0 indicates that all active trouble codes are listed.

A2949 B.12.2.2 tms.Fingerprint

A2950 See definition in Section [B.2.2.2](#).

A2951 B.12.2.3 tms.DiagnosticTroubleCodeSequence

A2952 PURPOSE: A sequence of DiagnosticTroubleCode.

A2953 EXTENSIBILITY: extensibility(FINAL)

A2954 PATTERN: Typedef

A2955 ORIGINAL TYPE: `sequence<DiagnosticTroubleCode,0,64>`

A2956 B.12.2.4 tms.DiagnosticTroubleCode

A2957 PURPOSE: Represent an individual alarm or fault condition originating from the indicated  
A2958 sensor or subsystem.

A2959 TOPIC USAGE: Nested

A2960 EXTENSIBILITY: extensibility(APPENDABLE)

A2961 PATTERN: Structure

A2962 ATTRIBUTES:

Name	Type and Description
suspectParameter	<p>SuspectParameterNumber</p> <p>Parameter representing this DTC.</p>
failureMode	<p>FailureModeIndicator</p> <p>Type of fault for this parameter.</p>
occurrenceCount	<p>uint32</p> <p>Number of times this DTC was raised. This field increments on a state change from inactive to active. In order to increment, the cleared state must be sensed. The count must not increment simply due to restarts. The count should be stored in non-volatile memory in order to preserve it over power cycles. The count saturates at 4294967294 (<math>2^{32} - 2</math>). The value 4294967295 (<math>2^{32} - 1</math>) indicates that the count is not available.</p>
timeRaised	<p>ClockMonotonic</p> <p>Timestamp when this DTC was raised or activated.</p>
severity	<p>DtcSeverity</p> <p>Expected impact of the DTC on grid operation. Intended to support automatic handling of parameters that are unknown to the receiving system.</p>
estimateTime	<p>optional&lt;ClockMonotonic&gt;</p> <p>Estimate of when the DTC severity will change, assuming continued operation under present conditions.</p>
estimateSeverity	<p>optional&lt;DtcSeverity&gt;</p> <p>Expected DTC severity at the estimateTime.</p>
hint	<p>String32</p> <p>Short, human-readable text description of the DTC. This should summarize to the operator what the issue is (e.g. "oil pressure" or "low oil pressure"). Intended to support human operators for parameters that are unknown to the receiving system.</p>

A2963 Timestamp when this DTC was raised or activated.

A2964

A2965 B.12.2.5 tms.SuspectParameterNumber

A2966 PURPOSE: Identifies the sensor or subsystem at issue or a particular event or condition.

A2967 EXTENSIBILITY: extensibility(APPENDABLE)

A2968 PATTERN: Typedef

A2969 ORIGINAL TYPE: uint32

A2970 B.12.2.6 tms.FailureModeIndicator

A2971 PURPOSE: Describes the type of fault

A2972 EXTENSIBILITY: extensibility(APPENDABLE)

A2973 PATTERN: Typedef

A2974 ORIGINAL TYPE: octet

A2975 CONSTANTS:

Name	Type [Units], Value, and Description
------	--------------------------------------

---

FMI_HIGH_MOST_SEVERE	
----------------------	--

octet, 0	
----------	--

Data valid but above normal operational range, most severe level.	
---	--

FMI_LOW_MOST_SEVERE	
---------------------	--

octet, 1	
----------	--

Data valid but below normal operational range, most severe level.	
---	--

FMI_HIGH_LEAST_SEVERE	
-----------------------	--

octet, 15	
-----------	--

A2976 Data valid but above normal operational range, least severe level.

FMI_HIGH_MODERATELY_SEVERE	
----------------------------	--

octet, 16	
-----------	--

Data valid but above normal operational range, moderately severe level.	
---	--

FMI_LOW_LEAST_SEVERE	
----------------------	--

octet, 17	
-----------	--

Data valid but below normal operational range, least severe level.	
--	--

FMI_LOW_MODERATELY_SEVERE	
---------------------------	--

octet, 18	
-----------	--

A2977 Data valid but below normal operational range, moderately severe level.

---

A2978 B.12.2.7 tms.ClockMonotonic

A2979 See definition in Section [B.8.2.3](#).

A2980 B.12.2.8 tms.DtcSeverity

A2981 PURPOSE: Describe how a DTC affects the availability of the device

A2982 EXTENSIBILITY: extensibility(APPENDABLE)

A2983 PATTERN: Enumeration

A2984 VALUES:

Name	Description
SEV_0_NOT_SPECIFIED	[PROPOSAL] Undefined severity. Never transmitted in this version of the standard.
SEV_1_CLEAR	[PROPOSAL] DTC has cleared. Never transmitted in this version of the standard.
SEV_2_INFORMATIVE	Informational only. No performance impact is expected.
SEV_3_PREVENTATIVE	Preventative maintenance has been scheduled.
SEV_4_DEGRADED	Poor configuration or operation resulting in degraded performance, including reduced efficiency and increased emissions.
SEV_5_WARNING	Increased wear and premature failure likely.
SEV_6_MINOR	The device may be failing to meet operating performance criteria. The DTC may represent cause or effect.
SEV_7_MAJOR	This condition may cause protection logic to enter a controlled shutdown sequence. A corrective action may be required before the device can be restarted.
SEV_8_CRITICAL	A loss of essential functionality has been detected. Routine maintenance or repair may be required.
SEV_9_FATAL	A loss of essential functionality has been detected. Permanent damage may have occurred.
SEV_10_HUMAN_SAFETY	Potential user safety risk.

A2987 B.12.2.9 tms.String32

A2988 See definition in Section [B.2.2.7](#).

A2989 B.13 Microgrid Controller Selection

A2990 All power devices shall follow the MC selection algorithm defined in Section [A.7](#). All power  
A2991 devices shall report which Microgrid Controller they are accepting requests from.

A2992

### A2993 B.13.1 Data Flows

A2994 Figure B.17 illustrates all the topics in the data flow.

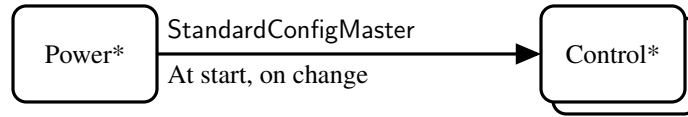


Figure B.17: Microgrid Controller Selection data flow

A2995 Table B.43 provides an overview of how each topic is used.

Table B.43: Description of the Microgrid Controller Selection topics.

Topic	Description
StandardConfigMaster	The selected master Microgrid Controller for a specific power device.

A2996 Table B.44 specifies the publishers and subscribers for each topic as well as usage  
A2997 requirements as described in Section A.8.7.

Table B.44: Participants of the Microgrid Controller Selection topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
StandardConfigMaster	REQUIRED				Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A2998 Table B.45 specifies the timing for when sample updates are sent to each topic. These  
A2999 parameters are defined in Section A.8.5.

Table B.45: Timing of the Microgrid Controller Selection topics.

Topic	Data Trigger	Rate Class	Burst Size	QoS Profile
StandardConfigMaster	At Start, On Change	1 s	1	PublishLast

### A3000 B.13.2 Data Types

A3001 Table B.46 specifies the data types that are used as sample values for each topic.

Table B.46: Data types of the Microgrid Controller Selection topics.

Topic	Data Type
StandardConfigMaster	tms.StandardConfigMaster

A3002 Topic-level data types are defined in the order shown in Table [B.46](#). Then nested data  
A3003 types are defined in the order of appearance. Cross-references are provided for data types  
A3004 that have already been defined.

#### A3005 B.13.2.1 tms.StandardConfigMaster

A3006 PURPOSE: Identify which :MC currently has write access to the standard configurations.

A3007 TOPIC USAGE: StandardConfigMaster

A3008 EXTENSIBILITY: extensibility(APPENDABLE)

A3009 PATTERN: Structure

A3010 ATTRIBUTES:

Name	Type and Description
<b>deviceId</b>	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
<b>masterId</b>	<b>Fingerprint</b> The MC that currently controls this device. If no MC is selected or available, use the NULL Fingerprint value (all 0s)

A3013 DESCRIPTION:

A3014 This information is published by each controllable device.

#### A3015 B.13.2.2 tms.Fingerprint

A3016 See definition in Section [B.2.2.2](#).

### A3017 B.14 Read Configuration Settings

A3018 Power devices should allow the MC to read out their configuration settings. Since this is  
A3019 an infrequent operation, the MC must request them explicitly and the response goes to the  
A3020 requesting MC only.

A3021

#### A3022 B.14.1 Data Flows

A3023 Figure [B.18](#) illustrates all the topics in the data flow.



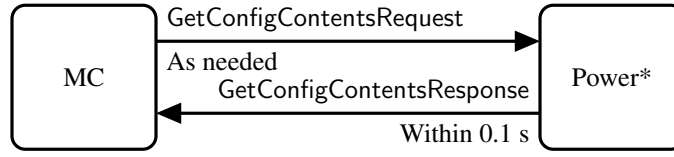


Figure B.18: Read Configuration Settings data flow

A3024 In a distributed system, it can be difficult to track the state of a remote device.  
 A3025 Commands may be sent from other devices, messages may be lost, devices may restart,  
 A3026 and other complications may occur. This request allows the MC to get a complete snapshot  
 A3027 of a device configuration at a single instant in time.

A3028 This request instructs the receiving device to send all commands stored in a configuration  
 A3029 to the requesting device.

A3030

A3031 Table B.47 provides an overview of how each topic is used.

Table B.47: Description of the Read Configuration Settings topics.

Topic	Description
GetConfigContentsRequest	Retrieve all commands stored in a device configuration.
GetConfigContentsResponse	Response to a GetConfigContentsRequest

A3032 Table B.48 specifies the publishers and subscribers for each topic as well as usage  
 A3033 requirements as described in Section A.8.7.

Table B.48: Participants of the Read Configuration Settings topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
GetConfigContentsRequest	OPTIONAL		Pub						
GetConfigContentsResponse	OPTIONAL			Sub	Sub	Sub	Sub	Sub	Sub
				Sub	Sub				

A3034 Table B.49 specifies the timing for when sample updates are sent to each topic. These  
 A3035 parameters are defined in Section A.8.5.

Table B.49: Timing of the Read Configuration Settings topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
GetConfigContentsRequest	As Needed	10 s	1			Command
GetConfigContentsResponse	Within 0.1 s	10 s	1			Response

A3036 If the requesting device lacks permission to access the `config` index, then `REPLY_-`  
 A3037 `BAD_REQUEST` is returned without any configuration data.

A3038

A3039 B.14.2 Data Types

A3040 Table B.50 specifies the data types that are used as sample values for each topic.

Table B.50: Data types of the Read Configuration Settings topics.

Topic	Data Type
GetConfigContentsRequest	<code>tms.GetConfigContentsRequest</code>
GetConfigContentsResponse	<code>tms.GetConfigContentsResponse</code>

A3041 Topic-level data types are defined in the order shown in Table B.50. Then nested data  
A3042 types are defined in the order of appearance. Cross-references are provided for data types  
A3043 that have already been defined.

A3044 B.14.2.1 `tms.GetConfigContentsRequest`

A3045 PURPOSE: Retrieve all commands stored in a configuration.

A3046 TOPIC USAGE: `GetConfigContentsRequest`

A3047 EXTENSIBILITY: extensibility(APPENDABLE)

A3048 PATTERN: Structure

A3049 ATTRIBUTES:

Name	Type and Description
<code>requestId</code>	
	<code>DeviceConfigRequest</code>
A3050	Unique request identifier, including the identity of the :MC making the request. Annotations: <i>keyval</i>
<code>sequenceId</code>	
	<code>tms::RequestSequence</code>
A3051	Request sequence data used to associate a request and returning reply.

A3052 B.14.2.2 tms.GetConfigContentsResponse

A3053 PURPOSE: Response to a GetConfigContentsRequest.

A3054 TOPIC USAGE: GetConfigContentsResponse

A3055 EXTENSIBILITY: extensibility(APPENDABLE)

A3056 PATTERN: Structure

A3057 ATTRIBUTES:

Name	Type and Description
<b>relatedRequestId</b>	<b>DeviceConfigRequest</b> Copy of the corresponding requestId. Annotations: <i>keyval</i>
<b>status</b>	<b>ReplyStatus</b> Indication of success or failure.
<b>powerDeviceConfigValue</b>	<b>PowerDeviceConfig</b> The requested power device configuration.

A3060 B.14.2.3 tms.DeviceConfigRequest

A3061 PURPOSE: Key value for a request that targets one power device configuration.

A3062 TOPIC USAGE: Nested

A3063 EXTENSIBILITY: extensibility(APPENDABLE)

A3064 PATTERN: Structure

A3065 ATTRIBUTES:

Name	Type and Description
<b>requestingDeviceId</b>	<b>Fingerprint</b> Identity of the device that sent this request.
<b>targetDeviceId</b>	<b>Fingerprint</b> Identity of the device that should receive this request.
<b>config</b>	<b>ConfigId</b> Power device configuration that this request belongs to.

A3068 B.14.2.4 tms.Fingerprint

A3069 See definition in Section [B.2.2.2](#).

A3070 B.14.2.5 tms.ConfigId

A3071 See definition in Section [B.2.2.3](#).

A3072 B.14.2.6 tms.RequestSequence

A3073 See definition in Section [B.2.2.5](#).

A3074 B.14.2.7 tms.ReplyStatus

A3075 See definition in Section [B.2.2.6](#).

A3076 B.14.2.8 tms.String32

A3077 See definition in Section [B.2.2.7](#).

A3078 B.14.2.9 tms.PowerDeviceConfig

A3079 PURPOSE: Specified a single power device configuration.

A3080 TOPIC USAGE: Nested

A3081 EXTENSIBILITY: extensibility(APPENDABLE)

A3082 PATTERN: Structure

A3083 ATTRIBUTES:

Name	Type and Description
<b>powerSwitchCommands</b>	<b>PowerSwitchCommandSequence</b> A list of power switch commands.
<b>sourceTransition</b>	<b>optional&lt;SourceTransitionRequest&gt;</b> A source transition request.
<b>controlParameter</b>	<b>optional&lt;ControlParameterRequest&gt;</b> A control parameter request.
<b>groundingCommands</b>	<b>GroundingCommandSequence</b> A list of grounding commands.
<b>acLoadSharingRequest</b>	<b>optional&lt;ac::LoadSharingRequest&gt;</b> An alternating current load sharing request.
<b>dcLoadSharingRequest</b>	<b>optional&lt;dc::LoadSharingRequest&gt;</b> A direct current load sharing request.
<b>storageControlRequest</b>	<b>optional&lt;stor::StorageControlRequest&gt;</b> A storage control request.

A3085

A3086 B.14.2.10 tms.PowerSwitchCommandSequence

A3087 PURPOSE: A sequence of PowerSwitchCommand.

A3088 EXTENSIBILITY: extensibility(FINAL)

A3089 PATTERN: Typedef

A3090 ORIGINAL TYPE: `sequence<PowerSwitchCommand,0,MAX_PORTS>`

A3091 B.14.2.11 tms.PowerSwitchCommand

A3092 PURPOSE: Remotely transition a power switch.

A3093 TOPIC USAGE: PowerSwitchRequest

A3094 EXTENSIBILITY: extensibility(APPENDABLE)

A3095 PATTERN: Structure

A3096 ATTRIBUTES:

Name	Type and Description
requestId	PowerPortConfigRequest Identity of this request. Annotations: <i>keyval</i>
sequenceId	tms::RequestSequence Request sequence data used to associate a request and returning reply.
continuity	DesiredCircuitContinuity Desired continuity through the switch.
minV	float32 Minimum synchronization RMS voltage. Annotations: <i>units</i> =Volt
maxV	float32 Maximum synchronization RMS voltage. Annotations: <i>units</i> =Volt
minF	float32 Minimum synchronization frequency. Annotations: <i>units</i> =hertz
maxF	float32 Maximum synchronization frequency. Annotations: <i>units</i> =hertz
maxPhase	float32 Maximum phase misalignment (absolute value). Annotations: <i>units</i> =radian

A3097

A3098

A3099 B.14.2.12 tms.PowerPortConfigRequest

A3100 PURPOSE: Key value for a request that targets one power port of a device.

A3101 TOPIC USAGE: Nested

A3102 EXTENSIBILITY: extensibility(APPENDABLE)

A3103 PATTERN: Structure

A3104 ATTRIBUTES:

Name	Type and Description
<u>requestingDeviceId</u>	<u>Fingerprint</u> Identity of the device that sent this request.
<u>targetDeviceId</u>	<u>Fingerprint</u> Identity of the device that should received this request.
<u>config</u>	<u>ConfigId</u> Power device configuration that this request belongs to.
<u>portNumber</u>	<u>PowerPortNumber</u> Power port number that should change state.

A3107 B.14.2.13 tms.PowerPortNumber

A3108 See definition in Section [B.2.2.4](#).



A3109 B.14.2.14 tms.DesiredCircuitContinuity

A3110 PURPOSE: Indicate the desired circuit continuity of a switch.

A3111 EXTENSIBILITY: extensibility(APPENDABLE)

A3112 PATTERN: Enumeration

A3113 VALUES:

Name	Description
------	-------------

---

DCC\_UNKNOWN

Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.

DCC\_OPEN

Transition to CC\_OPEN.

DCC\_CLOSED

Transition to CC\_CLOSED. Predicated on AuthorizationToEnergizeRequest.

A3114 DCC\_SYNC\_CLOSED

Transition to CC\_CLOSED when synchronization is achieved. May close to a dead circuit, predicated on AuthorizationToEnergizeRequest. May act as DCC\_CLOSED\_INTERNAL when the internal circuit is dead.

DCC\_CLOSED\_INTERNAL

Transition to CC\_CLOSED. One side of the switch must be a circuit enclosed inside a device with protective guards.

DCC\_NO\_CHANGE

No transition. May be used instead of DCC\_CLOSED in non-active settings groups.

---

A3115

A3116 B.14.2.15 tms.SourceTransitionRequest

A3117 PURPOSE: Change source state.

A3118 TOPIC USAGE: SourceTransitionRequest

A3119 EXTENSIBILITY: extensibility(APPENDABLE)

A3120 PATTERN: Structure

A3121 ATTRIBUTES:

Name	Type and Description
<b>requestId</b>	<b>DeviceConfigRequest</b> Identity of this request. Annotations: <i>keyval</i>
A3122 <b>sequenceId</b>	<b>tms::RequestSequence</b> Request sequence data used to associate a request and returning reply.
<b>desiredTransition</b>	<b>SourceTransition</b> Transition the device should complete.

A3124 B.14.2.16 tms.SourceTransition

A3125 PURPOSE: Indicate a transition in the source state transition model.

A3126 EXTENSIBILITY: extensibility(APPENDABLE)

A3127 PATTERN: Enumeration

A3128 VALUES:

Name	Description
ST_UNKNOWN	Uninitialized value.
ST_NONE	No transition (stay in present state).
ST_POWER_UP	Transition from OFF to ON.
ST_POWER_DOWN	Transition from ON to OFF.
ST_START	Transition from ON to RUNNING.
ST_STOP	Transition from RUNNING to ON.
ST_CONNECT	Transition from RUNNING to SOURCING.
ST_DISCONNECT	Transition from SOURCING to RUNNING.
ST_JUMP_START	Transition from ON to SOURCING.
ST_RAPID_STOP	Transition from SOURCING to ON.

A3131 B.14.2.17 tms.ControlParameterRequest

A3132 PURPOSE: Request for a control parameter value.

A3133 TOPIC USAGE: ControlParameterRequest

A3134 EXTENSIBILITY: extensibility(APPENDABLE)

A3135 PATTERN: Structure

A3136 ATTRIBUTES:

Name	Type and Description
<code>requestId</code>	<code>DeviceConfigRequest</code> Identity of this request. Annotations: <i>keyval</i>
A3137 <code>sequenceId</code>	<code>tms::RequestSequence</code> Request sequence data used to associate a request and returning reply.
<code>parameters</code>	<code>ParameterValueSequence</code> Selected parameter values to update (may be partial or full list).

A3139 B.14.2.18 tms.ParameterValueSequence

A3140 PURPOSE: A sequence of ParameterValue.

A3141 EXTENSIBILITY: extensibility(FINAL)

A3142 PATTERN: Typedef

A3143 ORIGINAL TYPE: `sequence<ParameterValue,0,128>`

A3144 B.14.2.19 tms.ParameterValue

A3145 PURPOSE: Provide the value for a parameter.

A3146 TOPIC USAGE: Nested

A3147 EXTENSIBILITY: extensibility(APPENDABLE)

A3148 PATTERN: Structure

A3149 ATTRIBUTES:

Name	Type and Description
<b>name</b>	<b>String1_32</b> Name of this parameter.
<b>value</b>	<b>float32</b> Value of this parameter, with units according to ParameterMetadata.

A3152 B.14.2.20 tms.String1-32

A3153 See definition in Section [B.5.2.10](#).

A3154 B.14.2.21 tms.GroundingCommandSequence

A3155 PURPOSE: A sequence array of GroundingCommand.

A3156 EXTENSIBILITY: extensibility(FINAL)

A3157 PATTERN: Typedef

A3158 ORIGINAL TYPE: `sequence<GroundingCommand,0,MAX_PORTS>`

A3159 B.14.2.22 tms.GroundingCommand

A3160 PURPOSE: Remotely transition a grounding circuit switch.

A3161 TOPIC USAGE: GroundingCircuitRequest

A3162 EXTENSIBILITY: extensibility(APPENDABLE)

A3163 PATTERN: Structure

A3164 ATTRIBUTES:

Name	Type and Description
<code>requestId</code>	<p><code>DeviceConfigRequest</code> Identity of this request. Annotations: <i>keyval</i></p>
<code>sequenceId</code>	<p><code>tms::RequestSequence</code> Request sequence data used to associate a request and returning reply.</p>
A3165 <code>groundNumber</code>	<p><code>GroundingCircuitNumber</code> Grounding circuit containing the switch that should change state.</p>
<code>control</code>	<p><code>DesiredCircuitContinuity</code> Desired continuity through the grounding control switch.</p>
<code>pulse</code>	<p><code>DesiredCircuitContinuity</code> Desired continuity through the grounding pulse switch.</p>

A3166

A3167 B.14.2.23 tms.GroundingCircuitNumber

A3168 See definition in Section [B.5.2.46](#).

A3169 B.14.2.24 tms.ac.LoadSharingRequest

A3170 PURPOSE: Sets load sharing parameters for alternating current devices.

A3171 TOPIC USAGE: LoadSharingRequest

A3172 EXTENSIBILITY: extensibility(APPENDABLE)

A3173 PATTERN: Structure

A3174 ATTRIBUTES:

Name	Type and Description
<code>requestId</code>	<code>tms::PowerPortConfigRequest</code> Identity of this request. Annotations: <i>keyval</i>
A3175 <code>sequenceId</code>	<code>tms::RequestSequence</code> Request sequence data used to associate a request and returning reply.
<code>parameters</code>	<code>LoadSharing</code> New value to use.

A3177 B.14.2.25 tms.ac.LoadSharing

A3178 PURPOSE: Represents the load sharing parameters for alternating current devices.

A3179 TOPIC USAGE: Nested

A3180 EXTENSIBILITY: extensibility(APPENDABLE)

A3181 PATTERN: Structure

A3182 ATTRIBUTES:

Name	Type and Description
<code>frequencyPowerCurve</code>	<code>tms::ControlCurve</code> Real power to frequency relationship curve, f_P. Annotations: <i>units</i> =watt, hertz
A3183 <code>voltageReactiveCurve</code>	<code>tms::ControlCurve</code> Reactive power to voltage relationship curve, v_Q. Annotations: <i>units</i> =volt ampere reactive, volts
<code>dls</code>	<code>tms::DLSSConfig</code> Digital load sharing configuration.

A3185 B.14.2.26 tms.ControlCurve

A3186 PURPOSE: Specify output variable y as a function of input variable x.

A3187 TOPIC USAGE: Nested

A3188 EXTENSIBILITY: extensibility(APPENDABLE)

A3189 PATTERN: Structure

A3190 ATTRIBUTES:

Name	Type and Description
minimum	<b>Curve2D</b> Minimum acceptable value of y. Devices should operate above this curve.
nominal	<b>Curve2D</b> Nominal value of y. Used as the set-point for PID controls.
maximum	<b>Curve2D</b> Maximum acceptable value of y. Devices should operate below this curve.

A3193 B.14.2.27 tms.Curve2D

A3194 See definition in Section [B.10.2.3](#).

A3195 B.14.2.28 tms.Point2DSequence

A3196 See definition in Section [B.10.2.4](#).

A3197 B.14.2.29 tms.Point2D

A3198 See definition in Section [B.10.2.5](#).



A3199 B.14.2.30 tms.DLSConfig

A3200 PURPOSE: Digital load sharing configuration parameters.

A3201 TOPIC USAGE: Nested

A3202 EXTENSIBILITY: extensibility(APPENDABLE)

A3203 PATTERN: Structure

A3204 ATTRIBUTES:

Name	Type and Description
<b>sendTopics</b>	<p><b>TopicList</b> List of topics for receiving DLSMeasurement data. If empty, then do not receive any data. The list must contain unique values.</p>
<b>receiveTopics</b>	<p><b>TopicList</b> List of topics for sending DLSMeasurement data. If empty, then do not send any data. The list must contain unique values.</p>
<b>gainReal</b>	<p><b>float32</b> Gain on the real-power sharing signal. Setting this to 0 makes f_LS equivalent to 0. Annotations: <i>min=0,max=1,units=per unit</i></p>
<b>gainReactive</b>	<p><b>float32</b> Gain on the reactive power sharing signal. Setting this to 0 makes v_LS equivalent to 0. Annotations: <i>min=0,max=1,units=per unit</i></p>
<b>biasRealPu</b>	<p><b>Curve2D</b> Bias to the real power sharing. If empty, the bias is 0. Annotations: <i>units=per unit watt, per unit watt</i></p>
<b>biasReactivePu</b>	<p><b>Curve2D</b> Bias to the reactive power sharing. If empty, the bias is 0. Annotations: <i>units=per unit volt ampere reactive, per unit volt ampere reactive</i></p>

A3205 Annotations: *min=0,max=1,units=per unit*

A3206

A3207 B.14.2.31 tms.TopicList

A3208 PURPOSE: Specifies a list of communications data topics.

A3209 EXTENSIBILITY: extensibility(APPENDABLE)

A3210 PATTERN: Typedef

A3211 ORIGINAL TYPE: `sequence<TopicName,0,8>`

A3212 B.14.2.32 tms.TopicName

A3213 PURPOSE: Specifies a communications data topic.

A3214 EXTENSIBILITY: extensibility(APPENDABLE)

A3215 PATTERN: Typedef

A3216 ORIGINAL TYPE: `string<1,64>`

A3217 B.14.2.33 tms.dc.LoadSharingRequest

A3218 PURPOSE: Sets load sharing parameters for direct current devices.

A3219 TOPIC USAGE: DcLoadSharingRequest

A3220 EXTENSIBILITY: extensibility(APPENDABLE)

A3221 PATTERN: Structure

A3222 ATTRIBUTES:

Name	Type and Description
<code>requestId</code>	<code>tms::PowerPortConfigRequest</code> Identity of this request. Annotations: <i>keyval</i>
A3223 <code>sequenceId</code>	<code>tms::RequestSequence</code> Request sequence data used to associate a request and returning reply.
<code>parameters</code>	<code>LoadSharing</code> New value to use.

A3225 B.14.2.34 tms.dc.LoadSharing

A3226 PURPOSE: Represents the load sharing parameters for direct current devices.

A3227 TOPIC USAGE: Nested

A3228 EXTENSIBILITY: extensibility(APPENDABLE)

A3229 PATTERN: Structure

A3230 ATTRIBUTES:

Name	Type and Description
<b>minV</b>	<b>float32</b> Minimum output voltage. Saturate if droop would go lower. Annotations: <i>units</i> =Volt
<b>maxV</b>	<b>float32</b> Maximum output voltage. Saturate if droop would go higher. Annotations: <i>units</i> =Volt
<b>minI</b>	<b>float32</b> Minimum output current. Saturate if droop would go lower. Annotations: <i>units</i> =Ampere
<b>maxI</b>	<b>float32</b> Maximum output current. Saturate if droop would go higher. Annotations: <i>units</i> =Ampere
<b>voltageDroop</b>	<b>tms::ControlCurve</b> Voltage to current droop curve. Annotations: <i>units</i> =volt, ampere

A3233 B.14.2.35 tms.stor.StorageControlRequest

A3234 PURPOSE: Set the storage control parameters.

A3235 TOPIC USAGE: StorageControlRequest

A3236 EXTENSIBILITY: extensibility(APPENDABLE)

A3237 PATTERN: Structure

A3238 ATTRIBUTES:

Name	Type and Description
<b>requestId</b>	<b>tms::DeviceConfigRequest</b> Identity of this request. Annotations: <i>keyval</i>
A3239 <b>sequenceId</b>	<b>tms::RequestSequence</b> Request sequence data used to associate a request and returning reply.
<b>parameters</b>	<b>StorageControlParameters</b> New value to use.

A3241 B.14.2.36 tms.stor.StorageControlParameters

A3242 PURPOSE: Parameters for on-platform capabilities of the ESU controller.

A3243 TOPIC USAGE: Nested

A3244 EXTENSIBILITY: extensibility(APPENDABLE)

A3245 PATTERN: Structure

A3246 ATTRIBUTES:

Name	Type and Description
highStateOfChange	<p>float32</p> <p>The high state of charge during continuous operations.</p> <p>Annotations: <i>min=0,max=1,units=per unit</i></p>
lowStateOfChange	<p>float32</p> <p>The low state of charge during continuous operations.</p> <p>Annotations: <i>min=0,max=1,units=per unit</i></p>
nominalChargeDelay	<p>tms::Duration</p> <p>If the ESU has not been required to discharge by more than 1kW for this period of time, then it may automatically initiate a charge cycle to nominalCharge. Similarly, if the ESU has not been required to discharge by more than 1kW for this period of time, then it may automatically initiate a discharge cycle to nominalCharge. Set to +infinity to disable.</p>
nominalChargeRate	<p>float32</p> <p>(<math>\geq 0</math>) If the ESU initiates a charge/discharge cycle to nominalCharge, then the charge/discharge power must stay below this value.</p> <p>Annotations: <i>units=watt</i></p>
rampRate	<p>float32</p> <p>Limit how fast the power output changes for on-platform capabilities that can be planned in advance. These include state of charge maintenance and power curtailment based on state of charge. This limit does not apply to LoadPolicy commands, UPS, overload protection, or motoring protection. Set to NaN for no limit.</p> <p>Annotations: <i>units=watt per second</i></p>
modeSelection	<p>StorageOperatingState</p> <p>Force the ESU to operate in a fixed state, or enable automatic detection.</p>
stopChargeInrush	<p>UnderFrequencyStopCharge</p> <p>Stop charging when there is sudden, severe overload.</p>
stopChargeSteady	<p>UnderFrequencyStopCharge</p> <p>Stop charging when there is persistent, light overload.</p>
stopDischargeInrush	<p>OverFrequencyStopDischarge</p> <p>Stop discharging when there is sudden, severe motoring.</p>
stopDischargeSteady	<p>OverFrequencyStopDischarge</p> <p>Stop discharging when there is persistent, light motoring.</p>

A3247

A3248

A3249 B.14.2.37 tms.Duration

A3250 See definition in Section [B.11.2.11](#).

A3251 B.14.2.38 tms.stor.StorageOperatingState

A3252 PURPOSE: Indicate the ESU operating state.

A3253 EXTENSIBILITY: extensibility(APPENDABLE)

A3254 PATTERN: Enumeration

A3255 VALUES:

	Name	Description
	EOS_UNKNOWN	Uninitialized value.
	EOS_STANDALONE	Standalone operation as only source.
A3256	EOS_PARALLEL	Parallel operation with other sources.
	EOS_AUTOMATIC	Automatic detection of standalone or parallel status.

A3258 B.14.2.39 tms.stor.UnderFrequencyStopCharge

A3259 PURPOSE: Stop charging the ESU if the available generation is overloaded.

A3260 TOPIC USAGE: Nested

A3261 EXTENSIBILITY: extensibility(APPENDABLE)

A3262 PATTERN: Structure

A3263 ATTRIBUTES:

	Name	Type and Description
	frequency	float32 Minimum frequency threshold. Set to NaN to disable this threshold. Annotations: <i>units</i> =hertz
A3264	delay	tms::Duration Time to wait before stopping charge.

A3266 B.14.2.40 tms.stor.OverFrequencyStopDischarge

A3267 PURPOSE: Stop discharging the ESU if the available generation is motoring.

A3268 TOPIC USAGE: Nested

A3269 EXTENSIBILITY: extensibility(APPENDABLE)

A3270 PATTERN: Structure

A3271 ATTRIBUTES:

Name	Type and Description
<b>frequency</b>	<b>float32</b> Maximum frequency threshold. Set to NaN to disable this threshold. Annotations: <i>units</i> =hertz
<b>delay</b>	<b>tms::Duration</b> Time to wait before stopping discharge.

A3274 B.15 Copy Configuration Settings

A3275 Power devices should allow a microgrid controller to copy configuration settings from one ID  
A3276 to another. This copy occurs atomically, with all changes taking effect at the same time, or  
A3277 all changes being rejected as a group.

A3278 B.15.1 Data Flows

A3279 Figure B.19 illustrates all the topics in the data flow.

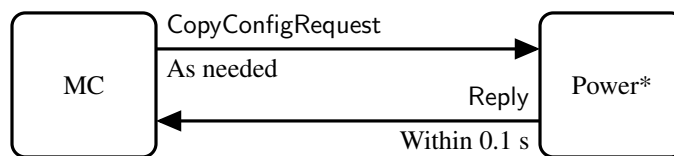


Figure B.19: Copy Configuration Settings data flow

A3280 *Editorial Note: This feature is being considered for removal. It can be approximated by*  
A3281 *reading out the settings and issuing commands individually. Given the granularity of the*  
A3282 *request topics, there has been no demand for this atomic copy.*

A3283

A3284 Table B.51 provides an overview of how each topic is used.



Table B.51: Description of the Copy Configuration Settings topics.

Topic	Description
CopyConfigRequest	Copy the commands from one device configuration index to another.
Reply	The reply to a request.

A3285 Table B.52 specifies the publishers and subscribers for each topic as well as usage  
A3286 requirements as described in Section A.8.7.

Table B.52: Participants of the Copy Configuration Settings topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
CopyConfigRequest	OPTIONAL		Pub						
				Sub	Sub	Sub	Sub	Sub	Sub
Reply	REQUIRED	Pub	Pub		Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A3287 Table B.53 specifies the timing for when sample updates are sent to each topic. These  
A3288 parameters are defined in Section A.8.5.

Table B.53: Timing of the Copy Configuration Settings topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
CopyConfigRequest	As Needed	Class	Size	Command
Reply	Within 0.1 s	All	1	Response

## A3289 B.15.2 Data Types

A3290 Table B.54 specifies the data types that are used as sample values for each topic.

Table B.54: Data types of the Copy Configuration Settings topics.

Topic	Data Type
CopyConfigRequest	tms.CopyConfigRequest
Reply	tms.Reply

A3291 Topic-level data types are defined in the order shown in Table B.54. Then nested data  
A3292 types are defined in the order of appearance. Cross-references are provided for data types  
A3293 that have already been defined.

A3294 B.15.2.1 tms.CopyConfigRequest

A3295 PURPOSE: Copy the commands from one configuration index to another.

A3296 TOPIC USAGE: CopyConfigRequest

A3297 EXTENSIBILITY: extensibility(APPENDABLE)

A3298 PATTERN: Structure

A3299 ATTRIBUTES:

Name	Type and Description
<code>requestId</code>	<code>DeviceConfigRequest</code> Unique request identifier, including the identity of the :MC making the request. Annotations: <i>keyval</i>
<code>sequenceId</code>	<code>tms::RequestSequence</code> Request sequence data used to associate a request and returning reply.
<code>target</code>	<code>ConfigId</code> Configuration index to overwrite with the source values.

A3302 DESCRIPTION:

A3303 This request instructs a device to initialize the `target` configuration with the  
A3304 values contained in the `source` configuration. The receiving device replies with a  
A3305 **Request/Response**, due within 100ms. If either index is invalid, or the requesting device  
A3306 does not have read/write permissions, then the request fails, no change is made to the  
A3307 configurations, and `REPLY_BAD_REQUEST` is returned (rate-10s-burst).

A3308 B.15.2.2 tms.Reply

A3309 See definition in Section [B.2.2.1](#).

A3310 B.15.2.3 tms.DeviceConfigRequest

A3311 See definition in Section [B.14.2.3](#).

A3312 B.15.2.4 tms.Fingerprint

A3313 See definition in Section [B.2.2.2](#).

A3314 B.15.2.5 tms.ConfigId

A3315 See definition in Section [B.2.2.3](#).

A3316 B.15.2.6 tms.RequestSequence

A3317 See definition in Section [B.2.2.5](#).

A3318 B.15.2.7 tms.PowerPortNumber

A3319 See definition in Section [B.2.2.4](#).

A3320 B.15.2.8 tms.ReplyStatus

A3321 See definition in Section [B.2.2.6](#).

A3322 B.15.2.9 tms.String32

A3323 See definition in Section [B.2.2.7](#).

A3324 B.16 Device Parameters

A3325 Device parameters provide a controlled mechanism for vendors to extend this data model  
A3326 with custom values. There are two types of device parameter. Control parameters are  
A3327 read/write values suitable for tuning operational behavior. Metric parameters are read-only  
A3328 values suitable for reporting status.

A3329 B.16.1 Data Flows

A3330 Figure [B.20](#) illustrates all the topics in the data flow.

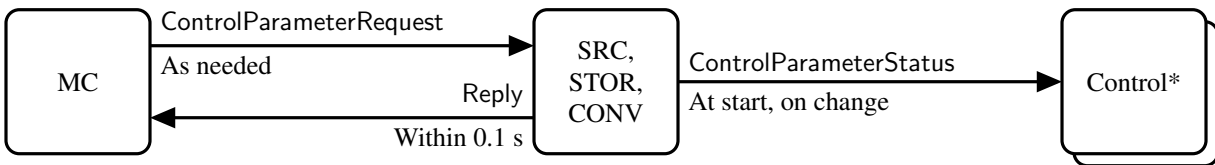


Figure B.20: Device Parameters data flow

A3331 Figure [B.21](#) illustrates all the topics in the data flow.

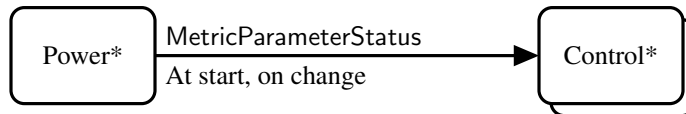


Figure B.21: Device Parameters data flow

A3332 Any power device that supports this mechanism shall declare the list of parameters in  
A3333 the Device Announcements, Section [B.5](#).

A3334

A3335 Table B.55 provides an overview of how each topic is used.

Table B.55: Description of the Device Parameters topics.

Topic	Description
ControlParameterStatus	The active power device control parameters for this power device.
ControlParameterRequest	Remote request to change the device control parameters as defined by the DeviceInfo.controlParameters.
Reply	The reply to a request.
MetricParameterStatus	The active metric parameter values for this power device.

A3336 Table B.56 specifies the publishers and subscribers for each topic as well as usage  
A3337 requirements as described in Section A.8.7.

Table B.56: Participants of the Device Parameters topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
ControlParameterStatus	OPTIONAL				Pub	Pub			Pub
ControlParameterRequest	OPTIONAL	Sub	Sub	Sub					
Reply	REQUIRED	Pub	Pub	Sub	Sub	Sub	Pub	Pub	Sub
MetricParameterStatus	OPTIONAL	Sub	Sub	Sub	Pub	Pub			Pub
		Sub	Sub	Sub					

A3338 Table B.57 specifies the timing for when sample updates are sent to each topic. These  
A3339 parameters are defined in Section A.8.5.

Table B.57: Timing of the Device Parameters topics.

Topic	Data Trigger	Rate Class	Burst Size	QoS Profile
ControlParameterStatus	At Start, On Change	10 s	10	PublishLast
ControlParameterRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response
MetricParameterStatus	At Start, On Change	10 s	10	PublishLast

## A3340 B.16.2 Data Types

A3341 Table B.58 specifies the data types that are used as sample values for each topic.

Table B.58: Data types of the Device Parameters topics.

Topic	Data Type
ControlParameterStatus	tms.ControlParameterStatus
ControlParameterRequest	tms.ControlParameterRequest
Reply	tms.Reply
MetricParameterStatus	tms.MetricParameterStatus

A3342 Topic-level data types are defined in the order shown in Table B.58. Then nested data  
A3343 types are defined in the order of appearance. Cross-references are provided for data types  
A3344 that have already been defined.

#### A3345 B.16.2.1 tms.ControlParameterStatus

A3346 PURPOSE: Status of control parameters.

A3347 TOPIC USAGE: ControlParameterStatus

A3348 EXTENSIBILITY: extensibility(APPENDABLE)

A3349 PATTERN: Structure

A3350 ATTRIBUTES:

Name	Type and Description
deviceId	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
config	<b>ConfigId</b> Configuration that these values belong to.
parameters	<b>ParameterValueSequence</b> A full list of the parameters available on this device.

#### A3353 B.16.2.2 tms.ControlParameterRequest

A3354 See definition in Section B.14.2.17.

#### A3355 B.16.2.3 tms.Reply

A3356 See definition in Section B.2.2.1.

A3357 B.16.2.4 tms.MetricParameterStatus

A3358 PURPOSE: Latest values for device metric parameters.

A3359 TOPIC USAGE: MetricParameterStatus

A3360 EXTENSIBILITY: extensibility(APPENDABLE)

A3361 PATTERN: Structure

A3362 ATTRIBUTES:

Name	Type and Description
<u>deviceId</u>	<p data-bbox="362 640 540 674"><b>Fingerprint</b></p> <p data-bbox="362 678 889 711">The device described by this structure.</p> <p data-bbox="362 716 634 747">Annotations: <i>keyval</i></p>
<u>metricParameters</u>	<p data-bbox="362 793 719 827"><b>ParameterValueSequence</b></p> <p data-bbox="362 831 1105 863">A full list of read-only parameter values on this device.</p>

A3365 B.16.2.5 tms.Fingerprint

A3366 See definition in Section [B.2.2.2](#).

A3367 B.16.2.6 tms.ConfigId

A3368 See definition in Section [B.2.2.3](#).

A3369 B.16.2.7 tms.ParameterValueSequence

A3370 See definition in Section [B.14.2.18](#).

A3371 B.16.2.8 tms.ParameterValue

A3372 See definition in Section [B.14.2.19](#).

A3373 B.16.2.9 tms.String1-32

A3374 See definition in Section [B.5.2.10](#).

A3375 B.16.2.10 tms.DeviceConfigRequest

A3376 See definition in Section [B.14.2.3](#).

A3377 B.16.2.11 tms.RequestSequence

A3378 See definition in Section [B.2.2.5](#).

A3379 B.16.2.12 tms.PowerPortNumber

A3380 See definition in Section [B.2.2.4](#).

A3381 B.16.2.13 tms.ReplyStatus

A3382 See definition in Section [B.2.2.6](#).

A3383 B.16.2.14 tms.String32

A3384 See definition in Section [B.2.2.7](#).

A3385 B.17 Grounding Circuits

A3386 Grounding circuits are used to limit the voltage between power lines and the local earth  
A3387 ground. Most power devices have a single grounding circuit and all parts of the device are  
A3388 bonded to this circuit. Power conversion devices may have different grounding circuits for  
A3389 each power port. Most tactical microgrid equipment is designed to be solidly grounded.  
A3390 Some DC microgrids may be high-resistance grounded.

A3391

A3392 B.17.1 Data Flows

A3393 Figure [B.22](#) illustrates all the topics in the data flow.

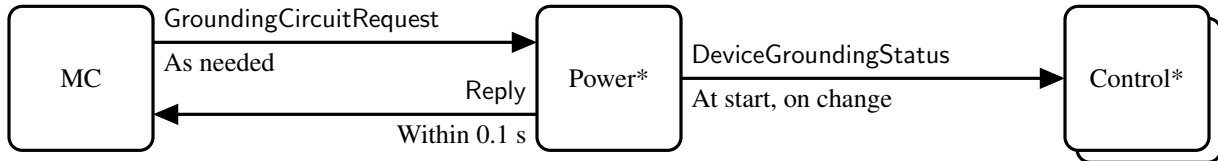


Figure B.22: Grounding Circuits data flow

A3394 All TMS power devices should report the status of their grounding circuits. Any power  
A3395 device having a switchable grounding circuit shall report their circuit status.

A3396

A3397 Table [B.59](#) provides an overview of how each topic is used.

Table B.59: Description of the Grounding Circuits topics.

Topic	Description
DeviceGroundingStatus	Grounding Circuit status.
GroundingCircuitRequest	Grounding Circuit Command.
Reply	The reply to a request.

A3398 Table [B.60](#) specifies the publishers and subscribers for each topic as well as usage  
A3399 requirements as described in Section [A.8.7](#).

Table B.60: Participants of the Grounding Circuits topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DeviceGroundingStatus	CONDITIONAL				Pub	Pub	Pub	Pub	Pub
GroundingCircuitRequest	OPTIONAL	Sub	Sub	Sub					
			Pub						
Reply	REQUIRED	Pub	Pub	Sub	Sub	Sub	Sub	Sub	Sub
		Sub	Sub	Sub	Pub	Pub	Pub	Pub	Pub

A3400 Table B.61 specifies the timing for when sample updates are sent to each topic. These  
A3401 parameters are defined in Section A.8.5.

Table B.61: Timing of the Grounding Circuits topics.

Topic	Data Trigger	Rate Class	Burst Size	QoS Profile
DeviceGroundingStatus	At Start, On Change	10 s	10	PublishLast
GroundingCircuitRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response

## A3402 B.17.2 Data Types

A3403 Table B.62 specifies the data types that are used as sample values for each topic.

Table B.62: Data types of the Grounding Circuits topics.

Topic	Data Type
DeviceGroundingStatus	<code>tms.DeviceGroundingStatus</code>
GroundingCircuitRequest	<code>tms.GroundingCommand</code>
Reply	<code>tms.Reply</code>

A3404 Topic-level data types are defined in the order shown in Table B.62. Then nested data  
A3405 types are defined in the order of appearance. Cross-references are provided for data types  
A3406 that have already been defined.



A3407 B.17.2.1 tms.DeviceGroundingStatus

A3408 PURPOSE: Report the present state of all the ground circuits in a device.

A3409 TOPIC USAGE: DeviceGroundingStatus

A3410 EXTENSIBILITY: extensibility(APPENDABLE)

A3411 PATTERN: Structure

A3412 ATTRIBUTES:

Name	Type and Description
<code>deviceId</code>	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
<code>grounds</code>	<b>GroundingStatusSequence</b> Information on each ground circuit.

A3415 B.17.2.2 tms.GroundingCommand

A3416 See definition in Section [B.14.2.22](#).

A3417 B.17.2.3 tms.Reply

A3418 See definition in Section [B.2.2.1](#).

A3419 B.17.2.4 tms.Fingerprint

A3420 See definition in Section [B.2.2.2](#).

A3421 B.17.2.5 tms.GroundingStatusSequence

A3422 PURPOSE: A sequence of GroundingStatus.

A3423 EXTENSIBILITY: extensibility(FINAL)

A3424 PATTERN: Typedef

A3425 ORIGINAL TYPE: `sequence<GroundingStatus,0,MAX_PORTS>`

A3426 B.17.2.6 tms.GroundingStatus

A3427 PURPOSE: Describe the state of a grounding circuit.

A3428 TOPIC USAGE: Nested

A3429 EXTENSIBILITY: extensibility(APPENDABLE)

A3430 PATTERN: Structure

A3431 ATTRIBUTES:

Name	Type and Description
<b>groundNumber</b>	<b>GroundingCircuitNumber</b> Number used to represent this circuit.
<b>control</b>	optional<PowerSwitchStatus> Status of the grounding control switch.
<b>pulse</b>	optional<PowerSwitchStatus> Status of the grounding pulse switch.
<b>faultDetection</b>	<b>GroundFaultDetection</b> Status of the ground fault detector. GFD_INVALID if no detector.

A3434 B.17.2.7 tms.GroundingCircuitNumber

A3435 See definition in Section [B.5.2.46](#).

A3436 B.17.2.8 tms.PowerSwitchStatus

A3437 PURPOSE: Describe the current state and most recent transition of a power switch

A3438 TOPIC USAGE: Nested

A3439 EXTENSIBILITY: extensibility(APPENDABLE)

A3440 PATTERN: Structure

A3441 ATTRIBUTES:

Name	Type and Description
<b>continuity</b>	<b>CircuitContinuity</b> Indicate circuit continuity through the switch. May be measured from the switch position or from electrical continuity. Continuity is unknown for manual switches with no monitoring capability. Continuity may be inconsistent for multi-phase switches.
<b>transitionFault</b>	<b>boolean</b> Indicate that the switch has failed to respond to attempted transitions and may be stuck in its current state. Not set if a lock or other protection prevents a transition.
<b>transitionLock</b>	<b>boolean</b> Indicate that a protective lock is preventing transitions from the current state.
<b>lastTransition</b>	<b>PowerSwitchReason</b> Type of event that caused the circuit to enter its present state.
<b>lastTransitionActor</b>	<b>Fingerprint</b> Device or user that caused the transition to occur.

A3444 B.17.2.9 tms.CircuitContinuity

A3445 PURPOSE: Indicate whether electricity can flow through a connection.

A3446 EXTENSIBILITY: extensibility(APPENDABLE)

A3447 PATTERN: Enumeration

A3448 VALUES:

Name	Description
CC_UNKNOWN	No information is available, either due to design or malfunction.
CC_OPEN	Electrical or mechanical sensors indicate that continuity has been broken.
A3449 CC_CLOSED	Electrical or mechanical sensors indicate that continuity has been established.
CC_INCONSISTENT	Electrical or mechanical sensors indicate that continuity has been established on some phases and broken on other phases.

A3451 B.17.2.10 tms.PowerSwitchReason

A3452 PURPOSE: Indicate why a power switch entered a continuity state.

A3453 EXTENSIBILITY: extensibility(APPENDABLE)

A3454 PATTERN: Enumeration

A3455 VALUES:

Name	Description
PSR_UNKNOWN	No information is available, either due to design or malfunction.
PSR_MANUAL	Sensors indicate manual operation.
PSR_COMMAND	The switch transitioned due to a (remote) command.
A3456 PSR_PROTECTION	The switch transitioned due to local protection function.
PSR_FAILURE	The transition appears to be due to an internal failure, unrelated to another reason.
PSR_LOAD_POLICY	The switch opened due to a load policy setting. See DeviceLoadPolicyStatus.

A3458 B.17.2.11 tms.GroundFaultDetection

A3459 PURPOSE: Ground fault detection.

A3460 EXTENSIBILITY: extensibility(APPENDABLE)

A3461 PATTERN: Enumeration

A3462 VALUES:

Name	Description
GFD_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
GFD_NONE	No ground fault detected.
GFD_ANY	Ground fault detected. Multiple lines detected or specific line unknown.
A3463 GFD_A	Ground fault detected on phase A.
GFD_B	Ground fault detected on phase B.
GFD_C	Ground fault detected on phase C.
GFD_DCPOS	Ground fault detected on DC+.
GFD_DCNEG	Ground fault detected on DC-.

A3465 B.17.2.12 tms.DeviceConfigRequest

A3466 See definition in Section [B.14.2.3](#).

A3467 B.17.2.13 tms.ConfigId

A3468 See definition in Section [B.2.2.3](#).

A3469 B.17.2.14 tms.RequestSequence

A3470 See definition in Section [B.2.2.5](#).

A3471 B.17.2.15 tms.DesiredCircuitContinuity

A3472 See definition in Section [B.14.2.14](#).

A3473 B.17.2.16 tms.PowerPortNumber

A3474 See definition in Section [B.2.2.4](#).

A3475 B.17.2.17 tms.ReplyStatus

A3476 See definition in Section B.2.2.6.

A3477 B.17.2.18 tms.String32

A3478 See definition in Section B.2.2.7.

A3479 B.18 Remote Start/Stop

A3480 B.18.1 Data Flows

A3481 Figure B.23 illustrates all the topics in the data flow.

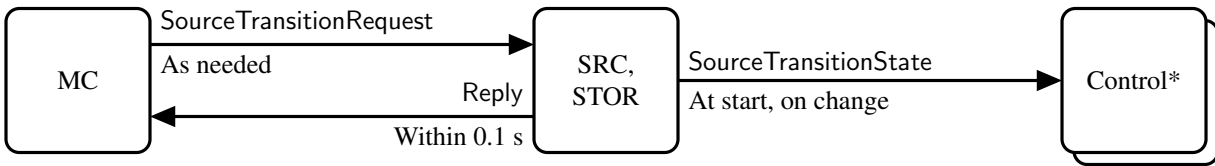


Figure B.23: Remote Start/Stop data flow

A3482 Table B.63 provides an overview of how each topic is used.

Table B.63: Description of the Remote Start/Stop topics.

Topic	Description
SourceTransitionState	Reports the operational state (off, on, running, sourcing) of a Source / Storage device
SourceTransitionRequest	Remote request to change the operational state of a Source / Storage device.
Reply	The reply to a request.

A3483 Table B.64 specifies the publishers and subscribers for each topic as well as usage

A3484 requirements as described in Section A.8.7.

Table B.64: Participants of the Remote Start/Stop topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
SourceTransitionState	REQUIRED				Pub	Pub			
SourceTransitionRequest	REQUIRED	Sub	Sub	Sub					
Reply	REQUIRED	Pub	Pub	Sub	Sub	Sub	Pub	Pub	Pub
		Sub	Sub	Sub					

A3485 Table B.65 specifies the timing for when sample updates are sent to each topic. These  
A3486 parameters are defined in Section A.8.5.

Table B.65: Timing of the Remote Start/Stop topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
		Class	Size	
SourceTransitionState	At Start, On Change	10 s	10	PublishLast
SourceTransitionRequest	As Needed	10 s	1	Command
Reply	Within 0.1 s	All	1	Response

### A3487 B.18.2 Data Types

A3488 Table B.66 specifies the data types that are used as sample values for each topic.

Table B.66: Data types of the Remote Start/Stop topics.

Topic	Data Type
SourceTransitionState	<code>tms.SourceTransitionState</code>
SourceTransitionRequest	<code>tms.SourceTransitionRequest</code>
Reply	<code>tms.Reply</code>

A3489 Topic-level data types are defined in the order shown in Table B.66. Then nested data  
A3490 types are defined in the order of appearance. Cross-references are provided for data types  
A3491 that have already been defined.

A3492 B.18.2.1 tms.SourceTransitionState

A3493 PURPOSE: Report the progress of transitions.

A3494 TOPIC USAGE: SourceTransitionState

A3495 EXTENSIBILITY: extensibility(APPENDABLE)

A3496 PATTERN: Structure

A3497 ATTRIBUTES:

Name	Type and Description
<b>deviceId</b>	<b>Fingerprint</b> The device described by this structure. Annotations: <i>keyval</i>
<b>presentState</b>	<b>SourceState</b> Present location in the state transition model.
<b>futureState</b>	<b>SourceState</b> State that the device is transitioning to, or presentState if no transition.
A3498 <b>elapsedTime</b>	<b>float32</b> How long the device has been in the present state or transition. Annotations: <i>units=second</i>
<b>remainingTime</b>	<b>float32</b> Estimates when the present transition will complete. It is zero (0.0) when the device is not in transition. Annotations: <i>units=second</i>
<b>relatedRequestId</b>	<b>DeviceConfigRequest</b> Copy of the most recent SourceTransitionRequest.requestId.

A3500 B.18.2.2 tms.SourceTransitionRequest

A3501 See definition in Section [B.14.2.15](#).

A3502 B.18.2.3 tms.Reply

A3503 See definition in Section [B.2.2.1](#).

A3504 B.18.2.4 tms.Fingerprint

A3505 See definition in Section [B.2.2.2](#).



A3506 B.18.2.5 tms.SourceState

A3507 PURPOSE: Indicate a state in the source state transition model.

A3508 EXTENSIBILITY: extensibility(APPENDABLE)

A3509 PATTERN: Enumeration

A3510 VALUES:

Name	Description
SS_UNKNOWN	Uninitialized value.
SS_OFF	OFF state.
A3511 SS_ON	ON state.
SS_RUNNING	RUNNING state.
SS_SOURCING	SOURCING state.

A3513 B.18.2.6 tms.DeviceConfigRequest

A3514 See definition in Section [B.14.2.3](#).

A3515 B.18.2.7 tms.ConfigId

A3516 See definition in Section [B.2.2.3](#).

A3517 B.18.2.8 tms.RequestSequence

A3518 See definition in Section [B.2.2.5](#).

A3519 B.18.2.9 tms.SourceTransition

A3520 See definition in Section [B.14.2.16](#).

A3521 B.18.2.10 tms.PowerPortNumber

A3522 See definition in Section [B.2.2.4](#).

A3523 B.18.2.11 tms.ReplyStatus

A3524 See definition in Section [B.2.2.6](#).

A3525 B.18.2.12 tms.String32

A3526 See definition in Section [B.2.2.7](#).

A3527 B.19 Power Switches and Connectors

A3528 All power devices shall report the status of their power switches and connectors. All power  
A3529 devices shall inventory their power switches and connectors in the Device Announcements,  
A3530 Section B.5.

A3531

A3532 *Editorial note: Considering splitting switches and connectors into separate flows.*

A3533

A3534 B.19.1 Data Flows

A3535 Figure B.24 illustrates all the topics in the data flow.

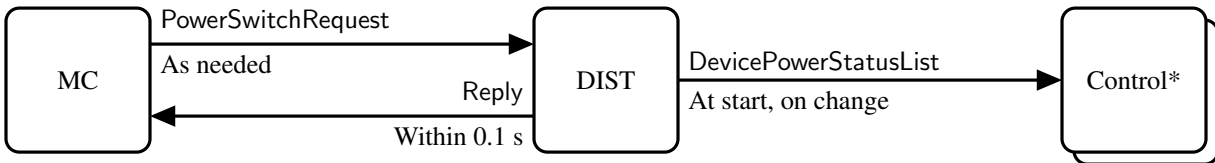


Figure B.24: Power Switches and Connectors data flow

A3536 Table B.67 provides an overview of how each topic is used.

Table B.67: Description of the Power Switches and Connectors topics.

Topic	Description
DevicePowerStatusList	Reports dynamic state changes for a power device ports.
PowerSwitchRequest	Remote request to change the circuit continuity of a single power port
Reply	The reply to a request.

A3537 Table B.68 specifies the publishers and subscribers for each topic as well as usage  
A3538 requirements as described in Section A.8.7.

Table B.68: Participants of the Power Switches and Connectors topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DevicePowerStatusList	REQUIRED				Pub	Pub	Pub	Pub	Pub
PowerSwitchRequest	CONDITIONAL	Sub	Sub	Sub					
Reply	REQUIRED	Pub	Pub	Sub	Pub	Pub	Sub	Pub	Sub
		Sub	Sub	Sub					

A3539 Any device that has remotely switchable ports shall implement the PowerSwitchRequest.

A3540 Table B.69 specifies the timing for when sample updates are sent to each topic. These  
A3541 parameters are defined in Section A.8.5.

Table B.69: Timing of the Power Switches and Connectors topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
		Class	Size	
DevicePowerStatusList	At Start, On Change	1 s	1	PublishLast
PowerSwitchRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response

## A3542 B.19.2 Data Types

A3543 Table B.70 specifies the data types that are used as sample values for each topic.

Table B.70: Data types of the Power Switches and Connectors topics.

Topic	Data Type
DevicePowerStatusList	tms.DevicePowerPortStatuses
PowerSwitchRequest	tms.PowerSwitchCommand
Reply	tms.Reply

A3544 Topic-level data types are defined in the order shown in Table B.70. Then nested data  
A3545 types are defined in the order of appearance. Cross-references are provided for data types  
A3546 that have already been defined.

### A3547 B.19.2.1 tms.DevicePowerPortStatuses

A3548 PURPOSE: Report the present state of all the power ports in a device.

A3549 TOPIC USAGE: DevicePowerStatusList

A3550 EXTENSIBILITY: extensibility(APPENDABLE)

A3551 PATTERN: Structure

A3552 ATTRIBUTES:

Name	Type and Description
deviceId	
	<b>Fingerprint</b>
	The device described by this structure.
	Annotations: <i>keyval</i>
powerPorts	
	<b>PowerPortStatusSequence</b>
	Information on each power port.

A3555 B.19.2.2 tms.PowerSwitchCommand

A3556 See definition in Section [B.14.2.11](#).

A3557 B.19.2.3 tms.Reply

A3558 See definition in Section [B.2.2.1](#).

A3559 B.19.2.4 tms.Fingerprint

A3560 See definition in Section [B.2.2.2](#).

A3561 B.19.2.5 tms.PowerPortStatusSequence

A3562 PURPOSE: A sequence of PowerPortStatus.

A3563 EXTENSIBILITY: extensibility(FINAL)

A3564 PATTERN: Typedef

A3565 ORIGINAL TYPE: `sequence<PowerPortStatus,0,MAX_PORTS>`

A3566 B.19.2.6 tms.PowerPortStatus

A3567 PURPOSE: Describe the state of a power port.

A3568 TOPIC USAGE: Nested

A3569 EXTENSIBILITY: extensibility(APPENDABLE)

A3570 PATTERN: Structure

A3571 ATTRIBUTES:

Name	Type and Description
------	----------------------

---

portNumber	
------------	--

PowerPortNumber	
-----------------	--

Number shown on the device exterior to represent this port.	
---	--

cableStatus	
-------------	--

CableSenseStatus	
------------------	--

Indicates whether a power cable is plugged in.	
--	--

cableId	
---------	--

optional<Fingerprint>	
-----------------------	--

Indicates that a smart cable is attached. This value is set when a smart cable is connected and cleared when it is disconnected.	
--	--

connection	
------------	--

optional<PowerPortId>	
-----------------------	--

Indicates that this port is connected to a remote port.	
---	--

switchStatus	
--------------	--

optional<PowerSwitchStatus>	
-----------------------------	--

Indicates the state of a switch associated with this port. Must be empty when PowerPortType.switch is NONE; otherwise it must be full.	
--	--

---

A3573

A3574 B.19.2.7 tms.PowerPortNumber

A3575 See definition in Section [B.2.2.4](#).

A3576 B.19.2.8 tms.CableSenseStatus

A3577 PURPOSE: Indicate whether a cable has been sensed in a power port.

A3578 EXTENSIBILITY: extensibility(APPENDABLE)

A3579 PATTERN: Enumeration

A3580 VALUES:

Name	Description
------	-------------

---

CS_UNKNOWN	
------------	--

	Cable presence is neither confirmed nor denied through active detection. Cable sense is not available, either due to device design or failure.
--	--

A3581 CS\_DISCONNECTED

	Sensing indicates that no cable is not connected. CST_MEASUREMENT devices must use CS_UNKNOWN instead of this value.
--	--

CS\_CONNECTED

	Sensing indicates that a cable is connected.
--	--

---

A3582

A3583 B.19.2.9 tms.PowerPortId

A3584 PURPOSE: Identify a power port (often remote)

A3585 TOPIC USAGE: Nested

A3586 EXTENSIBILITY: extensibility(APPENDABLE)

A3587 PATTERN: Structure

A3588 ATTRIBUTES:

Name	Type and Description
deviceId	Fingerprint
	Device containing the port.
portNumber	PowerPortNumber
	Number shown on the device exterior to represent this port.

A3591 B.19.2.10 tms.PowerSwitchStatus

A3592 See definition in Section [B.17.2.8](#).

A3593 B.19.2.11 tms.CircuitContinuity

A3594 See definition in Section [B.17.2.9](#).

A3595 B.19.2.12 tms.PowerSwitchReason

A3596 See definition in Section [B.17.2.10](#).

A3597 B.19.2.13 tms.PowerPortConfigRequest

A3598 See definition in Section [B.14.2.12](#).

A3599 B.19.2.14 tms.ConfigId

A3600 See definition in Section [B.2.2.3](#).

A3601 B.19.2.15 tms.RequestSequence

A3602 See definition in Section [B.2.2.5](#).

A3603 B.19.2.16 tms.DesiredCircuitContinuity

A3604 See definition in Section [B.14.2.14](#).

A3605 B.19.2.17 tms.ReplyStatus

A3606 See definition in Section [B.2.2.6](#).

A3607 B.19.2.18 tms.String32

A3608 See definition in Section [B.2.2.7](#).

A3609 B.20 AC Measurements

A3610 This topic provides power monitoring of the alternating current power device ports. Power  
A3611 monitoring is used to balance sources, avoid exceeding thermal constraints or tripping  
A3612 protection devices, and detect power-specific issues such as grounding. It is not used to  
A3613 measure power quality or perform synchronization.

A3614 B.20.1 Data Flows

A3615 For instantaneous measurements, the conventional current definition is used with positive  
A3616 current flowing from positive to negative terminals. For RMS measurements, Kirkchoff's  
A3617 convention is used. Positive RMS current indicates power flowing from the grid into a  
A3618 device. Negative RMS current indicates power flowing from a device into the grid.

A3619 Figure [B.25](#) illustrates all the topics in the data flow.

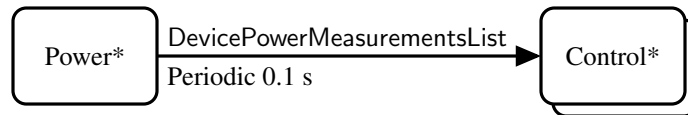


Figure B.25: AC Measurements data flow

A3620 Sources usually report negative current and loads usually report positive current. The  
A3621 sum of the currents reported by distribution devices should sum to zero, after accounting  
A3622 for internal losses.

A3623 Table [B.71](#) provides an overview of how each topic is used.

Table B.71: Description of the AC Measurements topics.

Topic	Description
DevicePowerMeasurementList	Reports dynamic power measurements for alternating current power device ports.

A3624 Table [B.72](#) specifies the publishers and subscribers for each topic as well as usage  
A3625 requirements as described in Section [A.8.7](#).



Table B.72: Participants of the AC Measurements topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DevicePowerMeasurementList	CONDITIONAL	Sub	Sub	Sub	Pub	Pub	Pub	Pub	Pub

A3626 Table B.73 specifies the timing for when sample updates are sent to each topic. These  
A3627 parameters are defined in Section A.8.5.

Table B.73: Timing of the AC Measurements topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
		Class	Size	
DevicePowerMeasurementList	Periodic 0.1 s	0.1 s	1	Continuous

## A3628 B.20.2 Data Types

A3629 Table B.74 specifies the data types that are used as sample values for each topic.

Table B.74: Data types of the AC Measurements topics.

Topic	Data Type
DevicePowerMeasurementList	<code>tms.ac.DevicePowerMeasurements</code>

A3630 Topic-level data types are defined in the order shown in Table B.74. Then nested data  
A3631 types are defined in the order of appearance. Cross-references are provided for data types  
A3632 that have already been defined.

A3633 B.20.2.1 tms.ac.DevicePowerMeasurements

A3634 PURPOSE: Report power monitoring for an alternating current device

A3635 TOPIC USAGE: DevicePowerMeasurementList

A3636 EXTENSIBILITY: extensibility(APPENDABLE)

A3637 PATTERN: Structure

A3638 ATTRIBUTES:

Name	Type and Description
<code>deviceId</code>	<code>tms::Fingerprint</code> The device described by this structure. Annotations: <i>keyval</i>
A3639 <code>timeMeasured</code>	<code>tms::ClockMonotonic</code> Time of these measurements.
<code>powerPorts</code>	<code>PowerPortMeasurementSequence</code> Measurements for all power ports on this device.

A3641 B.20.2.2 tms.Fingerprint

A3642 See definition in Section [B.2.2.2](#).

A3643 B.20.2.3 tms.ClockMonotonic

A3644 See definition in Section [B.8.2.3](#).

A3645 B.20.2.4 tms.ac.PowerPortMeasurementSequence

A3646 PURPOSE: A sequence of `ac::PowerPortMeasurement`.

A3647 EXTENSIBILITY: extensibility(FINAL)

A3648 PATTERN: Typedef

A3649 ORIGINAL TYPE: `sequence<PowerPortMeasurement,0,tms::MAX_PORTS>`

A3650 B.20.2.5 tms.ac.PowerPortMeasurement

A3651 PURPOSE: Power measurement for an alternating current port.

A3652 TOPIC USAGE: Nested

A3653 EXTENSIBILITY: extensibility(APPENDABLE)

A3654 PATTERN: Structure

A3655 ATTRIBUTES:

Name	Type and Description
<b>portNumber</b>	<b>tms::PowerPortNumber</b> Port being measured.
<b>measurement</b>	<b>PowerMeasurement</b> Power measurement for this port.

A3658 B.20.2.6 tms.PowerPortNumber

A3659 See definition in Section [B.2.2.4](#).

A3660 B.20.2.7 tms.ac.PowerMeasurement

A3661 PURPOSE: All measurements for an AC system.

A3662 TOPIC USAGE: Nested

A3663 EXTENSIBILITY: extensibility(APPENDABLE)

A3664 PATTERN: Structure

A3665 ATTRIBUTES:

Name	Type and Description
<b>phases</b>	<b>PowerMeasurementLineSequence</b> Measured values for all applicable AC phases. The actual length depends on the power ports <code>PowerConnectorPhases</code> . <code>PHASE_SINGLE=1</code> , <code>PHASE_SPLIT=2</code> , <code>PHASE_3WYE=4</code> , and <code>PHASE_3DELTA=3</code> .
<b>phaseAngles</b>	<b>PhaseAnglesSequence</b> Measured phase difference between the voltages for all applicable phase pairs. The actual length depends on the power ports <code>PowerConnectorPhases</code> . Annotations: <i>units</i> =degrees

A3668 For each power port, the order of values in **phases** and **phaseAngle** must match the  
A3669 corresponding value reported in `tms.PowerPortInfo.phases`. These mappings are given in

A3670 Table B.75, Table B.76, Table B.77, and Table B.78. The  $V_{NG}$  measurements are referenced  
A3671 to the protective chassis ground.

A3672 Values at the end of each sequence may be omitted when the device is not constructed to  
A3673 measure them. The number of values shall be constant and not change from sample to  
A3674 sample. Any entries that are “skipped” shall be filled with NaN, so that later values in the  
A3675 sequence are not mis-interpreted.

Table B.75: Voltage, current, and angle measurements for PHASE\_SINGLE.

Voltage	Current	Angle
$V$	$I$	N/A

Table B.76: Voltage, current, and angle measurements for PHASE\_SPLIT.

Voltage	Current	Angle
$V_{1N}$	$I_A$	$\angle 12$
$V_{2N}$	$I_B$	
$V_{NG}$	$I_N$	

Table B.77: Voltage, current, and angle measurements for PHASE\_3DELTA.

Voltage	Current	Angle
$V_{AB}$	$I_A$	$\angle AB$
$V_{BC}$	$I_B$	$\angle BC$
$V_{CA}$	$I_C$	$\angle CA$

Table B.78: Voltage, current, and angle measurements for PHASE\_3WYE.

Voltage	Current	Angle
$V_{AN}$	$I_A$	$\angle AB$
$V_{BN}$	$I_B$	$\angle BC$
$V_{CN}$	$I_C$	$\angle CA$
$V_{NG}$	$I_N$	

A3676 Figure B.26 illustrates the phase angles for a 208/120V wye system.

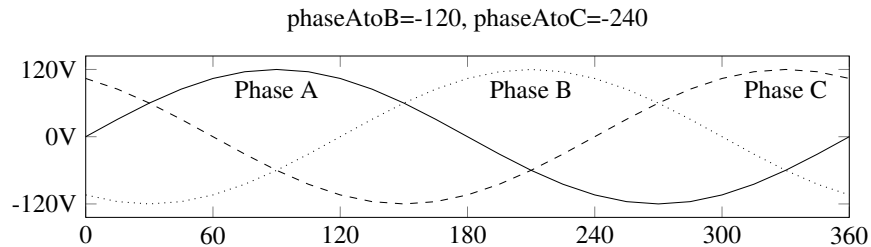


Figure B.26: Phase Angles for a 208/120V Wye System

A3677 B.20.2.8 tms.ac.PowerMeasurementLineSequence

A3678 PURPOSE: A sequence of PowerMeasurementLine.

A3679 EXTENSIBILITY: extensibility(FINAL)

A3680 PATTERN: Typedef

A3681 ORIGINAL TYPE: `sequence<PowerMeasurementLine,1,4>`

A3682 B.20.2.9 tms.ac.PowerMeasurementLine

A3683 PURPOSE: Measurements for a single line.

A3684 TOPIC USAGE: Nested

A3685 EXTENSIBILITY: extensibility(APPENDABLE)

A3686 PATTERN: Structure

A3687 ATTRIBUTES:

Name	Type and Description
v	<p><code>float32</code> RMS voltage. Annotations: <i>units</i>=Volt</p>
i	<p><code>float32</code> RMS current. Annotations: <i>units</i>=Ampere</p>
phi	<p><code>float32</code> Power factor angle (phase offset) between the voltage and current. Annotations: <i>min</i>=-180,<i>max</i>=180,<i>units</i>=degrees</p>
p	<p><code>float32</code> Active RMS power. Annotations: <i>units</i>=watt</p>
q	<p><code>float32</code> Reactive RMS power. Annotations: <i>units</i>=volt ampere reactive</p>
f	<p><code>float32</code> Dominant line frequency. Annotations: <i>min</i>=0,<i>units</i>=hertz</p>

A3690 These measurements represent a single line (power or neutral) in the system reported at a

A3691 constant 10 Hz rate. Unless otherwise noted, all of these measurements should be averaged  
A3692 over 100 ms so that each report contains strictly new information with no coverage gaps  
A3693 between reports.

A3694 Separate values for each line enable the detection of load imbalances, poor connections,  
A3695 and other power-specific issues. Measurements of the neutral line may be compared across  
A3696 different devices to detect certain grounding issues.

A3697 In a closed circuit, several of these measurements can be calculated from each other. These  
A3698 redundant measurements are included to improve performance as circuits change and to  
A3699 detect faults common in distributed systems.

A3700 Under normal synchronous operation, the frequency on all lines will be common throughout  
A3701 the grid. A drifting in frequencies from one location to another may indicate grid partitioning.  
A3702 This drift may help identify connected grid segments. This measurement can also be used  
A3703 to bring segments close enough together that a synchronizer can join them. A divergence  
A3704 in frequencies between the phases at a single location may indicated inverter regulation  
A3705 problems or another fault.

A3706 Values that are not measured, either due to design or malfunction, are filled with NaN.  
A3707 Default, nominal, previous, and projected values must not be used.

A3708 The figures below illustrate lead and lag power factor angles.

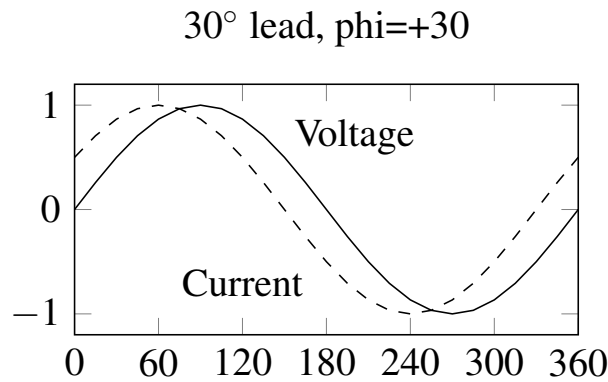


Figure B.27: Power Factor Lead Angle

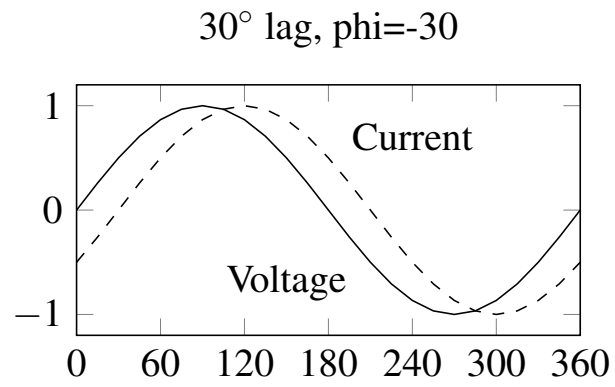


Figure B.28: Power Factor Lag Angle

A3709 B.20.2.10 tms.ac.PhaseAnglesSequence

A3710 PURPOSE: A sequence of floats representing phase angles.

A3711 EXTENSIBILITY: extensibility(FINAL)

A3712 PATTERN: Typedef

A3713 ORIGINAL TYPE: `sequence<float32,0,3>`

A3714 B.21 DC Measurements

A3715 This topic provides power monitoring of the direct current power device ports. Power  
A3716 monitoring is used to balance sources, avoid exceeding thermal constraints or tripping  
A3717 protection devices, and detect power-specific issues such as grounding. It is not used to  
A3718 measure power quality or perform synchronization.

A3719 B.21.1 Data Flows

A3720 Figure B.29 illustrates all the topics in the data flow.

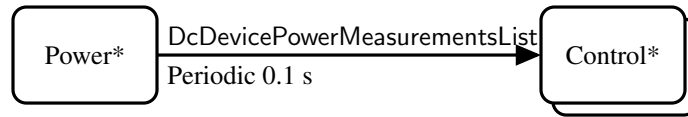


Figure B.29: DC Measurements data flow

A3721 Table B.79 provides an overview of how each topic is used.

Table B.79: Description of the DC Measurements topics.

Topic	Description
DcDevicePowerMeasurementList	Reports dynamic power measurements for a direct current power device ports.

A3722 Table B.80 specifies the publishers and subscribers for each topic as well as usage  
A3723 requirements as described in Section A.8.7.

Table B.80: Participants of the DC Measurements topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DcDevicePowerMeasurementList	CONDITIONAL				Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A3724 Table B.81 specifies the timing for when sample updates are sent to each topic. These  
A3725 parameters are defined in Section A.8.5.



Table B.81: Timing of the DC Measurements topics.

Topic	Data Trigger	Rate	Burst	Class	Size	QoS Profile
DcDevicePowerMeasurementList	Periodic	0.1 s	0.1 s	1	1	Continuous

A3726 B.21.2 Data Types

A3727 Table B.82 specifies the data types that are used as sample values for each topic.

Table B.82: Data types of the DC Measurements topics.

Topic	Data Type
DcDevicePowerMeasurementList	tms.dc.DevicePowerMeasurements

A3728 Topic-level data types are defined in the order shown in Table B.82. Then nested data  
A3729 types are defined in the order of appearance. Cross-references are provided for data types  
A3730 that have already been defined.

A3731 B.21.2.1 tms.dc.DevicePowerMeasurements

A3732 PURPOSE: Report power monitoring for a direct current device

A3733 TOPIC USAGE: DcDevicePowerMeasurementList

A3734 EXTENSIBILITY: extensibility(APPENDABLE)

A3735 PATTERN: Structure

A3736 ATTRIBUTES:

Name	Type and Description
deviceId	
	tms::Fingerprint
	The device described by this structure.
	Annotations: <i>keyval</i>
timeMeasured	
	tms::ClockMonotonic
A3737	Time of these measurements.
powerPorts	
	PowerPortMeasurementSequence
	Measurements for all ports on this device.
balanceResistorEnabled	
	tms::CircuitContinuity
A3738	Measurements for the DC balance resistor.

A3739 B.21.2.2 tms.Fingerprint

A3740 See definition in Section B.2.2.2.

A3741 B.21.2.3 tms.ClockMonotonic

A3742 See definition in Section [B.8.2.3](#).

A3743 B.21.2.4 tms.dc.PowerPortMeasurementSequence

A3744 PURPOSE: A sequence of dc:PowerPortMeasurement.

A3745 EXTENSIBILITY: extensibility(FINAL)

A3746 PATTERN: Typedef

A3747 ORIGINAL TYPE: `sequence<PowerPortMeasurement,0,tms::MAX_PORTS>`

A3748 B.21.2.5 tms.dc.PowerPortMeasurement

A3749 PURPOSE: Power measurement for a direct current power port.

A3750 TOPIC USAGE: Nested

A3751 EXTENSIBILITY: extensibility(APPENDABLE)

A3752 PATTERN: Structure

A3753 ATTRIBUTES:

Name	Type and Description
<code>portNumber</code>	<code>tms::PowerPortNumber</code> Port being measured.
<code>measurement</code>	<code>PowerMeasurement</code> Power measurement for this power port.

A3756 B.21.2.6 tms.PowerPortNumber

A3757 See definition in Section [B.2.2.4](#).

A3758 B.21.2.7 tms.dc.PowerMeasurement

A3759 PURPOSE: Power measurement for direct current devices.

A3760 TOPIC USAGE: Nested

A3761 EXTENSIBILITY: extensibility(APPENDABLE)

A3762 PATTERN: Structure

A3763 ATTRIBUTES:

Name	Type and Description
<b>power</b>	<b>float32</b> Average power. Annotations: <i>units</i> =watt
<b>v</b>	<b>float32</b> Voltage from DC+ to DC-. Annotations: <i>units</i> =Volt
<b>i</b>	<b>float32</b> Current from DC+ to DC-. Annotations: <i>units</i> =Ampere
<b>vGround</b>	<b>float32</b> Voltage from DC- to ground. Annotations: <i>units</i> =Volt

A3766 B.21.2.8 tms.CircuitContinuity

A3767 See definition in Section [B.17.2.9](#).

A3768 B.22 Power Topology

A3769 Power topology describes how power devices are connected to create a microgrid. Topology  
 A3770 may be reported by the operator, a power device, or the microgrid controller.

A3771 B.22.1 Data Flows

A3772 Figure [B.30](#) illustrates all the topics in the data flow.

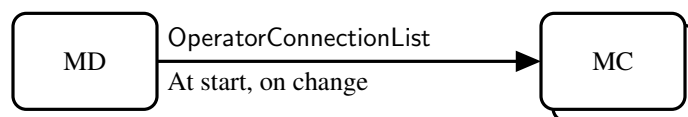


Figure B.30: Power Topology data flow

A3773 Figure B.31 illustrates all the topics in the data flow.

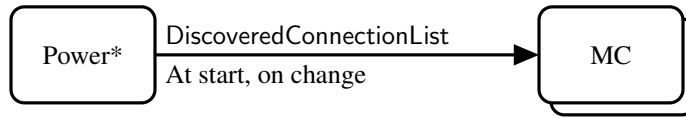


Figure B.31: Power Topology data flow

A3774 Figure B.32 illustrates all the topics in the data flow.

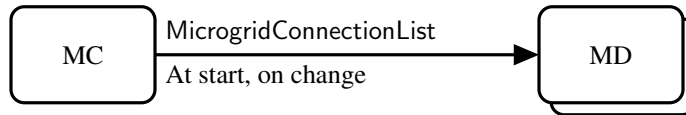


Figure B.32: Power Topology data flow

A3775 Table B.83 provides an overview of how each topic is used.

Table B.83: Description of the Power Topology topics.

Topic	Description
OperatorConnectionList	Power connections manually entered by a microgrid operator.
DiscoveredConnectionList	Power connections discovered by a power device, usually through a smart cable or operator entry.
MicrogridConnectionList	Power connections used by the microgrid controller. May include operator connections, discovered connections, and other connections inferred by the MC.

A3776 Table B.84 specifies the publishers and subscribers for each topic as well as usage  
A3777 requirements as described in Section A.8.7.

Table B.84: Participants of the Power Topology topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
OperatorConnectionList	OPTIONAL	Pub							
DiscoveredConnectionList	OPTIONAL		Sub	Sub					
MicrogridConnectionList	OPTIONAL		Sub	Sub					
			Pub						
		Sub		Sub					

A3778 Table B.85 specifies the timing for when sample updates are sent to each topic. These  
A3779 parameters are defined in Section A.8.5.

Table B.85: Timing of the Power Topology topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
OperatorConnectionList	At Start, On Change	10 s	1	PublishLast
DiscoveredConnectionList	At Start, On Change	10 s	1	PublishLast
MicrogridConnectionList	At Start, On Change	10 s	1	PublishLast

A3780 B.22.2 Data Types

A3781 Table B.86 specifies the data types that are used as sample values for each topic.

Table B.86: Data types of the Power Topology topics.

Topic	Data Type
OperatorConnectionList	tms.PowerConnectionList
DiscoveredConnectionList	tms.PowerConnectionList
MicrogridConnectionList	tms.PowerConnectionList

A3782 Topic-level data types are defined in the order shown in Table B.86. Then nested data  
A3783 types are defined in the order of appearance. Cross-references are provided for data types  
A3784 that have already been defined.

A3785 B.22.2.1 tms.PowerConnectionList

A3786 PURPOSE: List all power connections known to a device or user

A3787 TOPIC USAGE: OperatorConnectionList,DiscoveredConnectionList,MicrogridConnectionList

A3788 EXTENSIBILITY: extensibility(APPENDABLE)

A3789 PATTERN: Structure

A3790 ATTRIBUTES:

Name	Type and Description
deviceId	
	Fingerprint
	The reporting device or user.
	Annotations: <i>keyval</i>
connections	
	PowerConnectionSequence
	List of power connections being reported by this device or user.

A3793 B.22.2.2 tms.Fingerprint

A3794 See definition in Section B.2.2.2.

A3795 B.22.2.3 tms.PowerConnectionSequence

A3796 PURPOSE: A sequence of PowerConnection.

A3797 EXTENSIBILITY: extensibility(FINAL)

A3798 PATTERN: Typedef

A3799 ORIGINAL TYPE: `sequence<PowerConnection,0,200>`

A3800 B.22.2.4 tms.PowerConnection

A3801 PURPOSE: Identify links between devices

A3802 TOPIC USAGE: Nested

A3803 EXTENSIBILITY: extensibility(APPENDABLE)

A3804 PATTERN: Structure

A3805 ATTRIBUTES:

Name	Type and Description
<code>connectionId</code>	<code>PowerConnectionId</code> Connection described by this object.
<code>detectionType</code>	<code>PowerConnectionDetectionType</code> Technique used to detect this connection.
<code>detectionSource</code>	<code>Fingerprint</code> Device or user that detected this connection.
<code>detectionConfidence</code>	<code>float32</code> Confidence in the link detection. -1 = confident the link does not exist, 0 = no opinion, 1 = confident the link exists. Annotations: <i>min=-1,max=1</i>

A3808 DESCRIPTION:

A3809 All devices should monitor local connections for a change in state. Loss active monitoring  
A3810 may trigger a transition to PCDT\_HISTORY. This loss may occur due to device restart,  
A3811 cable sense indicating no cable (CS\_NONE), or loss of line voltage when cable sense is not  
A3812 available.

A3813 B.22.2.5 tms.PowerConnectionId

A3814 PURPOSE: Identify a connection between power ports

A3815 TOPIC USAGE: Nested

A3816 EXTENSIBILITY: extensibility(APPENDABLE)

A3817 PATTERN: Structure

A3818 ATTRIBUTES:

Name	Type and Description
portA	PowerPortId One end of the connection.
portB	PowerPortId The other end of the connection.

A3821 B.22.2.6 tms.PowerPortId

A3822 See definition in Section [B.19.2.9](#).

A3823 B.22.2.7 tms.PowerPortNumber

A3824 See definition in Section [B.2.2.4](#).

A3825 B.22.2.8 tms.PowerConnectionDetectionType

A3826 PURPOSE: Indicate how a power connection was detected.

A3827 EXTENSIBILITY: extensibility(APPENDABLE)

A3828 PATTERN: Enumeration

A3829 VALUES:

Name	Description
PCDT_UNKNOWN	Unknown value. This is a generic placeholder representing the case in which an appropriate value could not be determined.
PCDT_OPERATOR	Operator entry.
PCDT_CABLE_ID	Identification embedded in a smart cable.
A3830 PCDT_PROBE	Active probe signal sent down the line.
PCDT_CORRELATION	Correlation of power measurements was used.
PCDT_HISTORY	Device reset or other event has lost continuous monitoring of the connection. This connection was detected before the event and has neither been confirmed nor denied since.

A3832 B.23 AC Load Sharing

A3833 B.23.1 Data Flows

A3834 Figure B.33 illustrates all the topics in the data flow.

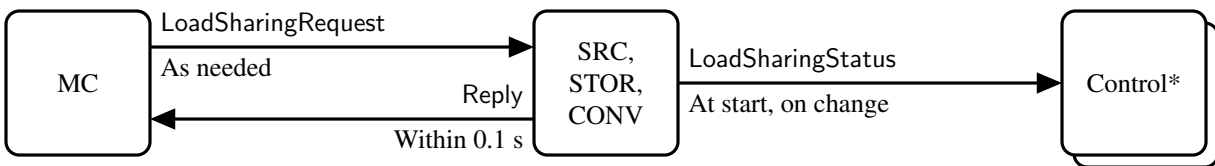


Figure B.33: AC Load Sharing data flow

A3835 Table B.87 provides an overview of how each topic is used.



Table B.87: Description of the AC Load Sharing topics.

Topic	Description
LoadSharingStatus	Reports the active load sharing status of an alternating current Source / Storage device.
LoadSharingRequest	Remote request to change the load sharing status of a Source / Storage device.
Reply	The reply to a request.

A3836 Table B.88 specifies the publishers and subscribers for each topic as well as usage  
A3837 requirements as described in Section A.8.7.

Table B.88: Participants of the AC Load Sharing topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
LoadSharingStatus	CONDITIONAL				Pub	Pub			Pub
LoadSharingRequest	OPTIONAL	Sub	Sub	Sub					
			Pub						
				Sub	Sub	Sub			Sub
Reply	REQUIRED	Pub	Pub		Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A3838 Table B.89 specifies the timing for when sample updates are sent to each topic. These  
A3839 parameters are defined in Section A.8.5.

Table B.89: Timing of the AC Load Sharing topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
LoadSharingStatus	At Start, On Change	10 s	10	PublishLast
LoadSharingRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response

## A3840 B.23.2 Data Types

A3841 Table B.90 specifies the data types that are used as sample values for each topic.

Table B.90: Data types of the AC Load Sharing topics.

Topic	Data Type
LoadSharingStatus	tms.ac.LoadSharingStatus
LoadSharingRequest	tms.ac.LoadSharingRequest
Reply	tms.Reply

A3842 Topic-level data types are defined in the order shown in Table [B.90](#). Then nested data  
A3843 types are defined in the order of appearance. Cross-references are provided for data types  
A3844 that have already been defined.

#### A3845 B.23.2.1 tms.ac.LoadSharingStatus

A3846 PURPOSE: Report the present value of the load sharing parameters for alternating current  
A3847 devices.

A3848 TOPIC USAGE: LoadSharingStatus

A3849 EXTENSIBILITY: extensibility(APPENDABLE)

A3850 PATTERN: Structure

A3851 ATTRIBUTES:

Name	Type and Description
<u>deviceId</u>	<p data-bbox="362 848 621 886"><b>tms::Fingerprint</b></p> <p data-bbox="362 886 889 924">The device described by this structure.</p> <p data-bbox="362 924 634 961">Annotations: <i>keyval</i></p>
A3852 <u>config</u>	<p data-bbox="362 1005 574 1043"><b>tms::ConfigId</b></p> <p data-bbox="362 1043 932 1079">Configuration that these values belong to.</p>
<u>parameters</u>	<p data-bbox="362 1123 542 1161"><b>LoadSharing</b></p> <p data-bbox="362 1161 602 1194">Parameter values.</p>

#### A3854 B.23.2.2 tms.ac.LoadSharingRequest

A3855 See definition in Section [B.14.2.24](#).

#### A3856 B.23.2.3 tms.Reply

A3857 See definition in Section [B.2.2.1](#).

#### A3858 B.23.2.4 tms.Fingerprint

A3859 See definition in Section [B.2.2.2](#).

#### A3860 B.23.2.5 tms.ConfigId

A3861 See definition in Section [B.2.2.3](#).

#### A3862 B.23.2.6 tms.ac.LoadSharing

A3863 See definition in Section [B.14.2.25](#).

A3864 B.23.2.7 tms.ControlCurve

A3865 See definition in Section [B.14.2.26](#).

A3866 B.23.2.8 tms.Curve2D

A3867 See definition in Section [B.10.2.3](#).

A3868 B.23.2.9 tms.Point2DSequence

A3869 See definition in Section [B.10.2.4](#).

A3870 B.23.2.10 tms.Point2D

A3871 See definition in Section [B.10.2.5](#).

A3872 B.23.2.11 tms.DLSConfig

A3873 See definition in Section [B.14.2.30](#).

A3874 B.23.2.12 tms.TopicList

A3875 See definition in Section [B.14.2.31](#).

A3876 B.23.2.13 tms.TopicName

A3877 See definition in Section [B.14.2.32](#).

A3878 B.23.2.14 tms.PowerPortConfigRequest

A3879 See definition in Section [B.14.2.12](#).

A3880 B.23.2.15 tms.PowerPortNumber

A3881 See definition in Section [B.2.2.4](#).

A3882 B.23.2.16 tms.RequestSequence

A3883 See definition in Section [B.2.2.5](#).

A3884 B.23.2.17 tms.ReplyStatus

A3885 See definition in Section [B.2.2.6](#).

A3886 B.23.2.18 tms.String32

A3887 See definition in Section [B.2.2.7](#).

A3888 B.24 DC Load Sharing

A3889 B.24.1 Data Flows

A3890 Figure B.34 illustrates all the topics in the data flow.

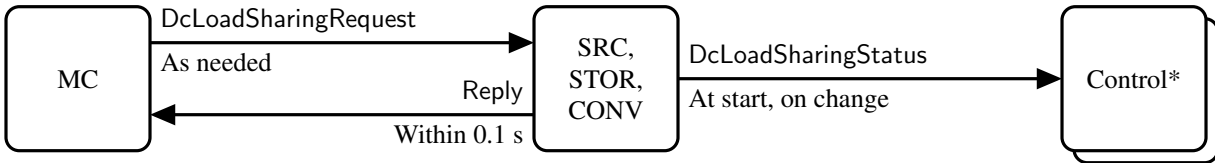


Figure B.34: DC Load Sharing data flow

A3891 Table B.91 provides an overview of how each topic is used.

Table B.91: Description of the DC Load Sharing topics.

Topic	Description
DcLoadSharingStatus	Reports the active direct current load sharing status of a device.
DcLoadSharingRequest	Remote request to change the DC load sharing status of a device.
Reply	The reply to a request.

A3892 Table B.92 specifies the publishers and subscribers for each topic as well as usage  
A3893 requirements as described in Section A.8.7.

Table B.92: Participants of the DC Load Sharing topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DcLoadSharingStatus	CONDITIONAL				Pub	Pub			Pub
DcLoadSharingRequest	OPTIONAL	Sub	Sub	Sub					
			Pub						
				Sub	Sub	Sub			Sub
Reply	REQUIRED	Pub	Pub		Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A3894 Table B.93 specifies the timing for when sample updates are sent to each topic. These  
A3895 parameters are defined in Section A.8.5.

Table B.93: Timing of the DC Load Sharing topics.  
Rate Burst

Topic	Data Trigger	Class	Size	QoS Profile
DcLoadSharingStatus	At Start, On Change	10 s	10	PublishLast
DcLoadSharingRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response

A3896 B.24.2 Data Types

A3897 Table B.94 specifies the data types that are used as sample values for each topic.

Table B.94: Data types of the DC Load Sharing topics.

Topic	Data Type
DcLoadSharingStatus	<code>tms.dc.LoadSharingStatus</code>
DcLoadSharingRequest	<code>tms.dc.LoadSharingRequest</code>
Reply	<code>tms.Reply</code>

A3898 Topic-level data types are defined in the order shown in Table B.94. Then nested data  
A3899 types are defined in the order of appearance. Cross-references are provided for data types  
A3900 that have already been defined.

A3901 B.24.2.1 `tms.dc.LoadSharingStatus`

A3902 PURPOSE: Report the present value of the load sharing parameters for direct current  
A3903 devices.

A3904 TOPIC USAGE: DcLoadSharingStatus

A3905 EXTENSIBILITY: extensibility(APPENDABLE)

A3906 PATTERN: Structure

A3907 ATTRIBUTES:

Name	Type and Description
<code>deviceId</code>	<code>tms::Fingerprint</code> The device described by this structure. Annotations: <i>keyval</i>
<code>config</code>	<code>tms::ConfigId</code> Configuration that these values belong to.
<code>parameters</code>	<code>LoadSharing</code> Parameter values.

A3909

A3910 B.24.2.2 tms.dc.LoadSharingRequest

A3911 See definition in Section [B.14.2.33](#).

A3912 B.24.2.3 tms.Reply

A3913 See definition in Section [B.2.2.1](#).

A3914 B.24.2.4 tms.Fingerprint

A3915 See definition in Section [B.2.2.2](#).

A3916 B.24.2.5 tms.ConfigId

A3917 See definition in Section [B.2.2.3](#).

A3918 B.24.2.6 tms.dc.LoadSharing

A3919 See definition in Section [B.14.2.34](#).

A3920 B.24.2.7 tms.ControlCurve

A3921 See definition in Section [B.14.2.26](#).

A3922 B.24.2.8 tms.Curve2D

A3923 See definition in Section [B.10.2.3](#).

A3924 B.24.2.9 tms.Point2DSequence

A3925 See definition in Section [B.10.2.4](#).

A3926 B.24.2.10 tms.Point2D

A3927 See definition in Section [B.10.2.5](#).

A3928 B.24.2.11 tms.PowerPortConfigRequest

A3929 See definition in Section [B.14.2.12](#).

A3930 B.24.2.12 tms.PowerPortNumber

A3931 See definition in Section [B.2.2.4](#).

A3932 B.24.2.13 tms.RequestSequence

A3933 See definition in Section [B.2.2.5](#).

A3934 B.24.2.14 tms.ReplyStatus

A3935 See definition in Section [B.2.2.6](#).

A3936 B.24.2.15 tms.String32

A3937 See definition in Section [B.2.2.7](#).

A3938 B.25 Load Policies

A3939 This section left intentionally blank.

A3940 B.26 Clock Status

A3941 B.26.1 Data Flows

A3942 Figure [B.35](#) illustrates all the topics in the data flow.

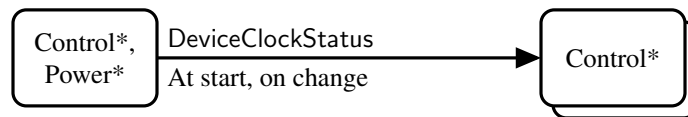


Figure B.35: Clock Status data flow

A3943 Table [B.95](#) provides an overview of how each topic is used.

Table B.95: Description of the Clock Status topics.

Topic	Description
DeviceClockStatus	Reports changes to a device clock(s). A device must support relative time and can optionally support absolute time.

A3944 Table [B.96](#) specifies the publishers and subscribers for each topic as well as usage  
 A3945 requirements as described in Section [A.8.7](#).

Table B.96: Participants of the Clock Status topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
DeviceClockStatus	OPTIONAL	Pub Sub	Pub Sub	Pub Sub	Pub	Pub	Pub	Pub	Pub

A3946 Table [B.97](#) specifies the timing for when sample updates are sent to each topic. These  
 A3947 parameters are defined in Section [A.8.5](#).

Table B.97: Timing of the Clock Status topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
		Class	Size	
DeviceClockStatus	At Start, On Change	10 s	1	PublishLast

A3948 B.26.2 Data Types

A3949 Table [B.98](#) specifies the data types that are used as sample values for each topic.

Table B.98: Data types of the Clock Status topics.

Topic	Data Type
DeviceClockStatus	<code>tms.ClockStatus</code>

A3950 Topic-level data types are defined in the order shown in Table [B.98](#). Then nested data  
A3951 types are defined in the order of appearance. Cross-references are provided for data types  
A3952 that have already been defined.



A3953 B.26.2.1 tms.ClockStatus

A3954 PURPOSE: Simultaneous measurement of both system clocks, and synchronization status  
A3955 for .

A3956 TOPIC USAGE: DeviceClockStatus

A3957 EXTENSIBILITY: extensibility(APPENDABLE)

A3958 PATTERN: Structure

A3959 ATTRIBUTES:

Name	Type and Description
deviceId	<p>Fingerprint</p> <p>The device described by this structure.</p> <p>Annotations: <i>keyval</i></p>
monotonic	<p>ClockMonotonic</p> <p>Measured monotonic time, sampled simultaneously with realtime.</p>
monotonicPersistence	<p>ClockMonotonicPersistenceType</p> <p>Indicate whether the monotonic clock initializes to 0 or accumulates operating time across restarts.</p>
realtime	<p>ClockRealtime</p> <p>Measured real time, sampled simultaneously with monotonic.</p>
realtimeSource	<p>octet</p> <p>Source of the realtime clock, as enumerated for timeSource in IEEE-1588-2008 Table 7.</p>
realtimeSetTime	<p>ClockRealtime</p> <p>Last time the realtime clock was updated by the source.</p>

A3961 CONSTANTS:

Name	Type [Units], Value, and Description
CLOCK_ATOMIC	octet, 0x10 Direct connection to a calibrated atomic clock.
CLOCK_GPS	octet, 0x20 Satellite-based system.
CLOCK_RADIO	octet, 0x30 Terrestrial radio system.
CLOCK_PTP	octet, 0x40 PTP, IEEE-1588.
CLOCK_NTP	octet, 0x50 NTP, RFC 5905 or SNTP, RFC 4330.
CLOCK_HAND	octet, 0x60 User input.
CLOCK_OTHER	octet, 0x90 Other source.
CLOCK_INTERNAL	octet, 0xA0 Arbitrary or unknown epoch.

A3962

A3963

A3964 B.26.2.2 tms.Fingerprint

A3965 See definition in Section [B.2.2.2](#).

A3966 B.26.2.3 tms.ClockMonotonic

A3967 See definition in Section [B.8.2.3](#).

A3968 B.26.2.4 tms.ClockMonotonicPersistenceType

A3969 PURPOSE: Indicate how the monotonic clock is initialized.

A3970 EXTENSIBILITY: extensibility(APPENDABLE)

A3971 PATTERN: Enumeration

A3972 VALUES:

Name	Description
CMPT_UNKNOWN	Fault detection indicates that initialization may be inconsistent.
CMPT_RESET	Clock initializes to 0.
CMPT_ACCUMULATE	Clock initializes to a value stored shortly before shutdown. This is effectively an aggregate runtime meter.
CMPT_OTHER	Another clock initialization scheme is used.

A3975 B.26.2.5 tms.ClockRealtime

A3976 PURPOSE: Clock synchronized to a global time base.

A3977 TOPIC USAGE: Nested

A3978 EXTENSIBILITY: extensibility(APPENDABLE)

A3979 PATTERN: Structure

A3980 ATTRIBUTES:

Name	Type and Description
epoch	uint16 Epoch.
seconds	uint32 integer portion of the timestamp.
nanoseconds	uint32 fractional portion of the timestamp.

A3983 DESCRIPTION:

A3984 This clock is used to measure the absolute time of events. The start time,  $t=00.0$ , of this  
A3985 clock is as specified in IEEE 1588-2008, Precision Time Protocol (PTP). The structure  
A3986 defined is based on the PTP Timestamp type. The **nanoseconds** is equivalent to the PTP  
A3987 **nanosecondsField**. The **seconds** is equivalent to the least-significant 32 bits of the PTP

A3988 secondsField. The epoch is the most-significant 16 bits of the PTP secondsField.

## A3989 B.27 Black Start

A3990 These objects are used to implement the manual black start protocol. These object and  
A3991 associated safety information shall be logged for traceability. By design, there is no support  
A3992 for saving them in the device configuration settings.

A3993

### A3994 B.27.1 Data Flows

A3995 Figure B.36 illustrates all the topics in the data flow.

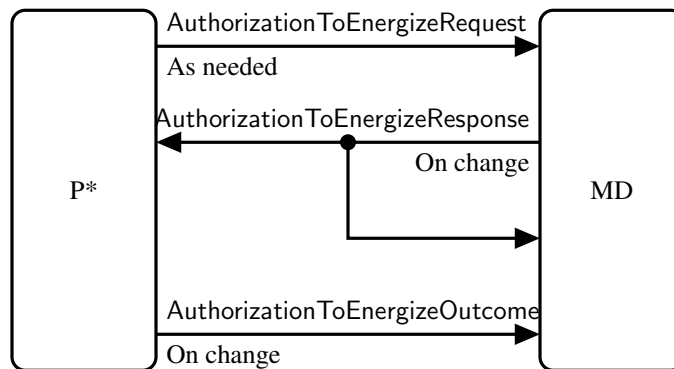


Figure B.36: Black Start data flow

A3996 When a device receives a manual black start command, it sends an  
A3997 AuthorizationToEnergizeRequest and waits for an AuthorizationToEnergizeResponse.  
A3998 When a response is received or times out, the device sends an AuthorizationToEnergizeOutcome.

A3999

A4000 Table B.99 provides an overview of how each topic is used.

Table B.99: Description of the Black Start topics.

Topic	Description
AuthorizationToEnergizeOutcome	Report the outcome of a command that required authorization.
AuthorizationToEnergizeRequest	Request authorization to execute a remote command when a black start is detected.
AuthorizationToEnergizeResponse	Provide the operator's response to an authorization request.

A4001 Table B.100 specifies the publishers and subscribers for each topic as well as usage  
A4002 requirements as described in Section A.8.7.

Table B.100: Participants of the Black Start topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
AuthorizationToEnergizeOutcome	OPTIONAL				Pub	Pub	Pub		Pub
AuthorizationToEnergizeRequest	OPTIONAL	Sub		Sub	Pub	Pub	Pub		Pub
AuthorizationToEnergizeResponse	CONDITIONAL	Sub		Sub					
		Pub							
		Sub	Sub		Sub	Sub	Sub		

A4003 Table B.101 specifies the timing for when sample updates are sent to each topic. These  
A4004 parameters are defined in Section A.8.5.

Table B.101: Timing of the Black Start topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
AuthorizationToEnergizeOutcome	Within 0.1 s	10 s	1	Response
AuthorizationToEnergizeRequest	As Needed	10 s	1	Command
AuthorizationToEnergizeResponse	As Needed	10 s	1	Response

## A4005 B.27.2 Data Types

A4006 Table B.102 specifies the data types that are used as sample values for each topic.

Table B.102: Data types of the Black Start topics.

Topic	Data Type
AuthorizationToEnergizeOutcome	tms.AuthorizationToEnergizeOutcome
AuthorizationToEnergizeRequest	tms.AuthorizationToEnergizeRequest
AuthorizationToEnergizeResponse	tms.AuthorizationToEnergizeResponse

A4007 Topic-level data types are defined in the order shown in Table B.102. Then nested data  
A4008 types are defined in the order of appearance. Cross-references are provided for data types  
A4009 that have already been defined.

A4010 B.27.2.1 tms.AuthorizationToEnergizeOutcome

A4011 PURPOSE: Report the outcome of a command that required authorization.

A4012 TOPIC USAGE: AuthorizationToEnergizeOutcome

A4013 EXTENSIBILITY: extensibility(APPENDABLE)

A4014 PATTERN: Structure

A4015 ATTRIBUTES:

*TMS Data Model*  
*DRAFT 19 Apr 2021*

Name	Type and Description
<b>relatedRequestId</b>	<b>DeviceRequest</b> Copy of the corresponding AuthorizationToEnergizeRequest.requestId. Annotations: <i>keyval</i>
<b>commandId</b>	<b>Fingerprint</b> Copy of the requestId from the corresponding command.
<b>deviceId</b>	<b>Fingerprint</b> Device that processed this authorization. Must match the corresponding command and request.
<b>accepted</b>	<b>boolean</b> True if the authorization was accepted, the command will be executed, and both relatedRequestId and commandId are valid.
<b>authReceived</b>	<b>boolean</b> True confirms that an authorization request was received. False indicates a response due to authorization timeout, and therefore relatedRequestId is invalid.
A4016 <b>authCommand</b>	<b>boolean</b> True confirms that commandId matched a pending command. False indicates the command was invalid, never received, timed out, or otherwise not pending execution.
<b>authLocation</b>	<b>boolean</b> True confirms that the authorization deviceId, portNumber, and continuity matched the command.
<b>authUser</b>	<b>boolean</b> True confirms that the userId was accepted.
<b>authTime</b>	<b>boolean</b> True confirms that the authorizationTime was accepted.
<b>authMonotonic</b>	<b>ClockMonotonic</b> Local monotonic clock when the acceptance decision was made.
<b>authRealtime</b>	<b>ClockRealtime</b> Local realtime clock when the acceptance decision was made.

**DESCRIPTION:**

This object is sent in response to an authorization request or the lack thereof. It documents

A4020 key details surrounding the authorization and resulting action. If the authorization is  
A4021 **accepted**, it documents that all required checks passed.

A4022 B.27.2.2 tms.AuthorizationToEnergizeRequest

A4023 PURPOSE: Request authorization for a manual black start command.

A4024 TOPIC USAGE: AuthorizationToEnergizeRequest

A4025 EXTENSIBILITY: extensibility(APPENDABLE)

A4026 PATTERN: Structure

A4027 ATTRIBUTES:

Name	Type and Description
<b>requestId</b>	<b>GridRequest</b> Identity of this request. Annotations: <i>keyval</i>
<b>sequenceId</b>	<b>tms::RequestSequence</b> GridRequest sequence data used to associate a request and returning reply.
<b>commandId</b>	<b>DeviceRequest</b> Copy of the requestId from the command that requires authorization.
<b>commandSequenceId</b>	<b>tms::RequestSequence</b> DeviceRequest sequence data used to associate a request and returning reply.
<b>deviceId</b>	<b>Fingerprint</b> Copy of the deviceId from the command that requires authorization.
<b>powerPorts</b>	<b>PowerPortNumberSequence</b> All power ports that would be energized by executing this command.

A4030 DESCRIPTION:

A4031 The **deviceId** and **ports** fully capture the location of the requested event. This request  
A4032 must be approved by an authorized operator.



A4033 B.27.2.3 tms.AuthorizationToEnergizeResponse

A4034 PURPOSE: Provide the operator's response to an authorization request.

A4035 TOPIC USAGE: AuthorizationToEnergizeResponse

A4036 EXTENSIBILITY: extensibility(APPENDABLE)

A4037 PATTERN: Structure

A4038 ATTRIBUTES:

Name	Type and Description
<b>relatedRequestId</b>	
	<b>GridRequest</b>
	Copy of the corresponding AuthorizationToEnergizeRequest.requestId.
	Annotations: <i>keyval</i>
<b>commandId</b>	
	<b>DeviceRequest</b>
	Copy of the requestId from the command that requires authorization.
<b>deviceId</b>	
	<b>Fingerprint</b>
	Copy of the deviceId from the command that requires authorization.
<b>powerPorts</b>	
	<b>PowerPortNumberSequence</b>
A4039	Copy of the AuthorizationToEnergizeRequest.ports.
<b>accept</b>	
	<b>boolean</b>
	True if the command is authorized to proceed.
<b>deny</b>	
	<b>boolean</b>
	True if the command must not be executed.
<b>userId</b>	
	<b>Fingerprint</b>
	Identity of the user that provided this response.
<b>authorizationTime</b>	
	<b>ClockRealtime</b>
A4040	Time when the user provided this response.

A4041 DESCRIPTION:

A4042 The black start policy specifies that a human operator needs to take positive steps to ensure  
A4043 the safety of everyone in the area when manual black start requirements are in effect. In  
A4044 accepting a request, the responding operator also confirms that appropriate safeguards are  
A4045 in place. A request may be denied for any reason.

A4046 The receiving device must compare **deviceId** and **ports** to the original command and any  
A4047 value mismatch shall cause this authorization to be rejected.

A4048 If both **accept** and **deny** are true in a single message or if multiple responses are received

A4049 before the command is completed, then any **deny** takes precedence and the command must  
A4050 not be executed.

A4051 In the user interface, **accept** should be red as a warning and **deny** should be green.

A4052 For validation, the **operatorId** may be compared to an access control list and the  
A4053 **authorizationTime** may be compared to the local clock.

A4054 This object must be published regardless of where the authorization occurs. For example,  
A4055 authorization may occur at the MC, at another remote user interface, or at a local user  
A4056 interface.

A4057 B.27.2.4 tms.DeviceRequest

A4058 See definition in Section [B.6.2.6](#).

A4059 B.27.2.5 tms.Fingerprint

A4060 See definition in Section [B.2.2.2](#).

A4061 B.27.2.6 tms.ClockMonotonic

A4062 See definition in Section [B.8.2.3](#).

A4063 B.27.2.7 tms.ClockRealtime

A4064 See definition in Section [B.26.2.5](#).

A4065 B.27.2.8 tms.GridRequest

A4066 See definition in Section [B.11.2.6](#).

A4067 B.27.2.9 tms.RequestSequence

A4068 See definition in Section [B.2.2.5](#).

A4069 B.27.2.10 tms.PowerPortNumberSequence

A4070 See definition in Section [B.5.2.48](#).

A4071 B.27.2.11 tms.PowerPortNumber

A4072 See definition in Section [B.2.2.4](#).

A4073 B.28 Microgrid Membership

A4074 Devices may be added to or removed from the microgrid at any time without notice. In  
A4075 some cases, adding or removing a device could cause an undesired transient, depending on

how the microgrid is configured. For example, adding loads, removing sources, or removing distribution could cause an undesirable load shed.

To avoid such issues, devices may have a manual or automatic option to use the microgrid membership messages. These messages give the MC time to adjust power flows in preparation for the change. For example, the MC may increase spinning reserves or enable backup power flows.

Under normal operation, devices may join or leave the microgrid at any time and without notification. The membership messages are advisory, and they cannot prevent an “unclean” membership change. Other authentication and authorization steps may be required before a new device is allowed to communicate.

*Editorial note: The leave notification only needs a yes/no indication, since the MC can observe the present status using other messages. The join notification may require additional information, such as the LoadChangeRequest*

### B.28.1 Data Flows

This flow starts when a power device publishes a **MicrogridMembershipRequest**. The MC acknowledges the request, prepares for the change, and sends the outcome when preparations are complete. The power device waits for this outcome. The MD subscribes to this flow to display its status. The MD may also be used to publish a membership change for a power device.

Figure B.37 illustrates all the topics in the data flow.

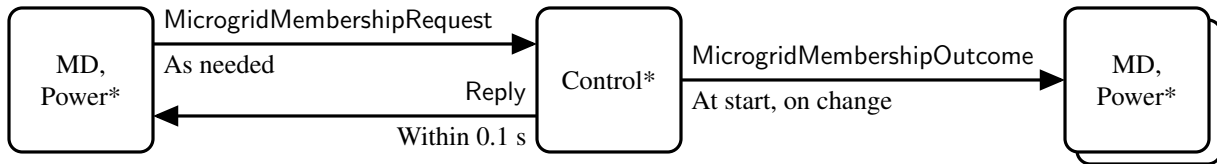


Figure B.37: Microgrid Membership data flow

When the MC receives a valid **MicrogridMembershipRequest**, it performs the following sequence of actions.

1. Acknowledge the request with **RequestResponse** set to **REPLY\_OK**.
2. Prepare for the change. Send all necessary commands and wait for status updates.
3. Complete the request with **MicrogridMembershipOutcome**.

The **MicrogridMembershipOutcome** is **MMR\_COMPLETE** if preparations completed successfully, otherwise it is **MMR\_BLOCKED**. Preparations may be blocked by manual steps or insufficient resources. The MC sends the blocked outcome with any available hint, optionally issues alerts for further explanation, and is finished with the request. It is then the operator’s responsibility to complete, defer, or cancel the membership change.

If the MC receives a new **MicrogridMembershipRequest** while an old request for the same device is still incomplete, then the MC performs the following sequence of actions.

- A4109 1. If the new request is invalid, then reject it with an error **RequestResponse** and proceed  
A4110 as if it had not been received.
- A4111 2. Acknowledge the new request with **RequestResponse** set to **REPLY\_OK**.
- A4112 3. Complete the old request with **MicrogridMembershipApproval** set to **MMR\_REPLACED**.
- A4113 4. Halt any incomplete preparations that are no longer necessary. Preparations that were  
A4114 completed for the old request may block preparation for the new request.
- A4115 5. Prepare for the new change. Send any new necessary commands and wait for status  
A4116 updates.
- A4117 6. Complete the new request with **MicrogridMembershipOutcome** set to **MMR\_COMPLETE** or  
A4118 **MMR\_BLOCKED**.
- A4119 Displays on the power device or MD may show the membership change as requested  
A4120 (waiting for response), pending (waiting for outcome), completed, blocked, or replaced. The  
A4121 display should also show the hint provided for blocked requests.
- A4122 Table [B.103](#) provides an overview of how each topic is used.

Table B.103: Description of the Microgrid Membership topics.

Topic	Description
<b>MicrogridMembershipOutcome</b>	Outcome of a Microgrid Membership Request.
<b>MicrogridMembershipRequest</b>	A power device requests to leave or join a grid.
<b>Reply</b>	The reply to a request.

- A4123 Table [B.104](#) specifies the publishers and subscribers for each topic as well as usage  
A4124 requirements as described in Section [A.8.7](#).

Table B.104: Participants of the Microgrid Membership topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
<b>MicrogridMembershipOutcome</b>	OPTIONAL		Pub						
<b>MicrogridMembershipRequest</b>	OPTIONAL			Sub	Sub	Sub	Sub	Sub	Sub
<b>Reply</b>	REQUIRED	Sub	Sub	Sub	Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

- A4125 Table [B.105](#) specifies the timing for when sample updates are sent to each topic. These  
A4126 parameters are defined in Section [A.8.5](#).

Table B.105: Timing of the Microgrid Membership topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
MicrogridMembershipOutcome	As Needed	10 s	10	Response
MicrogridMembershipRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response

A4127 B.28.2 Data Types

A4128 Table [B.106](#) specifies the data types that are used as sample values for each topic.

Table B.106: Data types of the Microgrid Membership topics.

Topic	Data Type
MicrogridMembershipOutcome	tms.MicrogridMembershipApproval
MicrogridMembershipRequest	tms.MicrogridMembershipRequest
Reply	tms.Reply

A4129 Topic-level data types are defined in the order shown in Table [B.106](#). Then nested data  
A4130 types are defined in the order of appearance. Cross-references are provided for data types  
A4131 that have already been defined.

A4132 B.28.2.1 tms.MicrogridMembershipApproval

A4133 PURPOSE: Return the results of a MicrogridMembershipRequest.

A4134 TOPIC USAGE: MicrogridMembershipOutcome

A4135 EXTENSIBILITY: extensibility(APPENDABLE)

A4136 PATTERN: Structure

A4137 ATTRIBUTES:

Name	Type and Description
requestId	GridRequest Identity of this request. Annotations: <i>keyval</i>
relatedRequestId	GridRequest Copy of the corresponding MicrogridMembershipRequest.requestId.
requestSequenceId	tms::RequestSequence Copy of the RequestSequence data from the processed request.
deviceId	Fingerprint Device or platform that has been approved.
membership	MicrogridMembership Requested state.
result	MicrogridMembershipResult Indicate whether the microgrid is prepared to complete this request.
hint	String32 Short, human-readable text description of a blocked request. This should summarize to the operator what the issue is. For example, blocked by interruptible load, blocked by critical load, or manual action required.

A4139

A4140 B.28.2.2 tms.MicrogridMembershipRequest

A4141 PURPOSE: Request preparations to cleanly join or leave the microgrid.

A4142 TOPIC USAGE: MicrogridMembershipRequest

A4143 EXTENSIBILITY: extensibility(APPENDABLE)

A4144 PATTERN: Structure

A4145 ATTRIBUTES:

Name	Type and Description
requestId	<b>GridRequest</b> Identity of this request. Annotations: <i>keyval</i>
sequenceId	<b>tms::RequestSequence</b> Request sequence data used to associate a request and returning reply.
requestingId	<b>Fingerprint</b> Device or platform requesting the change. A platform ID indicates that all devices on the platform are affected.
membership	<b>MicrogridMembership</b> The desired membership state.

A4148 B.28.2.3 tms.Reply

A4149 See definition in Section [B.2.2.1](#).

A4150 B.28.2.4 tms.GridRequest

A4151 See definition in Section [B.11.2.6](#).

A4152 B.28.2.5 tms.Fingerprint

A4153 See definition in Section [B.2.2.2](#).

A4154 B.28.2.6 tms.RequestSequence

A4155 See definition in Section [B.2.2.5](#).

A4156 B.28.2.7 tms.MicrogridMembership

A4157 PURPOSE: Indicate an intent to join or leave the microgrid.

A4158 EXTENSIBILITY: extensibility(APPENDABLE)

A4159 PATTERN: Enumeration

A4160 VALUES:

Name	Description
MM_UNKNOWN	Uninitialized value.
MM_JOIN	Join or stay in the microgrid.
MM_LEAVE	Leave or stay out of the microgrid.

A4163 B.28.2.8 tms.MicrogridMembershipResult

A4164 PURPOSE: Indicate the result of a membership request.

A4165 EXTENSIBILITY: extensibility(APPENDABLE)

A4166 PATTERN: Enumeration

A4167 VALUES:

Name	Description
MMR_UNKNOWN	Uninitialized value.
MMR_REPLACED	New request received before preparation for this request was complete. Device should wait for the new request to complete.
MMR_COMPLETE	Completed preparation for the requested action. No significant issues were encountered.
MMR_BLOCKED	Preparation could not be completed without a significant negative impact such as shedding load. Operator should resolve the issue before proceeding.

A4170 B.28.2.9 tms.String32

A4171 See definition in Section [B.2.2.7](#).

A4172 B.28.2.10 tms.ConfigId

A4173 See definition in Section [B.2.2.3](#).



A4174 B.28.2.11 tms.PowerPortNumber

A4175 See definition in Section [B.2.2.4](#).

A4176 B.28.2.12 tms.ReplyStatus

A4177 See definition in Section [B.2.2.6](#).

A4178 B.29 Energy Storage Control

A4179 B.29.1 Data Flows

A4180 Figure [B.38](#) illustrates all the topics in the data flow.

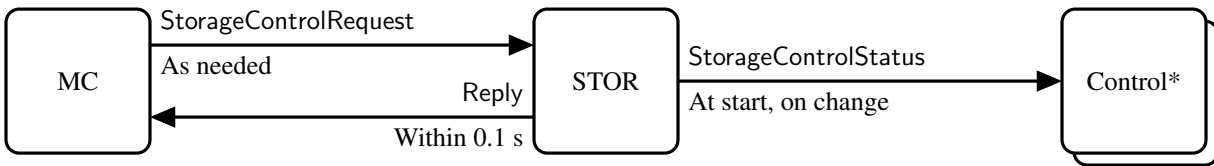


Figure B.38: Energy Storage Control data flow

A4181 Table [B.107](#) provides an overview of how each topic is used.

Table B.107: Description of the Energy Storage Control topics.

Topic	Description
StorageControlStatus	Latest storage control parameters.
StorageControlRequest	Remote request to set the storage control parameters.
Reply	The reply to a request.

A4182 Table [B.108](#) specifies the publishers and subscribers for each topic as well as usage  
A4183 requirements as described in Section [A.8.7](#).

Table B.108: Participants of the Energy Storage Control topics.

Topic	Usage	MD	MC	MON	SRC	STOR	DIST	LOAD	CONV
StorageControlStatus	REQUIRED					Pub			
StorageControlRequest	OPTIONAL	Sub	Sub	Sub					
			Pub						
				Sub		Sub			
Reply	REQUIRED	Pub	Pub		Pub	Pub	Pub	Pub	Pub
		Sub	Sub	Sub					

A4184 Table [B.109](#) specifies the timing for when sample updates are sent to each topic. These  
A4185 parameters are defined in Section [A.8.5](#).

Table B.109: Timing of the Energy Storage Control topics.

Topic	Data Trigger	Rate	Burst	QoS Profile
		Class	Size	
StorageControlStatus	At Start, On Change	10 s	10	PublishLast
StorageControlRequest	As Needed	10 s	10	Command
Reply	Within 0.1 s	All	1	Response

A4186 B.29.2 Data Types

A4187 Table B.110 specifies the data types that are used as sample values for each topic.

Table B.110: Data types of the Energy Storage Control topics.

Topic	Data Type
StorageControlStatus	<code>tms.stor.StorageControlStatus</code>
StorageControlRequest	<code>tms.stor.StorageControlRequest</code>
Reply	<code>tms.Reply</code>

A4188 Topic-level data types are defined in the order shown in Table B.110. Then nested data  
A4189 types are defined in the order of appearance. Cross-references are provided for data types  
A4190 that have already been defined.

A4191 B.29.2.1 tms.stor.StorageControlStatus

A4192 PURPOSE: Report the present value of the storage control parameters.

A4193 TOPIC USAGE: StorageControlStatus

A4194 EXTENSIBILITY: extensibility(APPENDABLE)

A4195 PATTERN: Structure

A4196 ATTRIBUTES:

Name	Type and Description
<code>deviceId</code>	<code>tms::Fingerprint</code> The device described by this structure. Annotations: <i>keyval</i>
<code>config</code>	<code>tms::ConfigId</code> Configuration that this command belongs to.
<code>parameters</code>	<code>StorageControlParameters</code> Parameter values.

A4198

- A4199 B.29.2.2 tms.stor.StorageControlRequest
- A4200 See definition in Section [B.14.2.35](#).
- A4201 B.29.2.3 tms.Reply
- A4202 See definition in Section [B.2.2.1](#).
- A4203 B.29.2.4 tms.Fingerprint
- A4204 See definition in Section [B.2.2.2](#).
- A4205 B.29.2.5 tms.ConfigId
- A4206 See definition in Section [B.2.2.3](#).
- A4207 B.29.2.6 tms.stor.StorageControlParameters
- A4208 See definition in Section [B.14.2.36](#).
- A4209 B.29.2.7 tms.Duration
- A4210 See definition in Section [B.11.2.11](#).
- A4211 B.29.2.8 tms.stor.StorageOperatingState
- A4212 See definition in Section [B.14.2.38](#).
- A4213 B.29.2.9 tms.stor.UnderFrequencyStopCharge
- A4214 See definition in Section [B.14.2.39](#).
- A4215 B.29.2.10 tms.stor.OverFrequencyStopDischarge
- A4216 See definition in Section [B.14.2.40](#).
- A4217 B.29.2.11 tms.DeviceConfigRequest
- A4218 See definition in Section [B.14.2.3](#).
- A4219 B.29.2.12 tms.RequestSequence
- A4220 See definition in Section [B.2.2.5](#).
- A4221 B.29.2.13 tms.PowerPortNumber
- A4222 See definition in Section [B.2.2.4](#).

A4223 B.29.2.14 tms.ReplyStatus

A4224 See definition in Section [B.2.2.6](#).

A4225 B.29.2.15 tms.String32

A4226 See definition in Section [B.2.2.7](#).

A4227 B.30 Cross Reference

A4228 This section contains alphabetical listings of all data types, with references to their definition.  
A4229 It contains one table for all namespaces, followed by separate tables for each namespace. An  
A4230 index of constants is also provided after the data types.

A4231 B.30.1 Data Types in All Namespaces

A4232 Each entry is of the format “type, namespace, page”.

A4233	ActiveConversionInfo, tms, <a href="#">107</a>	A4266	DeviceFingerprint, tms, <a href="#">153</a>
A4234	ActiveDiagnosticMessages, tms, <a href="#">158</a>	A4267	DeviceFingerprintSequence, tms, <a href="#">153</a>
A4235	AuthorizationToEnergizeOutcome, tms,	A4268	DeviceGroundingStatus, tms, <a href="#">196</a>
A4236	<a href="#">241</a>	A4269	DeviceIcon, tms, <a href="#">123</a>
A4237	AuthorizationToEnergizeRequest, tms,	A4270	DeviceInfo, tms, <a href="#">77</a>
A4238	<a href="#">243</a>	A4271	DeviceIntent, tms, <a href="#">154</a>
A4239	AuthorizationToEnergizeResponse, tms,	A4272	DeviceIntentSequence, tms, <a href="#">153</a>
A4240	<a href="#">244</a>	A4273	DevicePowerMeasurements, tms.ac, <a href="#">213</a>
		A4274	DevicePowerMeasurements, tms.dc, <a href="#">220</a>
A4241	CableSenseStatus, tms, <a href="#">209</a>	A4275	DevicePowerPortStatuses, tms, <a href="#">206</a>
A4242	ChangeNicknameRequest, tms, <a href="#">120</a>	A4276	DeviceRequest, tms, <a href="#">121</a>
A4243	CircuitContinuity, tms, <a href="#">199</a>	A4277	DeviceRole, tms, <a href="#">79</a>
A4244	ClockMonotonic, tms, <a href="#">127</a>	A4278	DiagnosticTroubleCode, tms, <a href="#">159</a>
A4245	ClockMonotonicPersistenceType, tms, <a href="#">238</a>	A4279	DiagnosticTroubleCodeSequence, tms, <a href="#">158</a>
A4246	ClockRealtime, tms, <a href="#">238</a>	A4280	DistributionFeature, tms, <a href="#">112</a>
A4247	ClockStatus, tms, <a href="#">236</a>	A4281	DistributionFeatureSequence, tms, <a href="#">112</a>
A4248	ConfigId, tms, <a href="#">68</a>	A4282	DistributionInfo, tms, <a href="#">112</a>
A4249	ConnectorFeature, tms, <a href="#">103</a>	A4283	DLSSConfig, tms, <a href="#">180</a>
A4250	ConnectorFeatureSequence, tms, <a href="#">103</a>	A4284	DtcSeverity, tms, <a href="#">161</a>
A4251	ControlCurve, tms, <a href="#">179</a>	A4285	Duration, tms, <a href="#">151</a>
A4252	ControlHardwareInfo, tms, <a href="#">83</a>		
A4253	ControlHardwareState, tms, <a href="#">126</a>	A4286	EnergyStorageInfo, tms, <a href="#">87</a>
A4254	ControlParameterRequest, tms, <a href="#">175</a>	A4287	EnergyStorageState, tms, <a href="#">132</a>
A4255	ControlParameterStatus, tms, <a href="#">192</a>	A4288	EngineInfo, tms, <a href="#">85</a>
A4256	ControlServiceInfo, tms, <a href="#">92</a>	A4289	EngineState, tms, <a href="#">130</a>
A4257	ConversionFeature, tms, <a href="#">106</a>	A4290	EnumLabelsSequence, tms, <a href="#">92</a>
A4258	ConversionFeatureSequence, tms, <a href="#">106</a>		
A4259	ConversionInfo, tms, <a href="#">106</a>	A4291	FailureModeIndicator, tms, <a href="#">160</a>
A4260	ConversionTapInfo, tms, <a href="#">111</a>	A4292	Fingerprint, tms, <a href="#">68</a>
A4261	ConversionTapInfoSequence, tms, <a href="#">111</a>	A4293	FingerprintNickname, tms, <a href="#">119</a>
A4262	CopyConfigRequest, tms, <a href="#">189</a>	A4294	FuelInfo, tms, <a href="#">86</a>
A4263	Curve2D, tms, <a href="#">139</a>	A4295	FuelState, tms, <a href="#">131</a>
A4264	DesiredCircuitContinuity, tms, <a href="#">172</a>	A4296	GeneratorInfo, tms, <a href="#">86</a>
A4265	DeviceConfigRequest, tms, <a href="#">166</a>	A4297	GeneratorState, tms, <a href="#">131</a>

A4298	GetConfigContentsRequest, tms, <a href="#">165</a>	A4340	OperatingMode, tms, <a href="#">149</a>
A4299	GetConfigContentsResponse, tms, <a href="#">166</a>	A4341	OperatorIntent, tms, <a href="#">147</a>
A4300	GlobalTradeItemNumber, tms, <a href="#">81</a>	A4342	OperatorIntentRequest, tms, <a href="#">146</a>
A4301	GridRequest, tms, <a href="#">147</a>	A4343	OperatorIntentState, tms, <a href="#">145</a>
A4302	GroundFaultDetection, tms, <a href="#">200</a>	A4344	OperatorIntentType, tms, <a href="#">148</a>
A4303	GroundingCircuitNumber, tms, <a href="#">104</a>	A4345	OperatorPriority, tms, <a href="#">154</a>
A4304	GroundingCommand, tms, <a href="#">177</a>	A4346	OperatorPriorityType, tms, <a href="#">151</a>
A4305	GroundingCommandSequence, tms, <a href="#">176</a>	A4347	OverFrequencyStopDischarge, tms.stor,
A4306	GroundingDesignType, tms, <a href="#">105</a>	A4348	<a href="#">187</a>
A4307	GroundingInfo, tms, <a href="#">104</a>	A4349	ParameterMetadata, tms, <a href="#">91</a>
A4308	GroundingInfoSequence, tms, <a href="#">103</a>	A4350	ParameterMetadataSequence, tms, <a href="#">90</a>
A4309	GroundingStatus, tms, <a href="#">197</a>	A4351	ParameterValue, tms, <a href="#">176</a>
A4310	GroundingStatusSequence, tms, <a href="#">196</a>	A4352	ParameterValueSequence, tms, <a href="#">175</a>
A4311	Heartbeat, tms, <a href="#">74</a>	A4353	PassiveConversionInfo, tms, <a href="#">110</a>
A4312	LoadFeature, tms, <a href="#">117</a>	A4354	PhaseAnglesSequence, tms.ac, <a href="#">219</a>
A4313	LoadFeatureSequence, tms, <a href="#">116</a>	A4355	Point2D, tms, <a href="#">140</a>
A4314	LoadInfo, tms, <a href="#">116</a>	A4356	Point2DSequence, tms, <a href="#">140</a>
A4315	LoadSharing, tms.ac, <a href="#">178</a>	A4357	PowerConnection, tms, <a href="#">225</a>
A4316	LoadSharing, tms.dc, <a href="#">182</a>	A4358	PowerConnectionDetectionType, tms, <a href="#">227</a>
A4317	LoadSharingInfo, tms, <a href="#">108</a>	A4359	PowerConnectionId, tms, <a href="#">226</a>
A4318	LoadSharingInfoSequence, tms, <a href="#">107</a>	A4360	PowerConnectionList, tms, <a href="#">224</a>
A4319	LoadSharingIntent, tms, <a href="#">150</a>	A4361	PowerConnectionSequence, tms, <a href="#">225</a>
A4320	LoadSharingRequest, tms.ac, <a href="#">178</a>	A4362	PowerConnectorPhases, tms, <a href="#">101</a>
A4321	LoadSharingRequest, tms.dc, <a href="#">181</a>	A4363	PowerConnectorPolarity, tms, <a href="#">101</a>
A4322	LoadSharingStatus, tms.ac, <a href="#">229</a>	A4364	PowerConnectorType, tms, <a href="#">100</a>
A4323	LoadSharingStatus, tms.dc, <a href="#">232</a>	A4365	PowerDeviceConfig, tms, <a href="#">168</a>
A4324	MetricParameterStatus, tms, <a href="#">193</a>	A4366	PowerDeviceInfo, tms, <a href="#">95</a>
A4325	MicrogridControllerFeature, tms, <a href="#">93</a>	A4367	PowerElectronicsInfo, tms, <a href="#">88</a>
A4326	MicrogridControllerFeatureSequence, tms,	A4368	PowerElectronicsState, tms, <a href="#">132</a>
A4327	<a href="#">93</a>	A4369	PowerHardwareInfo, tms, <a href="#">84</a>
A4328	MicrogridControllerInfo, tms, <a href="#">93</a>	A4370	PowerHardwareState, tms, <a href="#">129</a>
A4329	MicrogridDashboardFeature, tms, <a href="#">94</a>	A4371	PowerMeasurement, tms.ac, <a href="#">214</a>
A4330	MicrogridDashboardFeatureSequence,	A4372	PowerMeasurement, tms.dc, <a href="#">222</a>
A4331	tms, <a href="#">94</a>	A4373	PowerMeasurementLine, tms.ac, <a href="#">216</a>
A4332	MicrogridDashboardInfo, tms, <a href="#">94</a>	A4374	PowerMeasurementLineSequence, tms.ac,
A4333	MicrogridIntent, tms, <a href="#">148</a>	A4375	<a href="#">216</a>
A4334	MicrogridMembership, tms, <a href="#">251</a>	A4376	PowerPortConfigRequest, tms, <a href="#">171</a>
A4335	MicrogridMembershipApproval, tms, <a href="#">249</a>	A4377	PowerPortConversionInfo, tms, <a href="#">110</a>
A4336	MicrogridMembershipRequest, tms, <a href="#">250</a>	A4378	PowerPortConversionInfoSequence, tms,
A4337	MicrogridMembershipResult, tms, <a href="#">251</a>	A4379	<a href="#">110</a>
A4338	NatoStockNumber, tms, <a href="#">81</a>	A4380	PowerPortDirectionality, tms, <a href="#">99</a>
A4339	OctetSequence, tms, <a href="#">124</a>	A4381	PowerPortId, tms, <a href="#">210</a>
		A4382	PowerPortInfo, tms, <a href="#">97</a>
		A4383	PowerPortInfoSequence, tms, <a href="#">96</a>

*TMS Data Model*  
*DRAFT 19 Apr 2021*

A4384	PowerPortIntent, tms, <a href="#">155</a>	A4413	StorageCellState, tms.stor, <a href="#">141</a>
A4385	PowerPortIntentSequence, tms, <a href="#">155</a>	A4414	StorageCellStateSequence, tms.stor, <a href="#">140</a>
A4386	PowerPortMeasurement, tms.ac, <a href="#">214</a>	A4415	StorageControlParameters, tms.stor, <a href="#">184</a>
A4387	PowerPortMeasurement, tms.dc, <a href="#">221</a>	A4416	StorageControlRequest, tms.stor, <a href="#">183</a>
A4388	PowerPortMeasurementSequence, tms.ac, <a href="#">213</a>	A4417	StorageControlStatus, tms.stor, <a href="#">253</a>
A4389		A4418	StorageFeature, tms, <a href="#">115</a>
A4390	PowerPortMeasurementSequence, tms.dc, <a href="#">221</a>	A4419	StorageFeatureSequence, tms, <a href="#">115</a>
A4391		A4420	StorageInfo, tms, <a href="#">114</a>
A4392	PowerPortNumber, tms, <a href="#">69</a>	A4421	StorageIntent, tms, <a href="#">152</a>
A4393	PowerPortNumberSequence, tms, <a href="#">105</a>	A4422	StorageOperatingState, tms.stor, <a href="#">186</a>
A4394	PowerPortStatus, tms, <a href="#">208</a>	A4423	StorageState, tms.stor, <a href="#">136</a>
A4395	PowerPortStatusSequence, tms, <a href="#">207</a>	A4424	String1-32, tms, <a href="#">83</a>
A4396	PowerSwitchCommand, tms, <a href="#">170</a>	A4425	String16, tms, <a href="#">92</a>
A4397	PowerSwitchCommandSequence, tms, <a href="#">169</a>	A4426	String32, tms, <a href="#">72</a>
A4398	PowerSwitchReason, tms, <a href="#">199</a>	A4427	SuspectParameterNumber, tms, <a href="#">160</a>
A4399	PowerSwitchStatus, tms, <a href="#">198</a>	A4428	SwitchFeature, tms, <a href="#">102</a>
A4400	ProductInfo, tms, <a href="#">80</a>	A4429	SwitchFeatureSequence, tms, <a href="#">102</a>
A4401	Reply, tms, <a href="#">67</a>	A4430	TapNumber, tms, <a href="#">111</a>
A4402	ReplyStatus, tms, <a href="#">70</a>	A4431	ThermalInfo, tms, <a href="#">89</a>
A4403	RequestSequence, tms, <a href="#">69</a>	A4432	ThermalLoadSequence, tms, <a href="#">134</a>
A4404	SourceFeature, tms, <a href="#">114</a>	A4433	ThermalState, tms, <a href="#">133</a>
A4405	SourceFeatureSequence, tms, <a href="#">113</a>	A4434	ThermalZoneSequence, tms, <a href="#">89</a>
A4406	SourceInfo, tms, <a href="#">113</a>	A4435	TopicConditionSequence, tms, <a href="#">82</a>
A4407	SourceState, tms, <a href="#">204</a>	A4436	TopicInfo, tms, <a href="#">82</a>
A4408	SourceTransition, tms, <a href="#">174</a>	A4437	TopicList, tms, <a href="#">181</a>
A4409	SourceTransitionRequest, tms, <a href="#">173</a>	A4438	TopicName, tms, <a href="#">181</a>
A4410	SourceTransitionState, tms, <a href="#">203</a>	A4439	TransformerInfo, tms, <a href="#">89</a>
A4411	StandardConfigMaster, tms, <a href="#">163</a>	A4440	TransformerState, tms, <a href="#">133</a>
A4412	StorageCellNumber, tms.stor, <a href="#">142</a>	A4441	UnderFrequencyStopCharge, tms.stor, <a href="#">186</a>
A4442	<u>B.30.2 Data Types in Namespace tms</u>		
A4443	ActiveConversionInfo, <a href="#">107</a>	A4453	ClockRealtime, <a href="#">238</a>
A4444	ActiveDiagnosticMessages, <a href="#">158</a>	A4454	ClockStatus, <a href="#">236</a>
A4445	AuthorizationToEnergizeOutcome, <a href="#">241</a>	A4455	ConfigId, <a href="#">68</a>
A4446	AuthorizationToEnergizeRequest, <a href="#">243</a>	A4456	ConnectorFeature, <a href="#">103</a>
A4447	AuthorizationToEnergizeResponse, <a href="#">244</a>	A4457	ConnectorFeatureSequence, <a href="#">103</a>
A4448	CableSenseStatus, <a href="#">209</a>	A4458	ControlCurve, <a href="#">179</a>
A4449	ChangeNicknameRequest, <a href="#">120</a>	A4459	ControlHardwareInfo, <a href="#">83</a>
A4450	CircuitContinuity, <a href="#">199</a>	A4460	ControlHardwareState, <a href="#">126</a>
A4451	ClockMonotonic, <a href="#">127</a>	A4461	ControlParameterRequest, <a href="#">175</a>
A4452	ClockMonotonicPersistenceType, <a href="#">238</a>	A4462	ControlParameterStatus, <a href="#">192</a>
		A4463	ControlServiceInfo, <a href="#">92</a>

A4464	ConversionFeature, 106	A4506	GridRequest, 147
A4465	ConversionFeatureSequence, 106	A4507	GroundFaultDetection, 200
A4466	ConversionInfo, 106	A4508	GroundingCircuitNumber, 104
A4467	ConversionTapInfo, 111	A4509	GroundingCommand, 177
A4468	ConversionTapInfoSequence, 111	A4510	GroundingCommandSequence, 176
A4469	CopyConfigRequest, 189	A4511	GroundingDesignType, 105
A4470	Curve2D, 139	A4512	GroundingInfo, 104
A4471	DesiredCircuitContinuity, 172	A4513	GroundingInfoSequence, 103
A4472	DeviceConfigRequest, 166	A4514	GroundingStatus, 197
A4473	DeviceFingerprint, 153	A4515	GroundingStatusSequence, 196
A4474	DeviceFingerprintSequence, 153	A4516	Heartbeat, 74
A4475	DeviceGroundingStatus, 196	A4517	LoadFeature, 117
A4476	DeviceIcon, 123	A4518	LoadFeatureSequence, 116
A4477	DeviceInfo, 77	A4519	LoadInfo, 116
A4478	DeviceIntent, 154	A4520	LoadSharingInfo, 108
A4479	DeviceIntentSequence, 153	A4521	LoadSharingInfoSequence, 107
A4480	DevicePowerPortStatuses, 206	A4522	LoadSharingIntent, 150
A4481	DeviceRequest, 121	A4523	MetricParameterStatus, 193
A4482	DeviceRole, 79	A4524	MicrogridControllerFeature, 93
A4483	DiagnosticTroubleCode, 159	A4525	MicrogridControllerFeatureSequence, 93
A4484	DiagnosticTroubleCodeSequence, 158	A4526	MicrogridControllerInfo, 93
A4485	DistributionFeature, 112	A4527	MicrogridDashboardFeature, 94
A4486	DistributionFeatureSequence, 112	A4528	MicrogridDashboardFeatureSequence, 94
A4487	DistributionInfo, 112	A4529	MicrogridDashboardInfo, 94
A4488	DLSSConfig, 180	A4530	MicrogridIntent, 148
A4489	DtcSeverity, 161	A4531	MicrogridMembership, 251
A4490	Duration, 151	A4532	MicrogridMembershipApproval, 249
A4491	EnergyStorageInfo, 87	A4533	MicrogridMembershipRequest, 250
A4492	EnergyStorageState, 132	A4534	MicrogridMembershipResult, 251
A4493	EngineInfo, 85	A4535	NatoStockNumber, 81
A4494	EngineState, 130	A4536	OctetSequence, 124
A4495	EnumLabelsSequence, 92	A4537	OperatingMode, 149
A4496	FailureModeIndicator, 160	A4538	OperatorIntent, 147
A4497	Fingerprint, 68	A4539	OperatorIntentRequest, 146
A4498	FingerprintNickname, 119	A4540	OperatorIntentState, 145
A4499	FuelInfo, 86	A4541	OperatorIntentType, 148
A4500	FuelState, 131	A4542	OperatorPriority, 154
A4501	GeneratorInfo, 86	A4543	OperatorPriorityType, 151
A4502	GeneratorState, 131	A4544	ParameterMetadata, 91
A4503	GetConfigContentsRequest, 165	A4545	ParameterMetadataSequence, 90
A4504	GetConfigContentsResponse, 166	A4546	ParameterValue, 176
A4505	GlobalTradeItemNumber, 81		



A4547	ParameterValueSequence, <a href="#">175</a>	A4582	ProductInfo, <a href="#">80</a>
A4548	PassiveConversionInfo, <a href="#">110</a>	A4583	Reply, <a href="#">67</a>
A4549	Point2D, <a href="#">140</a>	A4584	ReplyStatus, <a href="#">70</a>
A4550	Point2DSequence, <a href="#">140</a>	A4585	RequestSequence, <a href="#">69</a>
A4551	PowerConnection, <a href="#">225</a>	A4586	SourceFeature, <a href="#">114</a>
A4552	PowerConnectionDetectionType, <a href="#">227</a>	A4587	SourceFeatureSequence, <a href="#">113</a>
A4553	PowerConnectionId, <a href="#">226</a>	A4588	SourceInfo, <a href="#">113</a>
A4554	PowerConnectionList, <a href="#">224</a>	A4589	SourceState, <a href="#">204</a>
A4555	PowerConnectionSequence, <a href="#">225</a>	A4590	SourceTransition, <a href="#">174</a>
A4556	PowerConnectorPhases, <a href="#">101</a>	A4591	SourceTransitionRequest, <a href="#">173</a>
A4557	PowerConnectorPolarity, <a href="#">101</a>	A4592	SourceTransitionState, <a href="#">203</a>
A4558	PowerConnectorType, <a href="#">100</a>	A4593	StandardConfigMaster, <a href="#">163</a>
A4559	PowerDeviceConfig, <a href="#">168</a>	A4594	StorageFeature, <a href="#">115</a>
A4560	PowerDeviceInfo, <a href="#">95</a>	A4595	StorageFeatureSequence, <a href="#">115</a>
A4561	PowerElectronicsInfo, <a href="#">88</a>	A4596	StorageInfo, <a href="#">114</a>
A4562	PowerElectronicsState, <a href="#">132</a>	A4597	StorageIntent, <a href="#">152</a>
A4563	PowerHardwareInfo, <a href="#">84</a>	A4598	String1-32, <a href="#">83</a>
A4564	PowerHardwareState, <a href="#">129</a>	A4599	String16, <a href="#">92</a>
A4565	PowerPortConfigRequest, <a href="#">171</a>	A4600	String32, <a href="#">72</a>
A4566	PowerPortConversionInfo, <a href="#">110</a>	A4601	SuspectParameterNumber, <a href="#">160</a>
A4567	PowerPortConversionInfoSequence, <a href="#">110</a>	A4602	SwitchFeature, <a href="#">102</a>
A4568	PowerPortDirectionality, <a href="#">99</a>	A4603	SwitchFeatureSequence, <a href="#">102</a>
A4569	PowerPortId, <a href="#">210</a>	A4604	TapNumber, <a href="#">111</a>
A4570	PowerPortInfo, <a href="#">97</a>	A4605	ThermalInfo, <a href="#">89</a>
A4571	PowerPortInfoSequence, <a href="#">96</a>	A4606	ThermalLoadSequence, <a href="#">134</a>
A4572	PowerPortIntent, <a href="#">155</a>	A4607	ThermalState, <a href="#">133</a>
A4573	PowerPortIntentSequence, <a href="#">155</a>	A4608	ThermalZoneSequence, <a href="#">89</a>
A4574	PowerPortNumber, <a href="#">69</a>	A4609	TopicConditionSequence, <a href="#">82</a>
A4575	PowerPortNumberSequence, <a href="#">105</a>	A4610	TopicInfo, <a href="#">82</a>
A4576	PowerPortStatus, <a href="#">208</a>	A4611	TopicList, <a href="#">181</a>
A4577	PowerPortStatusSequence, <a href="#">207</a>	A4612	TopicName, <a href="#">181</a>
A4578	PowerSwitchCommand, <a href="#">170</a>	A4613	TransformerInfo, <a href="#">89</a>
A4579	PowerSwitchCommandSequence, <a href="#">169</a>	A4614	TransformerState, <a href="#">133</a>
A4580	PowerSwitchReason, <a href="#">199</a>		
A4581	PowerSwitchStatus, <a href="#">198</a>		

A4615 B.30.3 Data Types in Namespace tms.ac

A4616 DevicePowerMeasurements, [213](#)

A4617 LoadSharing, [178](#)

A4618 LoadSharingRequest, [178](#)

A4619 LoadSharingStatus, [229](#)

A4620 PhaseAnglesSequence, [219](#)

A4621 PowerMeasurement, [214](#)

A4622 PowerMeasurementLine, [216](#)

A4623 PowerMeasurementLineSequence, [216](#)

A4624 PowerPortMeasurement, [214](#)

A4625 PowerPortMeasurementSequence, [213](#)

A4626 B.30.4 Data Types in Namespace tms.dc

A4627 DevicePowerMeasurements, [220](#)

A4628 LoadSharing, [182](#)

A4629 LoadSharingRequest, [181](#)

A4630 LoadSharingStatus, [232](#)

A4631 PowerMeasurement, [222](#)

A4632 PowerPortMeasurement, [221](#)

A4633 PowerPortMeasurementSequence, [221](#)

A4634 B.30.5 Defined Constants

A4635	CLOCK_ATOMIC, <a href="#">237</a>	A4654	MAX_TAPS, <a href="#">111</a>
A4636	CLOCK_GPS, <a href="#">237</a>	A4655	MAXLEN_storageCells, <a href="#">142</a>
A4637	CLOCK_HAND, <a href="#">237</a>		
A4638	CLOCK_INTERNAL, <a href="#">237</a>	A4656	ONLY_CELL, <a href="#">142</a>
A4639	CLOCK_NTP, <a href="#">237</a>	A4657	ONLY_PORT, <a href="#">69</a>
A4640	CLOCK_OTHER, <a href="#">237</a>		
A4641	CLOCK_PTP, <a href="#">237</a>	A4658	REPLY_BAD_REQUEST, <a href="#">71</a>
A4642	CLOCK_RADIO, <a href="#">237</a>	A4659	REPLY_CONFLICT, <a href="#">71</a>
		A4660	REPLY_GONE, <a href="#">71</a>
A4643	FMI_HIGH_LEAST_SEVERE, <a href="#">160</a>	A4661	REPLY_INTERNAL_SERVER_ERROR,
A4644	FMI_HIGH_MODERATELY_SEVERE,	A4662	<a href="#">71</a>
A4645	<a href="#">160</a>	A4663	REPLY_METHOD_NOT_ALLOWED, <a href="#">71</a>
A4646	FMI_HIGH_MOST_SEVERE, <a href="#">160</a>	A4664	REPLY_NOT_IMPLEMENTED, <a href="#">71</a>
A4647	FMI_LOW_LEAST_SEVERE, <a href="#">160</a>	A4665	REPLY_OK, <a href="#">71</a>
A4648	FMI_LOW_MODERATELY_SEVERE,	A4666	REPLY_PENDING_AUTHORIZATION,
A4649	<a href="#">160</a>	A4667	<a href="#">71</a>
A4650	FMI_LOW_MOST_SEVERE, <a href="#">160</a>	A4668	REPLY_PRECONDITION_FAILED, <a href="#">71</a>
		A4669	REPLY_REQUEST_ENTITY_TOO_LARGE,
A4651	MAX_GROUNDS, <a href="#">104</a>	A4670	<a href="#">71</a>
A4652	MAX_PARAMETERS, <a href="#">92</a>	A4671	REPLY_SERVICE_UNAVAILABLE, <a href="#">71</a>
A4653	MAX_PORTS, <a href="#">69</a>		

A4672 C. Cybersecurity Requirements

A4673 This section intentionally left blank.

## D. OMG DDS Requirements

The following Object Management Group standards are mandatory for all OMG DDS implementations of the data model.

Table D.1: Required DDS Standards

Name	Acronym	Version	Adoption Date
Data Distribution Service	DDS	1.4	March 2015
Interface Definition Language	IDL	4.2	March 2018
Extensible and Dynamic Topic Types for DDS	DDS-XTypes	1.3	February 2020
DDS Interoperability Wire Protocol	DDSI-RTPS	2.3	May 2019

The following Object Management Group standard is mandatory for security-enabled OMG DDS implementations of the data model.

Table D.2: DDS Security Standards

Name	Acronym	Version	Adoption Date
DDS Security	DDS-SECURITY	1.1	July 2018

OMG DDS products based on older versions of these standards may be compatible. The following sections explain how the data model is implemented using OMG DDS and what modifications may be made to support older implementations.

### D.1 Domains

All OMG DDS participants shall connect to domain ID 0 (zero) by default. A single domain participant may be used to create data readers and writers for all topics.

Other domain IDs may be used for development or future deployments. Any TMS device that can use a domain other than 0 shall provide a means for the operator to view and change the domain ID, without requiring external hardware or software.

Incorrect domain ID selection could prevent devices from communicating, even though they are on the same physical network.

### D.2 Topics

OMG DDS participants shall use the topic names specified in Chapter [B](#).

### A4692 D.3 IDL

A4693 The OMG Interface Definition Language (IDL) is used to provide a machine-readable  
A4694 definition of the data types defined by this standard. OMG DDS participants shall use  
A4695 the IDL definitions provided in Chapter E and described in Chapter B.

#### A4696 D.3.1 Data Types

A4697 OMG IDL has evolved over many years, and newer versions provide better ways of modeling  
A4698 data. For example, integer type names now clearly describe their size, and there is new  
A4699 support for optional structure attributes. The following table shows how types map from  
A4700 the TMS data model to IDL version 3 and IDL version 4.2.

Table D.3: IDL Type Substitutions

Data Model	IDL 3	IDL 4.2
boolean	boolean	boolean
char	char	char
octet	octet	octet
int8	char	int8
int16	short	int16
int32	long	int32
int64	long long	int64
uint8	octet	uint8
uint16	unsigned short	uint16
uint32	unsigned long	uint32
uint64	unsigned long long	uint64
float32	float	float
float64	double	double
Enumeration	enum	enum
Constant	const	const
Structure	struct	struct
<b>sequence</b> <T,min,max>	sequence<T,max>	sequence<T,max>
<b>sequence</b> <char,min,max>	string<max>	string<max>
<b>optional</b> <T>	sequence<T,1>	@optional T
Typedef	typedef	typedef

A4701 The official TMS IDL file will be released to target official IDL 4.2 syntax. Support for  
A4702 IDL 3 or non-standard DDS vendor extensions will be provided on a best effort basis. See  
A4703 accompanying technical package TBD for tools that may assist in this task. Any modified  
A4704 IDL must be interoperable with the official release file. Note that some changes, like the use  
A4705 of “@optional” are not backwards-compatible with their IDL 3 counterparts.

A4706 D.3.2 Annotations

A4707 OMG IDL provides a number of options for annotating data types. This section describes  
A4708 the annotations used by the standard TMS IDL file. It also describes what modifications  
A4709 may be made to support DDS implementations that do not support them.

A4710 D.3.2.1 Extensibility

A4711 OMG IDL provides multiple options for annotating types. The official TMS IDL file will  
A4712 contain the following annotations.

Table D.4: IDL Annotations for Extensibility

Annotation	Meaning
<code>@extensibility(FINAL)</code>	No changes
<code>@extensibility(APPENDABLE)</code>	Values may be appended
<code>@extensibility(MUTABLE)</code>	Values may be appended or modified

A4713 These annotations change the DDS serialization and must not be removed. Conformance  
A4714 testing will verify that devices interopate with appendable and mutable changes.

A4715 D.3.2.2 Topic Usage

A4716 IDL annotations are used to indicate whether each data type is used directly or indirectly  
A4717 by one or more topics.

Table D.5: IDL Annotations for Topic Usage

Annotation	Meaning
<code>@topic("Foo")</code>	Type is assigned to topic <b>Foo</b> .
<code>@nested</code>	Type is used indirectly by one or more topics.

A4718 It is expected that devices may remove these annotation with no impact to  
A4719 interoperability.

A4720 Note: The `@topic` annotation does not appear in IDL version 4.2. It is defined in DDS-  
A4721 XTypes, version 1.3, clause 7.3.1.2.1.16.

A4722 D.3.2.3 Key Values

A4723 The IDL `@key` annotation is used to indicate key values.

A4724 This annotation may be converted into equivalent vendor-specific syntax. Key values  
A4725 should not be removed entirely, as this may increase the DDS network traffic. Conformance  
A4726 testing will verify interoperability with devices that do use key values.

#### A4727 D.3.2.4 Bounded Values

A4728 IDL annotations are used to indicate when valid values are bounded by less than the full  
A4729 numeric range. The annotation `@min` indicates a lower bound, `@max` indicates an upper  
A4730 bound, and `@range` indicates both bounds.

Table D.6: IDL Annotations for Bounded Values

Annotation	Meaning
<code>@min(k1)</code>	$k_1 \leq X$
<code>@max(k2)</code>	$X \leq k_2$
<code>@range(min=k1, max=k2)</code>	$k_1 \leq X \leq k_2$

A4731 It is expected that devices may remove these annotation with no impact to  
A4732 interoperability. Conformance testing will verify that devices publish values within this  
A4733 range, and subscribers flag out-of-range values, but not whether these annotations are used.

#### A4734 D.3.2.5 Units of Measure

A4735 The IDL `@unit` annotation is used to indicate the units of measure for numeric values.

A4736 *Note: The IDL standard recommends using standardized abbreviations defined by BIPM*  
A4737 *whenever applicable.*

A4738 It is expected that devices may remove this annotation with no impact to interoperability.  
A4739 Conformance testing will verify that device use the proper units of measurement but not  
A4740 whether this annotation is used.

### A4741 D.4 Quality of Service

A4742 All devices shall use the following Quality of Service (QoS) profile definitions. These specify  
A4743 values for all QoS policies in the OMG DDS standard. Procurement documents may specify  
A4744 additional implementation-specific values.

A4745 Conformance testing will verify that devices under test interoperate with other devices  
A4746 using the given values.

A4747 Devices may use the OMG DDS XML definitions provided in Section [E.2](#).



A4748 D.4.1 PublishLast Profile

Table D.7: OMG DDS QoS Policies for the PublishLast Profile

QoS Policy	Notes	Value
USER_DATA	default	zero-length sequence
TOPIC_DATA	default	zero-length sequence
GROUP_DATA	default	zero-length sequence
DURABILITY	default, 1	TRANSIENT_LOCAL
DURABILITY_SERVICE	2	N/A
PRESENTATION	default	access_scope=INSTANCE
	default	coherent_access=false
	default	ordered_access=false
DEADLINE	default	INFINITE_SEC, INFINITE_NSEC
LATENCY_BUDGET	default	0
OWNERSHIP		EXCLUSIVE
OWNERSHIP_STRENGTH		1
LIVELINESS	default, 3	kind=AUTOMATIC
	default, 3	lease_duration=INFINITE_SEC, INFINITE_NSEC
TIME_BASED_FILTER	default, 4	0
PARTITION	default	zero-length sequence
RELIABILITY	5	kind=RELIABLE
	6	max_blocking_time = application specific
TRANSPORT_PRIORITY	default	0
LIFESPAN	default	INFINITE_SEC, INFINITE_NSEC
DESTINATION_ORDER	default	BY_RECEPTION_TIMESTAMP
HISTORY	default	kind=KEEP_LAST
	default	depth=1
RESOURCE_LIMITS	default, 7	max_samples = UNLIMITED
	default, 7	max_instances = UNLIMITED
	default, 7	max_samples_per_instance = UNLIMITED
ENTITY_FACTORY	6	application specific
WRITER_DATA_LIFECYCLE	6	application specific, default TRUE
READER_DATA_LIFECYCLE	6	application specific

A4749 NOTES:

- A4750 0. Default. This is the default value in the OMG DDS standard. Set it explicitly or verify  
A4751 that it is the default in your implementation.
- A4752 1. DURABILITY. It is expected that the DataWriter lifespan will match the operating  
A4753 time of the device. DataWriters should not be created and destroyed for each sample  
A4754 update.
- A4755 2. DURABILITY\_SERVICE. This value is not used with the selected DURABILITY.

- A4756 3. LIVELINESS. The publisher may choose to use a MANUAL kind and a shorter lease  
A4757 duration without affecting interoperability.
- A4758 4. TIME\_BASED\_FILTER. The subscriber may choose a longer minimum separation  
A4759 without affecting interoperability.
- A4760 5. RELIABILITY kind. Subscribers may select BEST\_EFFORT under some  
A4761 circumstances. For example, an MD may not require reliable delivery of power  
A4762 measurements.
- A4763 6. RELIABILITY max blocking time, FACTORY, and LIFECYCLE. These values are  
A4764 application-specific and may be changed without affecting interoperability.
- A4765 7. RESOURCE\_LIMITS. These values may be decreased by the publisher or subscriber,  
A4766 subject to the requirements in Section [A.10](#).

A4767 D.4.2 Rare Profile

Table D.8: OMG DDS QoS Policies for the Rare Profile

QoS Policy	Notes	Value
USER_DATA	default	zero-length sequence
TOPIC_DATA	default	zero-length sequence
GROUP_DATA	default	zero-length sequence
DURABILITY	1	VOLATILE
DURABILITY_SERVICE	2	N/A
PRESENTATION	default	access_scope=INSTANCE
	default	coherent_access=false
	default	ordered_access=false
DEADLINE		2000 sec, 0 nanosec
LATENCY_BUDGET	default	0
OWNERSHIP		EXCLUSIVE
OWNERSHIP_STRENGTH		1
LIVELINESS	default, 3	kind=AUTOMATIC
	default, 3	lease_duration=INFINITE_SEC, INFINITE_NSEC
TIME_BASED_FILTER	default, 4	0
PARTITION	default	zero-length sequence
RELIABILITY	5	kind=BEST_EFFORT
	6	max_blocking_time = application specific
TRANSPORT_PRIORITY	default	0
LIFESPAN	default	INFINITE_SEC, INFINITE_NSEC
DESTINATION_ORDER	default	BY_RECEPTION_TIMESTAMP
HISTORY	default	kind=KEEP_LAST
	default	depth=1
RESOURCE_LIMITS	default, 7	max_samples = UNLIMITED
	default, 7	max_instances = UNLIMITED
	default, 7	max_samples_per_instance = UNLIMITED
ENTITY_FACTORY	6	application specific
WRITER_DATA_LIFECYCLE	6	application specific, default TRUE
READER_DATA_LIFECYCLE	6	application specific

A4768 NOTES:

- A4769 0. Default. This is the default value in the OMG DDS standard. Set it explicitly or verify  
A4770 that it is the default in your implementation.
- A4771 1. DURABILITY. It is expected that the DataWriter lifespan will match the operating  
A4772 time of the device. DataWriters should not be created and destroyed for each sample  
A4773 update.
- A4774 2. DURABILITY\_SERVICE. This value is not used with the selected DURABILITY.

- A4775 3. LIVELINESS. The publisher may choose to use a MANUAL kind and a shorter lease  
A4776 duration without affecting interoperability.
- A4777 4. TIME\_BASED\_FILTER. The subscriber may choose a longer minimum separation  
A4778 without affecting interoperability.
- A4779 5. RELIABILITY kind. Publishers should not offer RELIABLE, as this may delay  
A4780 delivery of new sample updates.
- A4781 6. RELIABILITY max blocking time, FACTORY, and LIFECYCLE. These values are  
A4782 application-specific and may be changed without affecting interoperability.
- A4783 7. RESOURCE\_LIMITS. These values may be decreased by the publisher or subscriber,  
A4784 subject to the requirements in Section [A.10](#).

A4785 D.4.3 Slow Profile

Table D.9: OMG DDS QoS Policies for the Slow Profile

QoS Policy	Notes	Value
USER_DATA	default	zero-length sequence
TOPIC_DATA	default	zero-length sequence
GROUP_DATA	default	zero-length sequence
DURABILITY	1	VOLATILE
DURABILITY_SERVICE	2	N/A
PRESENTATION	default	access_scope=INSTANCE
	default	coherent_access=false
	default	ordered_access=false
DEADLINE		20 sec, 0 nanosec
LATENCY_BUDGET	default	0
OWNERSHIP		EXCLUSIVE
OWNERSHIP_STRENGTH		1
LIVELINESS	default, 3	kind=AUTOMATIC
	default, 3	lease_duration=INFINITE_SEC, INFINITE_NSEC
TIME_BASED_FILTER	default, 4	0
PARTITION	default	zero-length sequence
RELIABILITY	5	kind=BEST_EFFORT
	6	max_blocking_time = application specific
TRANSPORT_PRIORITY	default	0
LIFESPAN	default	INFINITE_SEC, INFINITE_NSEC
DESTINATION_ORDER	default	BY_RECEPTION_TIMESTAMP
HISTORY	default	kind=KEEP_LAST
	default	depth=1
RESOURCE_LIMITS	default, 7	max_samples = UNLIMITED
	default, 7	max_instances = UNLIMITED
	default, 7	max_samples_per_instance = UNLIMITED
ENTITY_FACTORY	6	application specific
WRITER_DATA_LIFECYCLE	6	application specific, default TRUE
READER_DATA_LIFECYCLE	6	application specific

A4786 NOTES:

- A4787 0. Default. This is the default value in the OMG DDS standard. Set it explicitly or verify  
A4788 that it is the default in your implementation.
- A4789 1. DURABILITY. It is expected that the DataWriter lifespan will match the operating  
A4790 time of the device. DataWriters should not be created and destroyed for each sample  
A4791 update.
- A4792 2. DURABILITY\_SERVICE. This value is not used with the selected DURABILITY.

- A4793 3. LIVELINESS. The publisher may choose to use a MANUAL kind and a shorter lease  
A4794 duration without affecting interoperability.
- A4795 4. TIME\_BASED\_FILTER. The subscriber may choose a longer minimum separation  
A4796 without affecting interoperability.
- A4797 5. RELIABILITY kind. Publishers should not offer RELIABLE, as this may delay  
A4798 delivery of new sample updates.
- A4799 6. RELIABILITY max blocking time, FACTORY, and LIFECYCLE. These values are  
A4800 application-specific and may be changed without affecting interoperability.
- A4801 7. RESOURCE\_LIMITS. These values may be decreased by the publisher or subscriber,  
A4802 subject to the requirements in Section [A.10](#).

A4803 D.4.4 Medium Profile

Table D.10: OMG DDS QoS Policies for the Medium Profile

QoS Policy	Notes	Value
USER_DATA	default	zero-length sequence
TOPIC_DATA	default	zero-length sequence
GROUP_DATA	default	zero-length sequence
DURABILITY	1	VOLATILE
DURABILITY_SERVICE	2	N/A
PRESENTATION	default	access_scope=INSTANCE
	default	coherent_access=false
	default	ordered_access=false
DEADLINE		3 sec, 0 nanosec
LATENCY_BUDGET	default	0
OWNERSHIP		EXCLUSIVE
OWNERSHIP_STRENGTH		1
LIVELINESS	default, 3	kind=AUTOMATIC
	default, 3	lease_duration=INFINITE_SEC, INFINITE_NSEC
TIME_BASED_FILTER	default, 4	0
PARTITION	default	zero-length sequence
RELIABILITY	5	kind=BEST_EFFORT
	6	max_blocking_time = application specific
TRANSPORT_PRIORITY	default	0
LIFESPAN	default	INFINITE_SEC, INFINITE_NSEC
DESTINATION_ORDER	default	BY_RECEPTION_TIMESTAMP
HISTORY	default	kind=KEEP_LAST
	default	depth=1
RESOURCE_LIMITS	default, 7	max_samples = UNLIMITED
	default, 7	max_instances = UNLIMITED
	default, 7	max_samples_per_instance = UNLIMITED
ENTITY_FACTORY	6	application specific
WRITER_DATA_LIFECYCLE	6	application specific, default TRUE
READER_DATA_LIFECYCLE	6	application specific

A4804 NOTES:

- A4805 0. Default. This is the default value in the OMG DDS standard. Set it explicitly or verify  
A4806 that it is the default in your implementation.
- A4807 1. DURABILITY. It is expected that the DataWriter lifespan will match the operating  
A4808 time of the device. DataWriters should not be created and destroyed for each sample  
A4809 update.
- A4810 2. DURABILITY\_SERVICE. This value is not used with the selected DURABILITY.

- A4811 3. LIVELINESS. The publisher may choose to use a MANUAL kind and a shorter lease  
A4812 duration without affecting interoperability.
- A4813 4. TIME\_BASED\_FILTER. The subscriber may choose a longer minimum separation  
A4814 without affecting interoperability.
- A4815 5. RELIABILITY kind. Publishers should not offer RELIABLE, as this may delay  
A4816 delivery of new sample updates.
- A4817 6. RELIABILITY max blocking time, FACTORY, and LIFECYCLE. These values are  
A4818 application-specific and may be changed without affecting interoperability.
- A4819 7. RESOURCE\_LIMITS. These values may be decreased by the publisher or subscriber,  
A4820 subject to the requirements in Section [A.10](#).



A4821 D.4.5 Continuous Profile

Table D.11: OMG DDS QoS Policies for the Continuous Profile

QoS Policy	Notes	Value
USER_DATA	default	zero-length sequence
TOPIC_DATA	default	zero-length sequence
GROUP_DATA	default	zero-length sequence
DURABILITY	1	VOLATILE
DURABILITY_SERVICE	2	N/A
PRESENTATION	default	access_scope=INSTANCE
	default	coherent_access=false
	default	ordered_access=false
DEADLINE		2 sec, 0 nanosec
LATENCY_BUDGET	default	0
OWNERSHIP		EXCLUSIVE
OWNERSHIP_STRENGTH		1
LIVELINESS	default, 3	kind=AUTOMATIC
	default, 3	lease_duration=INFINITE_SEC, INFINITE_NSEC
TIME_BASED_FILTER	default, 4	0
PARTITION	default	zero-length sequence
RELIABILITY	5	kind=BEST_EFFORT
	6	max_blocking_time = application specific
TRANSPORT_PRIORITY	default	0
LIFESPAN	default	INFINITE_SEC, INFINITE_NSEC
DESTINATION_ORDER	default	BY_RECEPTION_TIMESTAMP
HISTORY	default	kind=KEEP_LAST
	default	depth=1
RESOURCE_LIMITS	default, 7	max_samples = UNLIMITED
	default, 7	max_instances = UNLIMITED
	default, 7	max_samples_per_instance = UNLIMITED
ENTITY_FACTORY	6	application specific
WRITER_DATA_LIFECYCLE	6	application specific, default TRUE
READER_DATA_LIFECYCLE	6	application specific

A4822 NOTES:

- A4823 0. Default. This is the default value in the OMG DDS standard. Set it explicitly or verify  
A4824 that it is the default in your implementation.
- A4825 1. DURABILITY. It is expected that the DataWriter lifespan will match the operating  
A4826 time of the device. DataWriters should not be created and destroyed for each sample  
A4827 update.
- A4828 2. DURABILITY\_SERVICE. This value is not used with the selected DURABILITY.

- A4829 3. LIVELINESS. The publisher may choose to use a MANUAL kind and a shorter lease  
A4830 duration without affecting interoperability.
- A4831 4. TIME\_BASED\_FILTER. The subscriber may choose a longer minimum separation  
A4832 without affecting interoperability.
- A4833 5. RELIABILITY kind. Publishers should not offer RELIABLE, as this may delay  
A4834 delivery of new sample updates.
- A4835 6. RELIABILITY max blocking time, FACTORY, and LIFECYCLE. These values are  
A4836 application-specific and may be changed without affecting interoperability.
- A4837 7. RESOURCE\_LIMITS. These values may be decreased by the publisher or subscriber,  
A4838 subject to the requirements in Section [A.10](#).

A4839 D.4.6 Command Profile

Table D.12: OMG DDS QoS Policies for the Command Profile

QoS Policy	Notes	Value
USER_DATA	default	zero-length sequence
TOPIC_DATA	default	zero-length sequence
GROUP_DATA	default	zero-length sequence
DURABILITY	1	VOLATILE
DURABILITY_SERVICE	2	N/A
PRESENTATION	default	access_scope=INSTANCE
	default	coherent_access=false
	default	ordered_access=false
DEADLINE	default	INFINITE_SEC, INFINITE_NSEC
LATENCY_BUDGET	default	0
OWNERSHIP		EXCLUSIVE
OWNERSHIP_STRENGTH		1
LIVELINESS	default, 3	kind=AUTOMATIC
	default, 3	lease_duration=INFINITE_SEC, INFINITE_NSEC
TIME_BASED_FILTER	default, 4	0
PARTITION	default	zero-length sequence
RELIABILITY	5	kind=RELIABLE
	6	max_blocking_time = application specific
TRANSPORT_PRIORITY	default	0
LIFESPAN	default	INFINITE_SEC, INFINITE_NSEC
DESTINATION_ORDER	default	BY_RECEPTION_TIMESTAMP
HISTORY	default	kind=KEEP_LAST
	default	depth=1
RESOURCE_LIMITS	default, 7	max_samples = UNLIMITED
	default, 7	max_instances = UNLIMITED
	default, 7	max_samples_per_instance = UNLIMITED
ENTITY_FACTORY	6	application specific
WRITER_DATA_LIFECYCLE	6	application specific, default TRUE
READER_DATA_LIFECYCLE	6	application specific

A4840 NOTES:

- A4841 0. Default. This is the default value in the OMG DDS standard. Set it explicitly or verify  
A4842 that it is the default in your implementation.
- A4843 1. DURABILITY. It is expected that the DataWriter lifespan will match the operating  
A4844 time of the device. DataWriters should not be created and destroyed for each sample  
A4845 update.
- A4846 2. DURABILITY\_SERVICE. This value is not used with the selected DURABILITY.

- A4847 3. LIVELINESS. The publisher may choose to use a MANUAL kind and a shorter lease  
A4848 duration without affecting interoperability.
- A4849 4. TIME\_BASED\_FILTER. The subscriber may choose a longer minimum separation  
A4850 without affecting interoperability.
- A4851 5. RELIABILITY kind. Subscribers may select BEST\_EFFORT under some  
A4852 circumstances. For example, an MD may not require reliable delivery of power  
A4853 measurements.
- A4854 6. RELIABILITY max blocking time, FACTORY, and LIFECYCLE. These values are  
A4855 application-specific and may be changed without affecting interoperability.
- A4856 7. RESOURCE\_LIMITS. These values may be decreased by the publisher or subscriber,  
A4857 subject to the requirements in Section [A.10](#).

A4858 D.4.7 Response Profile

Table D.13: OMG DDS QoS Policies for the Response Profile

QoS Policy	Notes	Value
USER_DATA	default	zero-length sequence
TOPIC_DATA	default	zero-length sequence
GROUP_DATA	default	zero-length sequence
DURABILITY	1	VOLATILE
DURABILITY_SERVICE	2	N/A
PRESENTATION	default	access_scope=INSTANCE
	default	coherent_access=false
	default	ordered_access=false
DEADLINE	default	INFINITE_SEC, INFINITE_NSEC
LATENCY_BUDGET	default	0
OWNERSHIP		EXCLUSIVE
OWNERSHIP_STRENGTH		1
LIVELINESS	default, 3	kind=AUTOMATIC
	default, 3	lease_duration=INFINITE_SEC, INFINITE_NSEC
TIME_BASED_FILTER	default, 4	0
PARTITION	default	zero-length sequence
RELIABILITY	5	kind=RELIABLE
	6	max_blocking_time = application specific
TRANSPORT_PRIORITY	default	0
LIFESPAN	default	INFINITE_SEC, INFINITE_NSEC
DESTINATION_ORDER	default	BY_RECEPTION_TIMESTAMP
HISTORY	default	kind=KEEP_LAST
	default	depth=1
RESOURCE_LIMITS	default, 7	max_samples = UNLIMITED
	default, 7	max_instances = UNLIMITED
	default, 7	max_samples_per_instance = UNLIMITED
ENTITY_FACTORY	6	application specific
WRITER_DATA_LIFECYCLE	6	application specific, default TRUE
READER_DATA_LIFECYCLE	6	application specific

A4859 NOTES:

- A4860 0. Default. This is the default value in the OMG DDS standard. Set it explicitly or verify  
A4861 that it is the default in your implementation.
- A4862 1. DURABILITY. It is expected that the DataWriter lifespan will match the operating  
A4863 time of the device. DataWriters should not be created and destroyed for each sample  
A4864 update.
- A4865 2. DURABILITY\_SERVICE. This value is not used with the selected DURABILITY.

- A4866 3. LIVELINESS. The publisher may choose to use a MANUAL kind and a shorter lease  
A4867 duration without affecting interoperability.
- A4868 4. TIME\_BASED\_FILTER. The subscriber may choose a longer minimum separation  
A4869 without affecting interoperability.
- A4870 5. RELIABILITY kind. Subscribers may select BEST\_EFFORT under some  
A4871 circumstances. For example, an MD may not require reliable delivery of power  
A4872 measurements.
- A4873 6. RELIABILITY max blocking time, FACTORY, and LIFECYCLE. These values are  
A4874 application-specific and may be changed without affecting interoperability.
- A4875 7. RESOURCE\_LIMITS. These values may be decreased by the publisher or subscriber,  
A4876 subject to the requirements in Section [A.10](#).

A4877 D.5 DDS-Security

A4878 Intentionally left blank.

A4879 *Editorial note: Cybersecurity requirements will be added in a later draft. They will likely*  
A4880 *define multiple levels of security, from no protections, to a baseline use of DDS-Security, to*  
A4881 *future extensions.*

## E. Machine Readable Files

### E.1 IDL

The following is a syntax-highlighted, verbatim copy of the standard TMS IDL file. This file is normative.

Automatic line wraps are indicated by a  $\leftarrow$ . A separate, plain-text IDL file should be distributed with the standard.

```
/**
 * Tactical Microgrid Specification (TMS)
 * Distribution Statement: TBD
 * OMG DDS Interface Definition Language configuration for TMS as distributed as part of MIL-STD XXXX
 */
module tms {

    /** TMS Topic Names */
    module topic {

        /**
         * Liveliness heartbeat for all devices.
         * Version: R1
         * Section: DISCOVERY
         * QoS Profile: Medium
         * Publish Rate: Periodic 1 s
         * Data Type: tms.Heartbeat
         */
        const string TOPIC_HEARTBEAT = "Heartbeat";

        /**
         * Product information for the device. Used to discover TMS compliant devices.
         * Version: R1
         * Section: DISCOVERY
         * QoS Profile: PublishLast
         * Publish Rate: At Start
         * Data Type: tms.DeviceInfo
         */
        const string TOPIC_DEVICE_ANNOUNCEMENT = "DeviceAnnouncement";

        /**
         * Device image as an icon.
         * Version: R1
         * Section: DISCOVERY
         * QoS Profile: PublishLast
         * Publish Rate: At Start
         * Data Type: tms.DeviceIcon
         */
        const string TOPIC_DEVICE_ICON = "DeviceIcon";

        /**
         * Device and Platform Fingerprint Nickname.
         * Version: R1
         * Section: DISCOVERY
         * QoS Profile: PublishLast
         * Publish Rate: At Start, On Change
         * Data Type: tms.FingerprintNickname
         */
        const string TOPIC_FINGERPRINT_NICKNAME = "FingerprintNickname";

        /**
         * Request to change a Device or Platform Fingerprint Nickname.
         * Version: R1
         * Section: DISCOVERY
         * QoS Profile: Command
         * Publish Rate: As Needed
         * Data Type: tms.ChangeNicknameRequest
         * Reply Topic: Reply
         * Reply Topic: FingerprintNickname
         */
    }
}
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A4951 const string TOPIC_FINGERPRINT_NICKNAME_REQUEST = "FingerprintNicknameRequest";
A4952
A4953 /**
A4954  * Power connections manually entered by a microgrid operator.
A4955  * Version: R1
A4956  * Section: TOPOLOGY
A4957  * QoS Profile: PublishLast
A4958  * Publish Rate: At Start, On Change
A4959  * Data Type: tms.PowerConnectionList
A4960  */
A4961 const string TOPIC_OPERATOR_CONNECTION_LIST = "OperatorConnectionList";
A4962
A4963 /**
A4964  * Power connections discovered by a power device, usually through a smart cable or operator entry.
A4965  * Version: R1
A4966  * Section: TOPOLOGY
A4967  * QoS Profile: PublishLast
A4968  * Publish Rate: At Start, On Change
A4969  * Data Type: tms.PowerConnectionList
A4970  */
A4971 const string TOPIC_DISCOVERED_CONNECTION_LIST = "DiscoveredConnectionList";
A4972
A4973 /**
A4974  * Power connections used by the microgrid controller. May include operator connections, discovered connections, ←
A4975  * and other connections inferred by the MC.
A4976  * Version: R1
A4977  * Section: TOPOLOGY
A4978  * QoS Profile: PublishLast
A4979  * Publish Rate: At Start, On Change
A4980  * Data Type: tms.PowerConnectionList
A4981  */
A4982 const string TOPIC_MICROGRID_CONNECTION_LIST = "MicrogridConnectionList";
A4983
A4984 /**
A4985  * Reports a device diagnostics state including information and alarm states.
A4986  * Version: R1
A4987  * Section: DIAGNOSTIC
A4988  * QoS Profile: PublishLast
A4989  * Publish Rate: At Start, On Change
A4990  * Data Type: tms.ActiveDiagnosticMessages
A4991  */
A4992 const string TOPIC_ACTIVE_DIAGNOSTICS = "ActiveDiagnostics";
A4993
A4994 /**
A4995  * Reports changes to a device clock(s). A device must support relative time and can optionally support absolute ←
A4996  * time.
A4997  * Version: R1
A4998  * Section: STATUS
A4999  * QoS Profile: PublishLast
A5000  * Publish Rate: At Start, On Change
A5001  * Data Type: tms.ClockStatus
A5002  */
A5003 const string TOPIC_DEVICE_CLOCK_STATUS = "DeviceClockStatus";
A5004
A5005 /**
A5006  * The selected master Microgrid Controller for a specific power device.
A5007  * Version: R1
A5008  * Section: STATUS
A5009  * QoS Profile: PublishLast
A5010  * Publish Rate: At Start, On Change
A5011  * Data Type: tms.StandardConfigMaster
A5012  */
A5013 const string TOPIC_STANDARD_CONFIG_MASTER = "StandardConfigMaster";
A5014
A5015 /**
A5016  * Reports dynamic power measurements for alternating current power device ports.
A5017  * Version: R1
A5018  * Section: POWER_DEVICE
A5019  * QoS Profile: Continuous
A5020  * Publish Rate: Periodic 0.1 s
A5021  * Data Type: tms.ac.DevicePowerMeasurements
A5022  */
A5023 const string TOPIC_DEVICE_POWER_MEASUREMENT_LIST = "DevicePowerMeasurementList";
A5024
A5025 /**
A5026  * Reports dynamic state changes for a power device ports.
A5027  * Version: R1
A5028  * Section: POWER_DEVICE
A5029  * QoS Profile: PublishLast
A5030  * Publish Rate: At Start, On Change
A5031  * Data Type: tms.DevicePowerPortStatuses
A5032  */
A5033 const string TOPIC_DEVICE_POWER_STATUS_LIST = "DevicePowerStatusList";
A5034
A5035 /**
A5036  * Remote request to change the circuit continuity of a single power port
A5037  * Version: R1
A5038  * Section: DIST_DEVICE
A5039  * QoS Profile: Command
A5040  * Publish Rate: As Needed
A5041  * Data Type: tms.PowerSwitchCommand
A5042  * Reply Topic: Reply
```



*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5043      * Reply Topic: DevicePowerStatusList
A5044      */
A5045      const string TOPIC_POWER_SWITCH_REQUEST = "PowerSwitchRequest";
A5046
A5047      /**
A5048      * Reports the operational state (off, on, running, sourcing) of a Source / Storage device
A5049      * Version: R1
A5050      * Section: SOURCE_DEVICE
A5051      * QoS Profile: PublishLast
A5052      * Publish Rate: At Start, On Change
A5053      * Data Type: tms.SourceTransitionState
A5054      */
A5055      const string TOPIC_SOURCE_TRANSITION_STATE = "SourceTransitionState";
A5056
A5057      /**
A5058      * Remote request to change the operational state of a Source / Storage device.
A5059      * Version: R1
A5060      * Section: SOURCE_DEVICE
A5061      * QoS Profile: Command
A5062      * Publish Rate: As Needed
A5063      * Data Type: tms.SourceTransitionRequest
A5064      * Reply Topic: Reply
A5065      * Reply Topic: SourceTransitionState
A5066      */
A5067      const string TOPIC_SOURCE_TRANSITION_REQUEST = "SourceTransitionRequest";
A5068
A5069      /**
A5070      * Reports the active load sharing status of an alternating current Source / Storage device.
A5071      * Version: R1
A5072      * Section: SOURCE_DEVICE
A5073      * QoS Profile: PublishLast
A5074      * Publish Rate: At Start, On Change
A5075      * Data Type: tms.ac.LoadSharingStatus
A5076      */
A5077      const string TOPIC_LOAD_SHARING_STATUS = "LoadSharingStatus";
A5078
A5079      /**
A5080      * Remote request to change the load sharing status of a Source / Storage device.
A5081      * Version: R1
A5082      * Section: SOURCE_DEVICE
A5083      * QoS Profile: Command
A5084      * Publish Rate: As Needed
A5085      * Data Type: tms.ac.LoadSharingRequest
A5086      * Reply Topic: Reply
A5087      * Reply Topic: LoadSharingStatus
A5088      */
A5089      const string TOPIC_LOAD_SHARING_REQUEST = "LoadSharingRequest";
A5090
A5091      /**
A5092      * Reports dynamic state changes of control device hardware.
A5093      * Version: R1
A5094      * Section: SOURCE_DEVICE
A5095      * QoS Profile: Slow
A5096      * Publish Rate: Periodic 10 s
A5097      * Data Type: tms.ControlHardwareState
A5098      */
A5099      const string TOPIC_CONTROL_HARDWARE_STATUS = "ControlHardwareStatus";
A5100
A5101      /**
A5102      * Reports dynamic state of power hardware components.
A5103      * Version: R1
A5104      * Section: SOURCE_DEVICE
A5105      * QoS Profile: Slow
A5106      * Publish Rate: Periodic 10 s
A5107      * Data Type: tms.PowerHardwareState
A5108      */
A5109      const string TOPIC_POWER_HARDWARE_STATUS = "PowerHardwareStatus";
A5110
A5111      /**
A5112      * The dynamic state of a Storage device.
A5113      * Version: R1
A5114      * Section: STORAGE_DEVICE
A5115      * QoS Profile: Slow
A5116      * Publish Rate: Periodic 10 s
A5117      * Data Type: tms.stor.StorageState
A5118      */
A5119      const string TOPIC_STORAGE_STATE = "StorageState";
A5120
A5121      /**
A5122      * Latest storage control parameters.
A5123      * Version: R1
A5124      * Section: STORAGE_DEVICE
A5125      * QoS Profile: PublishLast
A5126      * Publish Rate: At Start, On Change
A5127      * Data Type: tms.stor.StorageControlStatus
A5128      */
A5129      const string TOPIC_STORAGE_CONTROL_STATUS = "StorageControlStatus";
A5130
A5131      /**
A5132      * Remote request to set the storage control parameters.
A5133      * Version: R1
A5134      * Section: STORAGE_DEVICE
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5135      * QoS Profile: Command
A5136      * Publish Rate: As Needed
A5137      * Data Type: tms.stor.StorageControlRequest
A5138      * Reply Topic: Reply
A5139      * Reply Topic: StorageControlStatus
A5140      */
A5141      const string TOPIC_STORAGE_CONTROL_REQUEST = "StorageControlRequest";
A5142
A5143      /**
A5144      * The reply to a request.
A5145      * Version: R1
A5146      * Section: RESPONSE
A5147      * QoS Profile: Response
A5148      * Publish Rate: Within 0.1 s
A5149      * Data Type: tms.Reply
A5150      */
A5151      const string TOPIC_REQUEST_RESPONSE = "Reply";
A5152
A5153      /**
A5154      * The active power device control parameters for this power device.
A5155      * Version: R1
A5156      * Section: DEVICE_PARAMETERS
A5157      * QoS Profile: PublishLast
A5158      * Publish Rate: At Start, On Change
A5159      * Data Type: tms.ControlParameterStatus
A5160      */
A5161      const string TOPIC_CONTROL_PARAMETER_STATUS = "ControlParameterStatus";
A5162
A5163      /**
A5164      * Remote request to change the device control parameters as defined by the DeviceInfo.controlParameters.
A5165      * Version: R1
A5166      * Section: DEVICE_PARAMETERS
A5167      * QoS Profile: Command
A5168      * Publish Rate: As Needed
A5169      * Data Type: tms.ControlParameterRequest
A5170      * Reply Topic: Reply
A5171      * Reply Topic: ControlParameterStatus
A5172      */
A5173      const string TOPIC_CONTROL_PARAMETER_REQUEST = "ControlParameterRequest";
A5174
A5175      /**
A5176      * The active metric parameter values for this power device.
A5177      * Version: R1
A5178      * Section: DEVICE_PARAMETERS
A5179      * QoS Profile: PublishLast
A5180      * Publish Rate: At Start, On Change
A5181      * Data Type: tms.MetricParameterStatus
A5182      */
A5183      const string TOPIC_METRIC_PARAMETER_STATUS = "MetricParameterStatus";
A5184
A5185      /**
A5186      * Provide the operator's response to an authorization request.
A5187      * Version: R1
A5188      * Section: BLACK_START
A5189      * QoS Profile: Response
A5190      * Publish Rate: As Needed
A5191      * Data Type: tms.AuthorizationToEnergizeResponse
A5192      */
A5193      const string TOPIC_AUTHORIZATION_TO_ENERGIZE_RESPONSE = "AuthorizationToEnergizeResponse";
A5194
A5195      /**
A5196      * Request authorization to execute a remote command when a black start is detected.
A5197      * Version: R1
A5198      * Section: BLACK_START
A5199      * QoS Profile: Command
A5200      * Publish Rate: As Needed
A5201      * Data Type: tms.AuthorizationToEnergizeRequest
A5202      * Reply Topic: Reply
A5203      * Reply Topic: AuthorizationToEnergizeResponse
A5204      */
A5205      const string TOPIC_AUTHORIZATION_TO_ENERGIZE_REQUEST = "AuthorizationToEnergizeRequest";
A5206
A5207      /**
A5208      * Report the outcome of a command that required authorization.
A5209      * Version: R1
A5210      * Section: BLACK_START
A5211      * QoS Profile: Response
A5212      * Publish Rate: Within 0.1 s
A5213      * Data Type: tms.AuthorizationToEnergizeOutcome
A5214      */
A5215      const string TOPIC_AUTHORIZATION_TO_ENERGIZE_OUTCOME = "AuthorizationToEnergizeOutcome";
A5216
A5217      /**
A5218      * The last published OperatorIntent received by a Microgrid Controller
A5219      * Version: R1
A5220      * Section: OPERATOR_INTENT
A5221      * QoS Profile: PublishLast
A5222      * Publish Rate: At Start, On Change
A5223      * Data Type: tms.OperatorIntentState
A5224      */
A5225      const string TOPIC_OPERATOR_INTENT_STATE = "OperatorIntentState";
A5226
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5227  /**
A5228  * Operator defined microgrid control directives that influence how the microgrid behaves.
A5229  * Version: R1
A5230  * Section: OPERATOR_INTENT
A5231  * QoS Profile: Command
A5232  * Publish Rate: As Needed
A5233  * Data Type: tms.OperatorIntentRequest
A5234  * Reply Topic: Reply
A5235  * Reply Topic: OperatorIntentState
A5236  */
A5237  const string TOPIC_OPERATOR_INTENT_REQUEST = "OperatorIntentRequest";
A5238
A5239  /**
A5240  * Response to a GetConfigContentsRequest
A5241  * Version: R1
A5242  * Section: DEVICE_CONFIG
A5243  * QoS Profile: Response
A5244  * Publish Rate: Within 0.1 s
A5245  * Data Type: tms.GetConfigContentsResponse
A5246  */
A5247  const string TOPIC_GET_CONFIG_CONTENTS_RESPONSE = "GetConfigContentsResponse";
A5248
A5249  /**
A5250  * Copy the commands from one device configuration index to another.
A5251  * Version: R1
A5252  * Section: DEVICE_CONFIG
A5253  * QoS Profile: Command
A5254  * Publish Rate: As Needed
A5255  * Data Type: tms.CopyConfigRequest
A5256  * Reply Topic: Reply
A5257  */
A5258  const string TOPIC_COPY_CONFIG_REQUEST = "CopyConfigRequest";
A5259
A5260  /**
A5261  * Retrieve all commands stored in a device configuration.
A5262  * Version: R1
A5263  * Section: DEVICE_CONFIG
A5264  * QoS Profile: Command
A5265  * Publish Rate: As Needed
A5266  * Data Type: tms.GetConfigContentsRequest
A5267  * Reply Topic: GetConfigContentsResponse
A5268  */
A5269  const string TOPIC_GET_CONFIG_CONTENTS_REQUEST = "GetConfigContentsRequest";
A5270
A5271  /**
A5272  * A power device requests to leave or join a grid.
A5273  * Version: R1
A5274  * Section: MICROGRID_MEMBERSHIP
A5275  * QoS Profile: Command
A5276  * Publish Rate: As Needed
A5277  * Data Type: tms.MicrogridMembershipRequest
A5278  * Reply Topic: Reply
A5279  */
A5280  const string TOPIC_MICROGRID_MEMBERSHIP_REQUEST = "MicrogridMembershipRequest";
A5281
A5282  /**
A5283  * Outcome of a Microgrid Membership Request.
A5284  * Version: R1
A5285  * Section: MICROGRID_MEMBERSHIP
A5286  * QoS Profile: Response
A5287  * Publish Rate: As Needed
A5288  * Data Type: tms.MicrogridMembershipApproval
A5289  */
A5290  const string TOPIC_MICROGRID_MEMBERSHIP_OUTCOME = "MicrogridMembershipOutcome";
A5291
A5292  /**
A5293  * Grounding Circuit status.
A5294  * Version: R1
A5295  * Section: GROUNDING_CIRCUIT
A5296  * QoS Profile: PublishLast
A5297  * Publish Rate: At Start, On Change
A5298  * Data Type: tms.DeviceGroundingStatus
A5299  */
A5300  const string TOPIC_DEVICE_GROUNDING_STATUS = "DeviceGroundingStatus";
A5301
A5302  /**
A5303  * Grounding Circuit Command.
A5304  * Version: R1
A5305  * Section: GROUNDING_CIRCUIT
A5306  * QoS Profile: Command
A5307  * Publish Rate: As Needed
A5308  * Data Type: tms.GroundingCommand
A5309  * Reply Topic: Reply
A5310  * Reply Topic: DeviceGroundingStatus
A5311  */
A5312  const string TOPIC_GROUNDING_CIRCUIT_REQUEST = "GroundingCircuitRequest";
A5313
A5314  /**
A5315  * Reports dynamic power measurements for a direct current power device ports.
A5316  * Version: R1
A5317  * Section: DC_POWER_DEVICE
A5318  * QoS Profile: Continuous
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5319      * Publish Rate: Periodic 0.1 s
A5320      * Data Type: tms.dc.DevicePowerMeasurements
A5321      */
A5322      const string TOPIC_DC_DEVICE_POWER_MEASUREMENT_LIST = "DcDevicePowerMeasurementList";
A5323
A5324      /**
A5325       * Reports the active direct current load sharing status of a device.
A5326       * Version: R1
A5327       * Section: DC_POWER_DEVICE
A5328       * QoS Profile: PublishLast
A5329       * Publish Rate: At Start, On Change
A5330       * Data Type: tms.dc.LoadSharingStatus
A5331       */
A5332      const string TOPIC_DC_LOAD_SHARING_STATUS = "DcLoadSharingStatus";
A5333
A5334      /**
A5335       * Remote request to change the DC load sharing status of a device.
A5336       * Version: R1
A5337       * Section: DC_POWER_DEVICE
A5338       * QoS Profile: Command
A5339       * Publish Rate: As Needed
A5340       * Data Type: tms.dc.LoadSharingRequest
A5341       * Reply Topic: Reply
A5342       * Reply Topic: DcLoadSharingStatus
A5343       */
A5344      const string TOPIC_DC_LOAD_SHARING_REQUEST = "DcLoadSharingRequest";
A5345      }; //end module topic
A5346
A5347      const string TMS_VERSION = "2.0";
A5348
A5349
A5350
A5351      const unsigned long TopicName_MINLEN = 1;
A5352      const unsigned long TopicName_MAXLEN = 64;
A5353      /**
A5354       * Specifies a communications data topic.
A5355       * Minimum length: 1
A5356       * Maximum length: 64
A5357       */
A5358      typedef string <TopicName_MAXLEN> TopicName;
A5359
A5360
A5361      const unsigned long TopicList_MINLEN = 0;
A5362      const unsigned long TopicList_MAXLEN = 8;
A5363      /**
A5364       * Specifies a list of communications data topics.
A5365       * Minimum length: 0
A5366       * Maximum length: 8
A5367       */
A5368      typedef sequence<TopicName, TopicList_MAXLEN> TopicList;
A5369
A5370
A5371      @extensibility(APPENDABLE)
A5372      enum CableSenseStatus {
A5373          CS_UNKNOWN, // Cable presence is neither confirmed nor denied through active detection. Cable sense is not ←
A5374                      // available, either due to device design or failure.
A5375          CS_DISCONNECTED, // Sensing indicates that no cable is not connected. CST_MEASUREMENT devices must use CS_UNKNOWN ←
A5376                      // instead of this value.
A5377          CS_CONNECTED // Sensing indicates that a cable is connected.
A5378      }; // enum CableSenseStatus
A5379
A5380
A5381      @extensibility(APPENDABLE)
A5382      enum ConnectorFeature {
A5383          CF_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ←
A5384                      // could not be determined.
A5385          CF_CABLE_SENSE, // A continuity circuit or similar feature detects when a cable is connected.
A5386          CF_CABLE_ID_READER, // A cable ID reader is present.
A5387          CF_CABLE_MEASUREMENT, // Voltage and current monitoring can establish the presence of a cable.
A5388          CF_COMMUNICATION, // Contains network interface.
A5389          CF_TOPOLOGY_DISCOVERY // Ability to discover connectivity to remote device power ports.
A5390      }; // enum ConnectorFeature
A5391
A5392
A5393      @extensibility(APPENDABLE)
A5394      enum CircuitContinuity {
A5395          CC_UNKNOWN, // No information is available, either due to design or malfunction.
A5396          CC_OPEN, // Electrical or mechanical sensors indicate that continuity has been broken.
A5397          CC_CLOSED, // Electrical or mechanical sensors indicate that continuity has been established.
A5398          CC_INCONSISTENT // Electrical or mechanical sensors indicate that continuity has been established on some phases ←
A5399                      // and broken on other phases.
A5400      }; // enum CircuitContinuity
A5401
A5402
A5403      @extensibility(APPENDABLE)
A5404      enum ClockMonotonicPersistenceType {
A5405          CMPT_UNKNOWN, // Fault detection indicates that initialization may be inconsistent.
A5406          CMPT_RESET, // Clock initializes to 0.
A5407          CMPT_ACCUMULATE, // Clock initializes to a value stored shortly before shutdown. This is effectively an aggregate ←
A5408                      // runtime meter.
A5409          CMPT_OTHER // Another clock initialization scheme is used.
A5410      }; // enum ClockMonotonicPersistenceType
```

# TMS Data Model

## DRAFT 19 Apr 2021

```
A5411
A5412
A5413 @extensibility(APPENDABLE)
A5414 enum DesiredCircuitContinuity {
A5415     DCC_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5416     could not be determined.
A5417     DCC_OPEN, // Transition to CC_OPEN.
A5418     DCC_CLOSED, // Transition to CC_CLOSED. Predicated on AuthorizationToEnergizeRequest.
A5419     DCC_SYNC_CLOSED, // Transition to CC_CLOSED when synchronization is achieved. May close to a dead circuit, ↵
A5420     predicated on AuthorizationToEnergizeRequest. May act as DCC_CLOSED_INTERNAL when the internal circuit is ↵
A5421     dead.
A5422     DCC_CLOSED_INTERNAL, // Transition to CC_CLOSED. One side of the switch must be a circuit enclosed inside a ↵
A5423     device with protective guards.
A5424     DCC_NO_CHANGE // No transition. May be used instead of DCC_CLOSED in non-active settings groups.
A5425 }; // enum DesiredCircuitContinuity
A5426
A5427
A5428 @extensibility(APPENDABLE)
A5429 enum DeviceRole {
A5430     ROLE_MICROGRID_CONTROLLER, // Microgrid controller, a control service.
A5431     ROLE_SOURCE, // Source power device.
A5432     ROLE_LOAD, // Load power device.
A5433     ROLE_STORAGE, // Storage power device.
A5434     ROLE_DISTRIBUTION, // Distribution power device.
A5435     ROLE_MICROGRID_DASHBOARD, // Microgrid dashboard, a control service.
A5436     ROLE_CONVERSION, // Conversion power device.
A5437     ROLE_MONITOR // Monitor microgrid, read-only access.
A5438 }; // enum DeviceRole
A5439
A5440
A5441 @extensibility(APPENDABLE)
A5442 enum DtcSeverity {
A5443     SEV_0_NOT_SPECIFIED, // [PROPOSAL] Undefined severity. Never transmitted in this version of the standard.
A5444     SEV_1_CLEAR, // [PROPOSAL] DTC has cleared. Never transmitted in this version of the standard.
A5445     SEV_2_INFORMATIVE, // Informational only. No performance impact is expected.
A5446     SEV_3_PREVENTATIVE, // Preventative maintenance has been scheduled.
A5447     SEV_4_DEGRADED, // Poor configuration or operation resulting in degraded performance, including reduced ↵
A5448     efficiency and increased emissions.
A5449     SEV_5_WARNING, // Increased wear and premature failure likely.
A5450     SEV_6_MINOR, // The device may be failing to meet operating performance criteria. The DTC may represent cause or ↵
A5451     effect.
A5452     SEV_7_MAJOR, // This condition may cause protection logic to enter a controlled shutdown sequence. A corrective ↵
A5453     action may be required before the device can be restarted.
A5454     SEV_8_CRITICAL, // A loss of essential functionality has been detected. Routine maintenance or repair may be ↵
A5455     required.
A5456     SEV_9_FATAL, // A loss of essential functionality has been detected. Permanent damage may have occurred.
A5457     SEV_10_HUMAN_SAFETY // Potential user safety risk.
A5458 }; // enum DtcSeverity
A5459
A5460
A5461 @extensibility(APPENDABLE)
A5462 enum GroundFaultDetection {
A5463     GFD_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5464     could not be determined.
A5465     GFD_NONE, // No ground fault detected.
A5466     GFD_ANY, // Ground fault detected. Multiple lines detected or specific line unknown.
A5467     GFD_A, // Ground fault detected on phase A.
A5468     GFD_B, // Ground fault detected on phase B.
A5469     GFD_C, // Ground fault detected on phase C.
A5470     GFD_DCPOS, // Ground fault detected on DC+.
A5471     GFD_DCNEG // Ground fault detected on DC-.
A5472 }; // enum GroundFaultDetection
A5473
A5474
A5475 @extensibility(APPENDABLE)
A5476 enum GroundingDesignType {
A5477     GROUNDING_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↵
A5478     value could not be determined.
A5479     GROUNDING_UNGROUNDED, // No grounding.
A5480     GROUNDING_SOLID, // Solidly grounded.
A5481     GROUNDING_HIGH_RESISTANCE, // High-resistance grounded system. Generally designed for 1-10 A ground fault current ↵
A5482     .
A5483     GROUNDING_LOW_RESISTANCE, // Low-resistance grounded system. Generally designed to limit equipment damage.
A5484     GROUNDING_REACTANCE // Reactance grounded system. Generally limits ground fault current to rated line current.
A5485 }; // enum GroundingDesignType
A5486
A5487
A5488 @extensibility(APPENDABLE)
A5489 enum MicrogridMembership {
A5490     MM_UNKNOWN, // Uninitialized value.
A5491     MM_JOIN, // Join or stay in the microgrid.
A5492     MM_LEAVE // Leave or stay out of the microgrid.
A5493 }; // enum MicrogridMembership
A5494
A5495
A5496 @extensibility(APPENDABLE)
A5497 enum MicrogridMembershipResult {
A5498     MMR_UNKNOWN, // Uninitialized value.
A5499     MMR_REPLACED, // New request received before preparation for this request was complete. Device should wait for ↵
A5500     the new request to complete.
A5501     MMR_COMPLETE, // Completed preparation for the requested action. No significant issues were encountered.
A5502     MMR_BLOCKED // Preparation could not be completed without a significant negative impact such as shedding load. ↵
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5503         Operator should resolve the issue before proceeding.
A5504     }; // enum MicrogridMembershipResult
A5505
A5506
A5507     @extensibility(APPENDABLE)
A5508     enum OperatingMode {
A5509         OPM_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5510             could not be determined.
A5511         OPM_NORMAL, // Normal operating mode.
A5512         OPM_EMERGENCY, // Emergency operating mode.
A5513         OPM_SILENT_WATCH // Silent watch mode.
A5514     }; // enum OperatingMode
A5515
A5516
A5517     @extensibility(APPENDABLE)
A5518     enum OperatorPriorityType {
A5519         OPT_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5520             could not be determined.
A5521         OPT_NEVER_OPERATE, // Transition the device to SS_ON or open the power port.
A5522         OPT_ALWAYS_OPERATE, // Transition the device to SS_SOURCING or close the power port.
A5523         OPT_NUMERIC_RANK // Priority based on OperatorPriority.numericRank.
A5524     }; // enum OperatorPriorityType
A5525
A5526
A5527     @extensibility(APPENDABLE)
A5528     enum OperatorIntentType {
A5529         OIT_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5530             could not be determined.
A5531         OIT_DEFAULT_INTENT, // Microgrid Controller defined default operator intent.
A5532         OIT_OPERATOR_DEFINED // Microgrid Dashboard activated operator intent.
A5533     }; // enum OperatorIntentType
A5534
A5535
A5536     @extensibility(APPENDABLE)
A5537     enum PowerConnectionDetectionType {
A5538         PCDT_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5539             could not be determined.
A5540         PCDT_OPERATOR, // Operator entry.
A5541         PCDT_CABLE_ID, // Identification embedded in a smart cable.
A5542         PCDT_PROBE, // Active probe signal sent down the line.
A5543         PCDT_CORRELATION, // Correlation of power measurements was used.
A5544         PCDT_HISTORY // Device reset or other event has lost continuous monitoring of the connection. This connection was ↵
A5545             detected before the event and has neither been confirmed nor denied since.
A5546     }; // enum PowerConnectionDetectionType
A5547
A5548
A5549     @extensibility(APPENDABLE)
A5550     enum PowerConnectorPhases {
A5551         PHASE_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↵
A5552             value could not be determined.
A5553         PHASE_SINGLE, // Single-phase circuit.
A5554         PHASE_SPLIT, // Split-phase circuit.
A5555         PHASE_3WYE, // 3-phase circuit, wye configuration.
A5556         PHASE_3DELTA, // 3-phase circuit, delta configuration.
A5557         PHASE_DC // DC circuit.
A5558     }; // enum PowerConnectorPhases
A5559
A5560
A5561     @extensibility(APPENDABLE)
A5562     enum PowerConnectorPolarity {
A5563         POLARITY_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↵
A5564             value could not be determined.
A5565         POLARITY_PIN, // This connector contains pins to insert in sockets.
A5566         POLARITY_SOCKET, // This connector contains sockets to receive pins. Includes terminal blocks.
A5567         POLARITY_UNIVERSAL // This connector is symmetric.
A5568     }; // enum PowerConnectorPolarity
A5569
A5570
A5571     @extensibility(APPENDABLE)
A5572     enum PowerConnectorType {
A5573         CONNECTOR_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↵
A5574             value could not be determined.
A5575         CONNECTOR_TERMINAL_BLOCK, // Terminal block.
A5576         CONNECTOR_MILSTD, // Generic MIL spec connector. MIL-STD-1651, MIL-DTL-22992, and MIL-DTL-53126, as used by PDISE ↵
A5577             .
A5578         CONNECTOR_NEMA5, // For convenience power ports, etc.
A5579         CONNECTOR_CAMLOCK, // As used by MEPDIS-R.
A5580         CONNECTOR_POWERLOCK, // ITT VEAM PowerLock
A5581         CONNECTOR_IEC60309, // As used by MEPDIS-R.
A5582         CONNECTOR_J1772, // Electric vehicle charging cable.
A5583         CONNECTOR_POWERLOK, // Amphenol PowerLok.
A5584         CONNECTOR_MILSTD1651, // MIL-STD-1651.
A5585         CONNECTOR_MILDTL22992, // MIL-DTL-22992.
A5586         CONNECTOR_MILDTL53126, // MIL-DTL-53126.
A5587         CONNECTOR_OTHER, // Other connector type, not listed elsewhere.
A5588         CONNECTOR_LOGICAL // Not a physical connector. Used to represent bus measurements or synchronizing voltage.
A5589     }; // enum PowerConnectorType
A5590
A5591
A5592     @extensibility(APPENDABLE)
A5593     enum PowerPortDirectionality {
A5594         PPD_UNKNOWN, // not initialized.
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5595     PPD_NONE, // directionality not applicable (e.g., for a logical port, expect zero power).
A5596     PPD_IN, // Port intended to receive power.
A5597     PPD_OUT, // Port intended to send power.
A5598     PPD_IN_OUT // Port intended to send and receive power.
A5599 }; // enum PowerPortDirectionality
A5600
A5601
A5602 @extensibility(APPENDABLE)
A5603 enum SwitchFeature {
A5604     SWITCH_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↩
A5605     value could not be determined.
A5606     SWITCH_MANUAL, // Manual operation.
A5607     SWITCH_AUTO, // Automatic (internal) logic such as load shed.
A5608     SWITCH_REMOTE, // Remote operation (PowerSwitchRequest).
A5609     SWITCH_RECLOSER, // Able to close without manual operation (automatic or remote).
A5610     SWITCH_BREAKER, // Over current / thermal protection.
A5611     SWITCH_GFI, // Ground fault protection.
A5612     SWITCH_ARC_FLASH, // Arc flash protection.
A5613     SWITCH_SYNCHRONIZER, // Synchronize before close.
A5614     SWITCH_SURGE // Over voltage protection.
A5615 }; // enum SwitchFeature
A5616
A5617
A5618 @extensibility(APPENDABLE)
A5619 enum PowerSwitchReason {
A5620     PSR_UNKNOWN, // No information is available, either due to design or malfunction.
A5621     PSR_MANUAL, // Sensors indicate manual operation.
A5622     PSR_COMMAND, // The switch transitioned due to a (remote) command.
A5623     PSR_PROTECTION, // The switch transitioned due to local protection function.
A5624     PSR_FAILURE, // The transition appears to be due to an internal failure, unrelated to another reason.
A5625     PSR_LOAD_POLICY // The switch opened due to a load policy setting. See DeviceLoadPolicyStatus.
A5626 }; // enum PowerSwitchReason
A5627
A5628
A5629 @extensibility(APPENDABLE)
A5630 enum SourceState {
A5631     SS_UNKNOWN, // Uninitialized value.
A5632     SS_OFF, // OFF state.
A5633     SS_ON, // ON state.
A5634     SS_RUNNING, // RUNNING state.
A5635     SS_SOURCING // SOURCING state.
A5636 }; // enum SourceState
A5637
A5638
A5639 @extensibility(APPENDABLE)
A5640 enum SourceTransition {
A5641     ST_UNKNOWN, // Uninitialized value.
A5642     ST_NONE, // No transition (stay in present state).
A5643     ST_POWER_UP, // Transition from OFF to ON.
A5644     ST_POWER_DOWN, // Transition from ON to OFF.
A5645     ST_START, // Transition from ON to RUNNING.
A5646     ST_STOP, // Transition from RUNNING to ON.
A5647     ST_CONNECT, // Transition from RUNNING to SOURCING.
A5648     ST_DISCONNECT, // Transition from SOURCING to RUNNING.
A5649     ST_JUMP_START, // Transition from ON to SOURCING.
A5650     ST_RAPID_STOP // Transition from SOURCING to ON.
A5651 }; // enum SourceTransition
A5652
A5653
A5654 @extensibility(APPENDABLE)
A5655 enum SourceFeature {
A5656     SRCF_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↩
A5657     could not be determined.
A5658     SRCF_GENSET, // Genset source.
A5659     SRCF_FUEL_CELL, // Fuel cell source.
A5660     SRCF_SOLAR, // Solar source.
A5661     SRCF_WIND, // Wind source.
A5662     SRCF_VEHICLE // Source shared with a vehicle power train.
A5663 }; // enum SourceFeature
A5664
A5665
A5666 @extensibility(APPENDABLE)
A5667 enum LoadFeature {
A5668     LOADF_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↩
A5669     value could not be determined.
A5670     LOADF_DEMAND_RESPONSE, // Device can shed load to provide grid support.
A5671     LOADF_CHANGE_NOTIFICATION, // Device can notify the microgrid before causing a large load transient.
A5672     LOADF_SOFT_START // Device can smoothly ramp up and down to minimize load transients.
A5673 }; // enum LoadFeature
A5674
A5675
A5676 @extensibility(APPENDABLE)
A5677 enum DistributionFeature {
A5678     DISTF_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↩
A5679     value could not be determined.
A5680     DISTF_CLAMP_METER, // clamp-on power meter, two logical power ports.
A5681     DISTF_TAP_METER, // pass-through power meter, two physical power ports.
A5682     DISTF_PCC, // has additional capabilities for PCC interconnect with another grid.
A5683     DISTF_FEEDER, // intended for supplying power to other DIST devices.
A5684     DISTF_DISTRIBUTION // intended for supplying power to LOAD devices.
A5685 }; // enum DistributionFeature
A5686
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5687
A5688 @extensibility(APPENDABLE)
A5689 enum ConversionFeature {
A5690     CONV_F_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↵
A5691         value could not be determined.
A5692     CONV_F_ACTIVE, // Uses power electronics or other active components (programmable).
A5693     CONV_F_PASSIVE // Uses transformers or other passive components (fixed function).
A5694 }; // enum ConversionFeature
A5695
A5696
A5697 @extensibility(APPENDABLE)
A5698 enum MicrogridDashboardFeature {
A5699     MDF_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5700         could not be determined.
A5701     MDF_DISPLAY, // Can show microgrid status.
A5702     MDF_CONTROL // Can control microgrid status.
A5703 }; // enum MicrogridDashboardFeature
A5704
A5705
A5706 @extensibility(APPENDABLE)
A5707 enum MicrogridControllerFeature {
A5708     MCF_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate value ↵
A5709         could not be determined.
A5710     MCF_FIXED, // special-purpose microgrid controller designed for a single platform or limited microgrid ↵
A5711         configuration.
A5712     MCF_GENERAL // general-purpose microgrid controller.
A5713 }; // enum MicrogridControllerFeature
A5714
A5715
A5716 @extensibility(APPENDABLE)
A5717 enum StorageFeature {
A5718     STORF_UNKNOWN, // Unknown value. This is a generic placeholder representing the case in which an appropriate ↵
A5719         value could not be determined.
A5720     STORF_GRID, // grid attached storage, does not require pass-through from source to load.
A5721     STORF_SUBCYCLE_UPS, // can provide fast voltage support to recover from a grid fault condition.
A5722     STORF_CHARGING, // charging station.
A5723     STORF_VEHICLE // storage shared with a vehicle power train.
A5724 }; // enum StorageFeature
A5725
A5726
A5727 @extensibility(APPENDABLE)
A5728 enum ConfigId {
A5729     CONFIG_DEFAULTS, // Read-only manufacturer defaults.
A5730     CONFIG_ACTIVE, // Settings for use at the present time.
A5731     CONFIG_ON_REBOOT, // Settings for use after a reboot / restart.
A5732     CONFIG_ON_COMMS_LOSS, // Settings for use after a communications loss timeout.
A5733     CONFIG_ON_COPY // Settings for use after a CopyConfigRequest.
A5734 }; // enum ConfigId
A5735
A5736
A5737 /**
A5738  * Describes the type of fault
A5739  */
A5740 typedef octet FailureModeIndicator;
A5741
A5742 /** Data valid but above normal operational range, most severe level. */
A5743 const octet FMI_HIGH_MOST_SEVERE = 0;
A5744
A5745 /** Data valid but below normal operational range, most severe level. */
A5746 const octet FMI_LOW_MOST_SEVERE = 1;
A5747
A5748 /** Data valid but above normal operational range, least severe level. */
A5749 const octet FMI_HIGH_LEAST_SEVERE = 15;
A5750
A5751 /** Data valid but above normal operational range, moderately severe level. */
A5752 const octet FMI_HIGH_MODERATELY_SEVERE = 16;
A5753
A5754 /** Data valid but below normal operational range, least severe level. */
A5755 const octet FMI_LOW_LEAST_SEVERE = 17;
A5756
A5757 /** Data valid but below normal operational range, moderately severe level. */
A5758 const octet FMI_LOW_MODERATELY_SEVERE = 18;
A5759
A5760
A5761 const unsigned long Fingerprint_LEN = 32;
A5762 /**
A5763  * Short representation of PublicKey, suitable for uniquely identifying a device, person, platform, or other entity. ↵
A5764  * A value of all 0s represents an invalid fingerprint
A5765  * Fixed length: 32
A5766  */
A5767 typedef octet Fingerprint[Fingerprint_LEN];
A5768
A5769
A5770 const unsigned long GlobalTradeItemNumber_LEN = 14;
A5771 /**
A5772  * Encodes a Global Trade Item Number (GTIN).
A5773  * Fixed length: 14
A5774  */
A5775 typedef char GlobalTradeItemNumber[GlobalTradeItemNumber_LEN];
A5776
A5777
A5778 /**
```



*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5779      * Identifies a grounding circuit within a device.
A5780      */
A5781      typedef uint32 GroundingCircuitNumber;
A5782
A5783      /** Maximum number of grounding circuits that a device may contain. */
A5784      const uint32 MAX_GROUNDS = 20;
A5785
A5786
A5787      const unsigned long NatoStockNumber_LEN = 13;
A5788      /**
A5789       * Encodes a NATO Stock Number (NSN), aka National Stock Number.
A5790       * Fixed length: 13
A5791       */
A5792      typedef char NatoStockNumber[NatoStockNumber_LEN];
A5793
A5794
A5795      /**
A5796       * Identifies a port within a device.
A5797       */
A5798      typedef uint32 PowerPortNumber;
A5799
A5800      /** Maximum number of power ports that a device may contain. */
A5801      const uint32 MAX_PORTS = 64;
A5802
A5803      /** Number to use for a single-port device with no listed number. */
A5804      const uint32 ONLY_PORT = 0;
A5805
A5806
A5807      const unsigned long String16_MINLEN = 0;
A5808      const unsigned long String16_MAXLEN = 16;
A5809      /**
A5810       * A variable length string.
A5811       * Minimum length: 0
A5812       * Maximum length: 16
A5813       */
A5814      typedef string <String16_MAXLEN> String16;
A5815
A5816
A5817      const unsigned long String32_MINLEN = 0;
A5818      const unsigned long String32_MAXLEN = 32;
A5819      /**
A5820       * A variable length string.
A5821       * Minimum length: 0
A5822       * Maximum length: 32
A5823       */
A5824      typedef string <String32_MAXLEN> String32;
A5825
A5826
A5827      const unsigned long String1_32_MINLEN = 1;
A5828      const unsigned long String1_32_MAXLEN = 32;
A5829      /**
A5830       * A variable length string with a fixed minimum.
A5831       * Minimum length: 1
A5832       * Maximum length: 32
A5833       */
A5834      typedef string <String1_32_MAXLEN> String1_32;
A5835
A5836
A5837      /**
A5838       * Identifies the sensor or subsystem at issue or a particular event or condition.
A5839       */
A5840      typedef uint32 SuspectParameterNumber;
A5841
A5842
A5843      /**
A5844       * Tap changer index in a passive conversion device. Defaults to 0 when no changer is present.
A5845       */
A5846      typedef int32 TapNumber;
A5847
A5848      /** Maximum number of positions in a tap changer. Many tap changers have 33 positions. */
A5849      const int32 MAX_TAPS = 64;
A5850
A5851
A5852      const unsigned long ConnectorFeatureSequence_MINLEN = 0;
A5853      const unsigned long ConnectorFeatureSequence_MAXLEN = 5;
A5854      /**
A5855       * A sequence of ConnectorFeature.
A5856       * Minimum length: 0
A5857       * Maximum length: 5
A5858       */
A5859      typedef sequence<ConnectorFeature, ConnectorFeatureSequence_MAXLEN> ConnectorFeatureSequence;
A5860
A5861
A5862      const unsigned long PowerPortNumberSequence_MINLEN = 0;
A5863      const unsigned long PowerPortNumberSequence_MAXLEN = MAX_PORTS;
A5864      /**
A5865       * A sequence of PowerPortNumber.
A5866       * Minimum length: 0
A5867       * Maximum length: MAX_PORTS
A5868       */
A5869      typedef sequence<PowerPortNumber, PowerPortNumberSequence_MAXLEN> PowerPortNumberSequence;
A5870
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5871
A5872     const unsigned long ConversionFeatureSequence_MINLEN = 0;
A5873     const unsigned long ConversionFeatureSequence_MAXLEN = 2;
A5874     /**
A5875      * A sequence of ConversionFeature.
A5876      * Minimum length: 0
A5877      * Maximum length: 2
A5878      */
A5879     typedef sequence<ConversionFeature, ConversionFeatureSequence_MAXLEN> ConversionFeatureSequence;
A5880
A5881
A5882     const unsigned long OctetSequence_MINLEN = 1;
A5883     const unsigned long OctetSequence_MAXLEN = 32768;
A5884     /**
A5885      * A sequence of Octet (byte).
A5886      * Minimum length: 1
A5887      * Maximum length: 32768
A5888      */
A5889     typedef sequence<octet, OctetSequence_MAXLEN> OctetSequence;
A5890
A5891
A5892     const unsigned long DistributionFeatureSequence_MINLEN = 0;
A5893     const unsigned long DistributionFeatureSequence_MAXLEN = 5;
A5894     /**
A5895      * A sequence of DistributionFeature.
A5896      * Minimum length: 0
A5897      * Maximum length: 5
A5898      */
A5899     typedef sequence<DistributionFeature, DistributionFeatureSequence_MAXLEN> DistributionFeatureSequence;
A5900
A5901
A5902     const unsigned long SwitchFeatureSequence_MINLEN = 0;
A5903     const unsigned long SwitchFeatureSequence_MAXLEN = 9;
A5904     /**
A5905      * A sequence of SwitchFeature.
A5906      * Minimum length: 0
A5907      * Maximum length: 9
A5908      */
A5909     typedef sequence<SwitchFeature, SwitchFeatureSequence_MAXLEN> SwitchFeatureSequence;
A5910
A5911
A5912     const unsigned long LoadFeatureSequence_MINLEN = 0;
A5913     const unsigned long LoadFeatureSequence_MAXLEN = 3;
A5914     /**
A5915      * A sequence of LoadFeature.
A5916      * Minimum length: 0
A5917      * Maximum length: 3
A5918      */
A5919     typedef sequence<LoadFeature, LoadFeatureSequence_MAXLEN> LoadFeatureSequence;
A5920
A5921
A5922     const unsigned long MicrogridControllerFeatureSequence_MINLEN = 1;
A5923     const unsigned long MicrogridControllerFeatureSequence_MAXLEN = 64;
A5924     /**
A5925      * A sequence of MicrogridControllerFeature.
A5926      * Minimum length: 1
A5927      * Maximum length: 64
A5928      */
A5929     typedef sequence<MicrogridControllerFeature, MicrogridControllerFeatureSequence_MAXLEN> ↵
A5930         MicrogridControllerFeatureSequence;
A5931
A5932
A5933     const unsigned long MicrogridDashboardFeatureSequence_MINLEN = 1;
A5934     const unsigned long MicrogridDashboardFeatureSequence_MAXLEN = 2;
A5935     /**
A5936      * A sequence of MicrogridDashboardFeature.
A5937      * Minimum length: 1
A5938      * Maximum length: 2
A5939      */
A5940     typedef sequence<MicrogridDashboardFeature, MicrogridDashboardFeatureSequence_MAXLEN> ↵
A5941         MicrogridDashboardFeatureSequence;
A5942
A5943
A5944     const unsigned long EnumLabelsSequence_MINLEN = 0;
A5945     const unsigned long EnumLabelsSequence_MAXLEN = 100;
A5946     /**
A5947      * A sequence of String1_32.
A5948      * Minimum length: 0
A5949      * Maximum length: 100
A5950      */
A5951     typedef sequence<String1_32, EnumLabelsSequence_MAXLEN> EnumLabelsSequence;
A5952
A5953
A5954     const unsigned long SourceFeatureSequence_MINLEN = 0;
A5955     const unsigned long SourceFeatureSequence_MAXLEN = 5;
A5956     /**
A5957      * A sequence of SourceFeature.
A5958      * Minimum length: 0
A5959      * Maximum length: 5
A5960      */
A5961     typedef sequence<SourceFeature, SourceFeatureSequence_MAXLEN> SourceFeatureSequence;
A5962
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A5963
A5964     const unsigned long StorageFeatureSequence_MINLEN = 0;
A5965     const unsigned long StorageFeatureSequence_MAXLEN = 4;
A5966     /**
A5967      * A sequence of StorageFeature.
A5968      * Minimum length: 0
A5969      * Maximum length: 4
A5970      */
A5971     typedef sequence<StorageFeature, StorageFeatureSequence_MAXLEN> StorageFeatureSequence;
A5972
A5973
A5974     const unsigned long ThermalLoadSequence_MINLEN = 0;
A5975     const unsigned long ThermalLoadSequence_MAXLEN = 5;
A5976     /**
A5977      * A sequence of float32.
A5978      * Minimum length: 0
A5979      * Maximum length: 5
A5980      */
A5981     typedef sequence<float, ThermalLoadSequence_MAXLEN> ThermalLoadSequence;
A5982
A5983
A5984     const unsigned long ThermalZoneSequence_MINLEN = 0;
A5985     const unsigned long ThermalZoneSequence_MAXLEN = 5;
A5986     /**
A5987      * A sequence of String1_32.
A5988      * Minimum length: 0
A5989      * Maximum length: 5
A5990      */
A5991     typedef sequence<String1_32, ThermalZoneSequence_MAXLEN> ThermalZoneSequence;
A5992
A5993
A5994     const unsigned long TopicConditionSequence_MINLEN = 0;
A5995     const unsigned long TopicConditionSequence_MAXLEN = 5;
A5996     /**
A5997      * A sequence of TopicCondition.
A5998      * Minimum length: 0
A5999      * Maximum length: 5
A6000      */
A6001     typedef sequence<String1_32, TopicConditionSequence_MAXLEN> TopicConditionSequence;
A6002
A6003
A6004
A6005     /**
A6006      * Key value for a request that targets all devices in the grid.
A6007      */
A6008     @nested
A6009     @extensibility(APPENDABLE)
A6010     struct GridRequest {
A6011
A6012         /**
A6013          * Identity of the device that sent this request.
A6014          */
A6015         Fingerprint requestingDeviceId;
A6016     }; // end struct GridRequest
A6017
A6018
A6019
A6020     /**
A6021      * Local oscillator timestamp, not affected by clock jumps.
A6022      */
A6023     @nested
A6024     @extensibility(APPENDABLE)
A6025     struct ClockMonotonic {
A6026
A6027         /**
A6028          * integer portion of the timestamp.
A6029          */
A6030         uint32 seconds;
A6031
A6032         /**
A6033          * fractional portion of the timestamp.
A6034          */
A6035         uint32 nanoseconds;
A6036     }; // end struct ClockMonotonic
A6037
A6038
A6039
A6040     /**
A6041      * Design information for thermal management.
A6042      */
A6043     @nested
A6044     @extensibility(APPENDABLE)
A6045     struct ThermalInfo {
A6046
A6047         /**
A6048          * Name of each thermal zone.
A6049          */
A6050         ThermalZoneSequence thermalZone;
A6051     }; // end struct ThermalInfo
A6052
A6053
A6054
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6055 /**
A6056  * Provides operating information of the thermal management in this device.
A6057  */
A6058 @nested
A6059 @extensibility(APPENDABLE)
A6060 struct ThermalState {
A6061
A6062     /**
A6063      * Load for each thermalZone, in same order published in tms.ThermalInfo.
A6064     */
A6065     ThermalLoadSequence thermalLoad;
A6066 }; // end struct ThermalState
A6067
A6068
A6069
A6070 /**
A6071  * Design information for power transformers.
A6072  */
A6073 @nested
A6074 @extensibility(APPENDABLE)
A6075 struct TransformerInfo {
A6076
A6077     /**
A6078      * TODO feedback requested.
A6079     */
A6080     boolean placeholder;
A6081 }; // end struct TransformerInfo
A6082
A6083
A6084
A6085 /**
A6086  * Provides operating information about about power transformer devices.
A6087  */
A6088 @nested
A6089 @extensibility(APPENDABLE)
A6090 struct TransformerState {
A6091
A6092     /**
A6093      * TODO feedback requested.
A6094     */
A6095     boolean placeholder;
A6096 }; // end struct TransformerState
A6097
A6098
A6099
A6100 /**
A6101  * Design information for power electronics.
A6102  */
A6103 @nested
A6104 @extensibility(APPENDABLE)
A6105 struct PowerElectronicsInfo {
A6106
A6107     /**
A6108      * Minimum rated operating temperature.
A6109      * units: Celsius (C)
A6110     */
A6111     @range(min=-273, max=10000)
A6112     @unit("C")
A6113     float minTemperature;
A6114
A6115     /**
A6116      * Maximum rated operating temperature.
A6117      * units: Celsius (C)
A6118     */
A6119     @range(min=-273, max=10000)
A6120     @unit("C")
A6121     float maxTemperature;
A6122
A6123     /**
A6124      * Minimum (greatest output) rated operating power.
A6125      * units: watt (W)
A6126     */
A6127     @unit("W")
A6128     float minPower;
A6129
A6130     /**
A6131      * Minimum (maximum input) rated operating power.
A6132      * units: watt (W)
A6133     */
A6134     @unit("W")
A6135     float maxPower;
A6136
A6137     /**
A6138      * Maximum (in/out) rated apparent power.
A6139      * units: volt-ampere (VA)
A6140     */
A6141     @unit("VA")
A6142     float maxApparentPower;
A6143
A6144     /**
A6145      * Rated short circuit to not cause damage to the device.
A6146      * units: Ampere (A)
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6147      */
A6148      @unit("A")
A6149      float shortCircuitTolerance;
A6150 }; // end struct PowerElectronicsInfo
A6151
A6152
A6153
A6154 /**
A6155  * Operating state of the power electronics hardware.
A6156  */
A6157 @nested
A6158 @extensibility(APPENDABLE)
A6159 struct PowerElectronicsState {
A6160
A6161     /**
A6162     * Present temperature.
A6163     * units: Celsius (C)
A6164     */
A6165     @range(min=-273, max=10000)
A6166     @unit("C")
A6167     float temperature;
A6168 }; // end struct PowerElectronicsState
A6169
A6170
A6171
A6172 /**
A6173  * Operating state of the energy storage device or unit.
A6174  */
A6175 @nested
A6176 @extensibility(APPENDABLE)
A6177 struct EnergyStorageState {
A6178
A6179     /**
A6180     * Estimated energy stored.
A6181     * units: joule (J)
A6182     */
A6183     @unit("J")
A6184     float stateOfCharge;
A6185
A6186     /**
A6187     * Present temperature of the energy storage components.
A6188     * units: Celsius (C)
A6189     */
A6190     @range(min=-273.15, max=10000)
A6191     @unit("C")
A6192     float temperature;
A6193 }; // end struct EnergyStorageState
A6194
A6195
A6196
A6197 /**
A6198  * Design information for an energy storage device or unit.
A6199  */
A6200 @nested
A6201 @extensibility(APPENDABLE)
A6202 struct EnergyStorageInfo {
A6203
A6204     /**
A6205     * The high state of charge during continuous operations, only to be exceeded for limited-duration operation. ↩
A6206     * Increased wear when charging above this point
A6207     * units: per unit (p.u.)
A6208     */
A6209     @range(min=0, max=1)
A6210     @unit("p.u.")
A6211     float highStateOfCharge;
A6212
A6213     /**
A6214     * The low state of charge during continuous operations, only to go below for limited-duration operation. ↩
A6215     * Increased wear when discharging below this point.
A6216     * units: per unit (p.u.)
A6217     */
A6218     @range(min=0, max=1)
A6219     @unit("p.u.")
A6220     float lowStateOfCharge;
A6221
A6222     /**
A6223     * Rated energy at full state of charge.
A6224     * units: joule (J)
A6225     */
A6226     @unit("J")
A6227     float maxChargeEnergy;
A6228
A6229     /**
A6230     * Minimum rated operating temperature.
A6231     * units: Celsius (C)
A6232     */
A6233     @range(min=-273, max=10000)
A6234     @unit("C")
A6235     float minTemperature;
A6236
A6237     /**
A6238     * Most efficient operating temperature.
```

# TMS Data Model

## DRAFT 19 Apr 2021

```

A6239         * units: Celsius (C)
A6240         */
A6241         @range(min=-273, max=10000)
A6242         @unit("C")
A6243         float nomTemperature;
A6244
A6245         /**
A6246          * Maximum rated operating temperature.
A6247          * units: Celsius (C)
A6248          */
A6249         @range(min=-273, max=10000)
A6250         @unit("C")
A6251         float maxTemperature;
A6252     }; // end struct EnergyStorageInfo
A6253
A6254
A6255
A6256     /**
A6257      * Design information for a power generator.
A6258      */
A6259     @nested
A6260     @extensibility(APPENDABLE)
A6261     struct GeneratorInfo {
A6262
A6263         /**
A6264          * Minimum field current. Indicates field collapse?
A6265          * units: Ampere (A)
A6266          */
A6267         @unit("A")
A6268         float minFieldCurrent;
A6269
A6270         /**
A6271          * Maximum field current. Indicates saturation.
A6272          * units: Ampere (A)
A6273          */
A6274         @unit("A")
A6275         float maxFieldCurrent;
A6276
A6277         /**
A6278          * Maximum stator temperature. Indicates overheating.
A6279          * units: Celsius (C)
A6280          */
A6281         @range(min=-273, max=10000)
A6282         @unit("C")
A6283         float maxStatorTemperature;
A6284     }; // end struct GeneratorInfo
A6285
A6286
A6287
A6288     /**
A6289      * Report information on AVR and generator condition.
A6290      */
A6291     @nested
A6292     @extensibility(APPENDABLE)
A6293     struct GeneratorState {
A6294
A6295         /**
A6296          * field current, as controlled by AVR.
A6297          * units: Ampere (A)
A6298          */
A6299         @unit("A")
A6300         float fieldCurrent;
A6301
A6302         /**
A6303          * stator temperature.
A6304          * units: Celsius (C)
A6305          */
A6306         @unit("C")
A6307         float statorTemperature;
A6308     }; // end struct GeneratorState
A6309
A6310
A6311
A6312     /**
A6313      * Design information for an engine, such as on a generator set (genset).
A6314      */
A6315     @nested
A6316     @extensibility(APPENDABLE)
A6317     struct EngineInfo {
A6318
A6319         /**
A6320          * Minimum oil pressure for the operation of the engine that prevents engine damage.
A6321          * units: pascal (Pa)
A6322          */
A6323         @unit("Pa")
A6324         float minOilPressure;
A6325
A6326         /**
A6327          * Maximum oil pressure for the operation of the engine that prevents engine damage.
A6328          * units: pascal (Pa)
A6329          */
A6330         @unit("Pa")

```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6331     float maxOilPressure;
A6332
A6333     /**
A6334      * Minimum coolant temperature for the operation of the engine that prevents engine damage.
A6335      * units: Celsius (C)
A6336      */
A6337     @unit("C")
A6338     float minCoolantTemperature;
A6339
A6340     /**
A6341      * Maximum coolant temperature for the operation of the engine that prevents engine damage.
A6342      * units: Celsius (C)
A6343      */
A6344     @unit("C")
A6345     float maxCoolantTemperature;
A6346
A6347     /**
A6348      * Stall warning speed of the engine. Speeds below this threshold may cause a malfunction or protective shutdown.
A6349      * units: radian per second (rad/s)
A6350      */
A6351     @unit("rad/s")
A6352     float minEngineSpeed;
A6353
A6354     /**
A6355      * Maximum engine speed. I.e. the red line
A6356      * units: radian per second (rad/s)
A6357      */
A6358     @unit("rad/s")
A6359     float maxEngineSpeed;
A6360
A6361     /**
A6362      * Minimum load required to prevent wet stack build up.
A6363      * units: watt (W)
A6364      */
A6365     @unit("W")
A6366     float minWetStackPreventionLoad;
A6367
A6368     /**
A6369      * Minimum load required to clear wet stack build up.
A6370      * units: watt (W)
A6371      */
A6372     @unit("W")
A6373     float minWetStackMitigationLoad;
A6374 }; // end struct EngineInfo
A6375
A6376
A6377
A6378 /**
A6379  * Report key parameters about the state of the engine.
A6380  */
A6381 @nested
A6382 @extensibility(APPENDABLE)
A6383 struct EngineState {
A6384
A6385     /**
A6386      * Oil pressure.
A6387      * units: pascal (Pa)
A6388      */
A6389     @unit("Pa")
A6390     float oilPressure;
A6391
A6392     /**
A6393      * Coolant temperature.
A6394      * units: Celsius (C)
A6395      */
A6396     @range(min=-273.15, max=10000)
A6397     @unit("C")
A6398     float coolantTemperature;
A6399
A6400     /**
A6401      * Speed.
A6402      * units: radian per second (rad/s)
A6403      */
A6404     @min(0)
A6405     @unit("rad/s")
A6406     float speed;
A6407
A6408     /**
A6409      * wet stack per unit, mitigation required or indicated at 1.
A6410      * units: per unit (p.u.)
A6411      */
A6412     @range(min=0, max=1)
A6413     @unit("p.u.")
A6414     @optional
A6415     float wetStack;
A6416
A6417     /**
A6418      * Engine runtime hours.
A6419      * units: hour (h)
A6420      */
A6421     @min(0)
A6422     @unit("h")
```

# TMS Data Model

## DRAFT 19 Apr 2021

```

A6423         float engineRuntime;
A6424
A6425         /**
A6426          * Fuel consumption rate at the present power level.
A6427          * units: liters per second (L/s)
A6428          */
A6429         @min(0)
A6430         @unit("L/s")
A6431         @optional
A6432         float fuelConsumptionRate;
A6433     }; // end struct EngineState
A6434
A6435
A6436
A6437     /**
A6438      * Design information for the fuel associated with a power device.
A6439      */
A6440     @nested
A6441     @extensibility(APPENDABLE)
A6442     struct FuelInfo {
A6443
A6444         /**
A6445          * The rated fuel capacity of the device.
A6446          * units: liter (L)
A6447          */
A6448         @min(0)
A6449         @unit("L")
A6450         float maxFuelLevel;
A6451
A6452         /**
A6453          * Level where engine automatically shuts down.
A6454          * units: per unit (p.u.)
A6455          */
A6456         @range(min=0, max=1)
A6457         @unit("p.u.")
A6458         float lowFuelLevelCutoff;
A6459     }; // end struct FuelInfo
A6460
A6461
A6462
A6463     /**
A6464      * Provides dynamic state information regarding fuel levels and pumps.
A6465      */
A6466     @nested
A6467     @extensibility(APPENDABLE)
A6468     struct FuelState {
A6469
A6470         /**
A6471          * The current fuel level of the device.
A6472          * units: liter (L)
A6473          */
A6474         @unit("L")
A6475         float fuelLevel;
A6476
A6477         /**
A6478          * True indicates that fuel is being pulled from external tank (fuel pump is on). False otherwise or if the
A6479          * device does not have a fuel pump.
A6480          */
A6481         boolean fuelPumpRunning;
A6482     }; // end struct FuelState
A6483
A6484
A6485
A6486     /**
A6487      * Design information for a device capable of load sharing.
A6488      */
A6489     @nested
A6490     @extensibility(APPENDABLE)
A6491     struct LoadSharingInfo {
A6492
A6493         /**
A6494          * The power port number.
A6495          */
A6496         @optional
A6497         PowerPortNumber portNumber;
A6498
A6499         /**
A6500          * Indicates that the device can allow the frequency (AC) and voltage (AC and DC) to sag as load increases.
A6501          */
A6502         boolean supportsDroopCurve;
A6503
A6504         /**
A6505          * Supports droop curves having multiple piecewise linear segments.
A6506          */
A6507         boolean supportsNonlinearDroopCurve;
A6508
A6509         /**
A6510          * TBD: units and number of measurements
A6511          */
A6512         float inertia;
A6513
A6514         /**

```



*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6515     * Minimum (greatest output) rated real power (without overload)
A6516     * units: watt (W)
A6517     */
A6518     @unit("W")
A6519     float minP;
A6520
A6521     /**
A6522     * Maximum (greatest input) rated real power (without overload)
A6523     * units: watt (W)
A6524     */
A6525     @unit("W")
A6526     float maxP;
A6527
A6528     /**
A6529     * Greatest overload output real power (short duration or under ideal conditions)
A6530     * units: watt (W)
A6531     */
A6532     @unit("W")
A6533     float minOverloadP;
A6534
A6535     /**
A6536     * Minimum Q value.
A6537     * units: volt ampere reactive (var)
A6538     */
A6539     @unit("var")
A6540     float minQ;
A6541
A6542     /**
A6543     * Maximum Q value.
A6544     * units: volt ampere reactive (var)
A6545     */
A6546     @unit("var")
A6547     float maxQ;
A6548
A6549     /**
A6550     * Rated apparent power
A6551     */
A6552     float maxVA;
A6553
A6554     /**
A6555     * Rated power factor, usually 0.8
A6556     */
A6557     float powerFactor;
A6558 }; // end struct LoadSharingInfo
A6559
A6560
A6561 const unsigned long LoadSharingInfoSequence_MINLEN = 1;
A6562 const unsigned long LoadSharingInfoSequence_MAXLEN = MAX_PORTS;
A6563 /**
A6564 * A sequence of LoadSharingInfo.
A6565 * Minimum length: 1
A6566 * Maximum length: MAX_PORTS
A6567 */
A6568 typedef sequence<LoadSharingInfo, LoadSharingInfoSequence_MAXLEN> LoadSharingInfoSequence;
A6569
A6570
A6571
A6572 /**
A6573 * Design information for power storage devices.
A6574 */
A6575 @nested
A6576 @extensibility(APPENDABLE)
A6577 struct StorageInfo {
A6578
A6579     /**
A6580     * A set of features this device supports.
A6581     */
A6582     StorageFeatureSequence features;
A6583
A6584     /**
A6585     * Load sharing capabilities of this device.
A6586     */
A6587     LoadSharingInfo loadSharing;
A6588 }; // end struct StorageInfo
A6589
A6590
A6591
A6592 /**
A6593 * Design information for control hardware.
A6594 */
A6595 @nested
A6596 @extensibility(APPENDABLE)
A6597 struct ControlHardwareInfo {
A6598
A6599     /**
A6600     * True indicates that the device has a real time clock. False otherwise.
A6601     */
A6602     boolean hasRealtimeClock;
A6603
A6604     /**
A6605     * Minimum rated operating temperature.
A6606     * units: Celsius (C)
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6607      */
A6608      @range(min=-273, max=10000)
A6609      @unit("C")
A6610      float minTemperature;
A6611
A6612      /**
A6613       * Maximum rated operating temperature.
A6614       * units: Celsius (C)
A6615       */
A6616      @range(min=-273, max=10000)
A6617      @unit("C")
A6618      float maxTemperature;
A6619  }; // end struct ControlHardwareInfo
A6620
A6621
A6622
A6623  /**
A6624   * Operating state for the control hardware in this device.
A6625   * Topic Usage:
A6626   * - TOPIC_CONTROL_HARDWARE_STATUS
A6627   */
A6628  @topic
A6629  @extensibility(APPENDABLE)
A6630  struct ControlHardwareState {
A6631
A6632      /**
A6633       * The device described by this structure.
A6634       */
A6635      @key
A6636      Fingerprint deviceId;
A6637
A6638      /**
A6639       * Time of these measurements.
A6640       */
A6641      ClockMonotonic timestamp;
A6642
A6643      /**
A6644       * Central Processing Unit (CPU) usage, as a per unit measure of the whole capacity.
A6645       * units: per unit (p.u.)
A6646       */
A6647      @range(min=0, max=1)
A6648      @unit("p.u.")
A6649      float processorUsage;
A6650
A6651      /**
A6652       * Memory utilization of the hardware, as a per unit measure of the whole capacity.
A6653       * units: per unit (p.u.)
A6654       */
A6655      @range(min=0, max=1)
A6656      @unit("p.u.")
A6657      float memoryUsage;
A6658
A6659      /**
A6660       * Network utilization of the hardware, as a per unit measure of the whole capacity. If the device has multiple ↵
A6661         network interfaces, this field will describe the network interface utilized by TMS communication protocols.
A6662       * units: per unit (p.u.)
A6663       */
A6664      @range(min=0, max=1)
A6665      @unit("p.u.")
A6666      float networkUsage;
A6667
A6668      /**
A6669       * Disk storage utilization of the hardware, as a per unit measure of the whole capacity.
A6670       * units: per unit (p.u.)
A6671       */
A6672      @range(min=0, max=1)
A6673      @unit("p.u.")
A6674      float storageUsage;
A6675
A6676      /**
A6677       * Temperature measurement of the control hardware.
A6678       * units: Celsius (C)
A6679       */
A6680      @range(min=-273, max=10000)
A6681      @unit("C")
A6682      float temperature;
A6683  }; // end struct ControlHardwareState
A6684
A6685
A6686
A6687  /**
A6688   * Design information for power hardware. Power hardware refers to accessories or supporting components of a power ↵
A6689     device, such as an engine, control boards, etc.
A6690   */
A6691  @nested
A6692  @extensibility(APPENDABLE)
A6693  struct PowerHardwareInfo {
A6694
A6695      /**
A6696       * Engine information.
A6697       */
A6698      @optional
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6699     EngineInfo engine;
A6700
A6701     /**
A6702      * Fuel information.
A6703      */
A6704     @optional
A6705     FuelInfo fuel;
A6706
A6707     /**
A6708      * Generator information.
A6709      */
A6710     @optional
A6711     GeneratorInfo generator;
A6712
A6713     /**
A6714      * Energy storage information.
A6715      */
A6716     @optional
A6717     EnergyStorageInfo energyStorage;
A6718
A6719     /**
A6720      * Power electronics information.
A6721      */
A6722     @optional
A6723     PowerElectronicsInfo powerElectronics;
A6724
A6725     /**
A6726      * Transformer information.
A6727      */
A6728     @optional
A6729     TransformerInfo transformer;
A6730
A6731     /**
A6732      * Thermal zone information.
A6733      */
A6734     @optional
A6735     ThermalInfo thermal;
A6736 }; // end struct PowerHardwareInfo
A6737
A6738
A6739
A6740 /**
A6741  * Operating state of the power hardware.
A6742  * Topic Usage:
A6743  * - TOPIC_POWER_HARDWARE_STATUS
A6744  */
A6745 @topic
A6746 @extensibility(APPENDABLE)
A6747 struct PowerHardwareState {
A6748
A6749     /**
A6750      * The device described by this structure.
A6751      */
A6752     @key
A6753     Fingerprint deviceId;
A6754
A6755     /**
A6756      * Time of these measurements.
A6757      */
A6758     ClockMonotonic timestamp;
A6759
A6760     /**
A6761      * Operating state of the engine if present.
A6762      */
A6763     @optional
A6764     EngineState engine;
A6765
A6766     /**
A6767      * Fuel state, if present.
A6768      */
A6769     @optional
A6770     FuelState fuel;
A6771
A6772     /**
A6773      * Operating state of a generator, if present.
A6774      */
A6775     @optional
A6776     GeneratorState generator;
A6777
A6778     /**
A6779      * Operating state of energy storage if present.
A6780      */
A6781     @optional
A6782     EnergyStorageState energyStorage;
A6783
A6784     /**
A6785      * Operating state of active components including inverters and rectifiers, if present.
A6786      */
A6787     @optional
A6788     PowerElectronicsState powerElectronics;
A6789
A6790     /**
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6791      * Operating state of passive components, if present.
A6792      */
A6793      @optional
A6794      TransformerState transformer;
A6795
A6796      /**
A6797       * Thermal operating state including heating and cooling components, if present.
A6798       */
A6799      @optional
A6800      ThermalState thermal;
A6801  }; // end struct PowerHardwareState
A6802
A6803
A6804
A6805      /**
A6806       * Represent an individual alarm or fault condition originating from the indicated sensor or subsystem.
A6807       */
A6808      @nested
A6809      @extensibility(APPENDABLE)
A6810      struct DiagnosticTroubleCode {
A6811
A6812          /**
A6813           * Parameter representing this DTC.
A6814           */
A6815          SuspectParameterNumber suspectParameter;
A6816
A6817          /**
A6818           * Type of fault for this parameter.
A6819           */
A6820          FailureModeIndicator failureMode;
A6821
A6822          /**
A6823           * Number of times this DTC was raised. This field increments on a state change from inactive to active. In order
A6824             to increment, the cleared state must be sensed. The count must not increment simply due to restarts. The
A6825             count should be stored in non-volatile memory in order to preserve it over power cycles. The count
A6826             saturates at 4294967294  $(2^{32}-2)$ . The value 4294967295  $(2^{32}-1)$  indicates that the count is not
A6827             available.
A6828           */
A6829          uint32 occurrenceCount;
A6830
A6831          /**
A6832           * Timestamp when this DTC was raised or activated.
A6833           */
A6834          ClockMonotonic timeRaised;
A6835
A6836          /**
A6837           * Expected impact of the DTC on grid operation. Intended to support automatic handling of parameters that are
A6838             unknown to the receiving system.
A6839           */
A6840          DtcSeverity severity;
A6841
A6842          /**
A6843           * Estimate of when the DTC severity will change, assuming continued operation under present conditions.
A6844           */
A6845          @optional
A6846          ClockMonotonic estimateTime;
A6847
A6848          /**
A6849           * Expected DTC severity at the estimateTime.
A6850           */
A6851          @optional
A6852          DtcSeverity estimateSeverity;
A6853
A6854          /**
A6855           * Short, human-readable text description of the DTC. This should summarize to the operator what the issue is (e.g.
A6856             'oil pressure' or 'low oil pressure'). Intended to support human operators for parameters that are
A6857             unknown to the receiving system.
A6858           */
A6859          String32 hint;
A6860  }; // end struct DiagnosticTroubleCode
A6861
A6862
A6863      const unsigned long DiagnosticTroubleCodeSequence_MINLEN = 0;
A6864      const unsigned long DiagnosticTroubleCodeSequence_MAXLEN = 64;
A6865      /**
A6866       * A sequence of DiagnosticTroubleCode.
A6867       * Minimum length: 0
A6868       * Maximum length: 64
A6869       */
A6870      typedef sequence<DiagnosticTroubleCode, DiagnosticTroubleCodeSequence_MAXLEN> DiagnosticTroubleCodeSequence;
A6871
A6872
A6873
A6874      /**
A6875       * Clock synchronized to a global time base.
A6876       */
A6877      @nested
A6878      @extensibility(APPENDABLE)
A6879      struct ClockRealtime {
A6880
A6881          /**
A6882           * Epoch.
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6883      */
A6884      uint16 epoch;
A6885
A6886      /**
A6887       * integer portion of the timestamp.
A6888       */
A6889      uint32 seconds;
A6890
A6891      /**
A6892       * fractional portion of the timestamp.
A6893       */
A6894      uint32 nanoseconds;
A6895 }; // end struct ClockRealtime
A6896
A6897
A6898
A6899 /**
A6900  * Simultaneous measurement of both system clocks, and synchronization status for .
A6901  * Topic Usage:
A6902  * - TOPIC_DEVICE_CLOCK_STATUS
A6903  */
A6904 @topic
A6905 @extensibility(APPENDABLE)
A6906 struct ClockStatus {
A6907
A6908     /**
A6909      * The device described by this structure.
A6910      */
A6911     @key
A6912     Fingerprint deviceId;
A6913
A6914     /**
A6915      * Measured monotonic time, sampled simultaneously with realtime.
A6916      */
A6917     ClockMonotonic monotonic;
A6918
A6919     /**
A6920      * Indicate whether the monotonic clock initializes to 0 or accumulates operating time across restarts.
A6921      */
A6922     ClockMonotonicPersistenceType monotonicPersistence;
A6923
A6924     /**
A6925      * Measured real time, sampled simultaneously with monotonic.
A6926      */
A6927     ClockRealtime realtime;
A6928
A6929     /**
A6930      * Source of the realtime clock, as enumerated for timeSource in IEEE-1588-2008 Table 7.
A6931      */
A6932     octet realtimeSource;
A6933
A6934     /**
A6935      * Last time the realtime clock was updated by the source.
A6936      */
A6937     ClockRealtime realtimeSetTime;
A6938 }; // end struct ClockStatus
A6939
A6940 /** Direct connection to a calibrated atomic clock. */
A6941 const octet CLOCK_ATOMIC = 0x10;
A6942
A6943 /** Satellite-based system. */
A6944 const octet CLOCK_GPS = 0x20;
A6945
A6946 /** Terrestrial radio system. */
A6947 const octet CLOCK_RADIO = 0x30;
A6948
A6949 /** PTP, IEEE-1588. */
A6950 const octet CLOCK_PTP = 0x40;
A6951
A6952 /** NTP, RFC 5905 or SNTP, RFC 4330. */
A6953 const octet CLOCK_NTP = 0x50;
A6954
A6955 /** User input. */
A6956 const octet CLOCK_HAND = 0x60;
A6957
A6958 /** Other source. */
A6959 const octet CLOCK_OTHER = 0x90;
A6960
A6961 /** Arbitrary or unknown epoch. */
A6962 const octet CLOCK_INTERNAL = 0xA0;
A6963
A6964
A6965
A6966 /**
A6967  * Allows Request / Reply topics to be paired together.
A6968  */
A6969 @nested
A6970 @extensibility(APPENDABLE)
A6971 struct RequestSequence {
A6972
A6973     /**
A6974      * Requester assigned sequence number sent in a request and returned in a reply.
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A6975     */
A6976     uint64 sequenceNumber;
A6977 }; // end struct RequestSequence
A6978
A6979
A6980
A6981 /**
A6982  * Key value for a request that targets one power device configuration.
A6983  */
A6984 @nested
A6985 @extensibility(APPENDABLE)
A6986 struct DeviceConfigRequest {
A6987
A6988     /**
A6989      * Identity of the device that sent this request.
A6990      */
A6991     Fingerprint requestingDeviceId;
A6992
A6993     /**
A6994      * Identity of the device that should receive this request.
A6995      */
A6996     Fingerprint targetDeviceId;
A6997
A6998     /**
A6999      * Power device configuration that this request belongs to.
A7000      */
A7001     ConfigId config;
A7002 }; // end struct DeviceConfigRequest
A7003
A7004
A7005
A7006 /**
A7007  * Key value for a request that targets one device.
A7008  */
A7009 @nested
A7010 @extensibility(APPENDABLE)
A7011 struct DeviceRequest {
A7012
A7013     /**
A7014      * Identity of the device that sent this request.
A7015      */
A7016     Fingerprint requestingDeviceId;
A7017
A7018     /**
A7019      * Identity of the device that should receive this request.
A7020      */
A7021     Fingerprint targetDeviceId;
A7022 }; // end struct DeviceRequest
A7023
A7024
A7025
A7026 /**
A7027  * Specify a time interval. The total duration is the sum of seconds and nanoseconds.
A7028  */
A7029 @nested
A7030 @extensibility(APPENDABLE)
A7031 struct Duration {
A7032
A7033     /**
A7034      * Time in seconds.
A7035      */
A7036     uint64 second;
A7037
A7038     /**
A7039      * Time in nanoseconds.
A7040      */
A7041     uint64 nanosecond;
A7042 }; // end struct Duration
A7043
A7044
A7045
A7046 /**
A7047  * Represent a point in a two-dimensional space.
A7048  */
A7049 @nested
A7050 @extensibility(APPENDABLE)
A7051 struct Point2D {
A7052
A7053     /**
A7054      * Horizontal coordinate.
A7055      */
A7056     float x;
A7057
A7058     /**
A7059      * Vertical coordinate.
A7060      */
A7061     float y;
A7062 }; // end struct Point2D
A7063
A7064
A7065 const unsigned long Point2DSequence_MINLEN = 0;
A7066 const unsigned long Point2DSequence_MAXLEN = 21;
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7067 /**
A7068  * A sequence of Point2D.
A7069  * Minimum length: 0
A7070  * Maximum length: 21
A7071 */
A7072 typedef sequence<Point2D, Point2DSequence_MAXLEN> Point2DSequence;
A7073
A7074
A7075
A7076 /**
A7077  * Represent a curve in a two-dimensional space.
A7078 */
A7079 @nested
A7080 @extensibility(APPENDABLE)
A7081 struct Curve2D {
A7082
A7083     /**
A7084      * Vertices on the curve.
A7085     */
A7086     Point2DSequence points;
A7087 }; // end struct Curve2D
A7088
A7089
A7090
A7091 /**
A7092  * Specify output variable y as a function of input variable x.
A7093 */
A7094 @nested
A7095 @extensibility(APPENDABLE)
A7096 struct ControlCurve {
A7097
A7098     /**
A7099      * Minimum acceptable value of y. Devices should operate above this curve.
A7100     */
A7101     Curve2D minimum;
A7102
A7103     /**
A7104      * Nominal value of y. Used as the set-point for PID controls.
A7105     */
A7106     Curve2D nominal;
A7107
A7108     /**
A7109      * Maximum acceptable value of y. Devices should operate below this curve.
A7110     */
A7111     Curve2D maximum;
A7112 }; // end struct ControlCurve
A7113
A7114
A7115
A7116 /**
A7117  * Identify a power port (often remote)
A7118 */
A7119 @nested
A7120 @extensibility(APPENDABLE)
A7121 struct PowerPortId {
A7122
A7123     /**
A7124      * Device containing the port.
A7125     */
A7126     Fingerprint deviceId;
A7127
A7128     /**
A7129      * Number shown on the device exterior to represent this port.
A7130     */
A7131     PowerPortNumber portNumber;
A7132 }; // end struct PowerPortId
A7133
A7134
A7135
A7136 /**
A7137  * Design information for the connector, rated circuit, and switching capabilities of a power port.
A7138 */
A7139 @nested
A7140 @extensibility(APPENDABLE)
A7141 struct PowerPortInfo {
A7142
A7143     /**
A7144      * Number shown on the device exterior to represent this port.
A7145     */
A7146     PowerPortNumber portNumber;
A7147
A7148     /**
A7149      * Anticipated direction of power flow.
A7150     */
A7151     PowerPortDirectionality directionality;
A7152
A7153     /**
A7154      * Type of connector built in to the port.
A7155     */
A7156     PowerConnectorType connectorType;
A7157
A7158     /**
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7159      * Polarity of the connector built in to the port.
A7160      */
A7161      PowerConnectorPolarity polarity;
A7162
A7163      /**
A7164      * Type of circuit supported by this connector.
A7165      */
A7166      PowerConnectorPhases phases;
A7167
A7168      /**
A7169      * Indicate whether this power port has a switch.
A7170      */
A7171      boolean hasSwitch;
A7172
A7173      /**
A7174      * Indicate whether this power port has an external meter.
A7175      */
A7176      boolean hasExternalMeter;
A7177
A7178      /**
A7179      * Indicate whether this power port has an internal meter.
A7180      */
A7181      boolean hasInternalMeter;
A7182
A7183      /**
A7184      * minimum (greatest output) rated operating current.
A7185      * units: Ampere (A)
A7186      */
A7187      @unit("A")
A7188      float minAmperage;
A7189
A7190      /**
A7191      * minimum (greatest output) rated operating current.
A7192      * units: Ampere (A)
A7193      */
A7194      @unit("A")
A7195      float maxAmperage;
A7196
A7197      /**
A7198      * maximum (in/out) interrupt current. Not present if no switch. Required if switch exists.
A7199      * units: Ampere (A)
A7200      */
A7201      @unit("A")
A7202      @optional
A7203      float interruptAmperage;
A7204
A7205      /**
A7206      * minimum rated operating voltage.
A7207      * units: Volt (V)
A7208      */
A7209      @unit("V")
A7210      float minVoltage;
A7211
A7212      /**
A7213      * maximum rated operating voltage.
A7214      * units: Volt (V)
A7215      */
A7216      @unit("V")
A7217      float maxVoltage;
A7218
A7219      /**
A7220      * minimum rated operating frequency, not present for DC, required for AC
A7221      * units: Volt (V)
A7222      */
A7223      @unit("V")
A7224      @optional
A7225      float minFrequency;
A7226
A7227      /**
A7228      * maximum rated operating frequency, not present for DC, required for AC.
A7229      * units: Volt (V)
A7230      */
A7231      @unit("V")
A7232      @optional
A7233      float maxFrequency;
A7234
A7235      /**
A7236      * Not present if no switch. Required if switch exists. Length 0 indicates no features.
A7237      */
A7238      SwitchFeatureSequence switchFeatures;
A7239
A7240      /**
A7241      * Not present if no switch. Required if switch exists. Length 0 indicates no features.
A7242      */
A7243      ConnectorFeatureSequence connectorFeatures;
A7244      }; // end struct PowerPortInfo
A7245
A7246
A7247      const unsigned long PowerPortInfoSequence_MINLEN = 0;
A7248      const unsigned long PowerPortInfoSequence_MAXLEN = MAX_PORTS;
A7249      /**
A7250      * A sequence of PowerPortInfo.
```



*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7251      * Minimum length: 0
A7252      * Maximum length: MAX_PORTS
A7253      */
A7254      typedef sequence<PowerPortInfo, PowerPortInfoSequence_MAXLEN> PowerPortInfoSequence;
A7255
A7256
A7257
A7258      /**
A7259       * Describe the current state and most recent transition of a power switch
A7260       */
A7261      @nested
A7262      @extensibility(APPENDABLE)
A7263      struct PowerSwitchStatus {
A7264
A7265          /**
A7266           * Indicate circuit continuity through the switch. May be measured from the switch position or from electrical ↔
A7267           * continuity. Continuity is unknown for manual switches with no monitoring capability. Continuity may be ↔
A7268           * inconsistent for multi-phase switches.
A7269           */
A7270          CircuitContinuity continuity;
A7271
A7272          /**
A7273           * Indicate that the switch has failed to respond to attempted transitions and may be stuck in its current state.↔
A7274           * Not set if a lock or other protection prevents a transition.
A7275           */
A7276          boolean transitionFault;
A7277
A7278          /**
A7279           * Indicate that a protective lock is preventing transitions from the current state.
A7280           */
A7281          boolean transitionLock;
A7282
A7283          /**
A7284           * Type of event that caused the circuit to enter its present state.
A7285           */
A7286          PowerSwitchReason lastTransition;
A7287
A7288          /**
A7289           * Device or user that caused the transition to occur.
A7290           */
A7291          Fingerprint lastTransitionActor;
A7292      }; // end struct PowerSwitchStatus
A7293
A7294
A7295
A7296      /**
A7297       * Describe the state of a grounding circuit.
A7298       */
A7299      @nested
A7300      @extensibility(APPENDABLE)
A7301      struct GroundingStatus {
A7302
A7303          /**
A7304           * Number used to represent this circuit.
A7305           */
A7306          GroundingCircuitNumber groundNumber;
A7307
A7308          /**
A7309           * Status of the grounding control switch.
A7310           */
A7311          @optional
A7312          PowerSwitchStatus control;
A7313
A7314          /**
A7315           * Status of the grounding pulse switch.
A7316           */
A7317          @optional
A7318          PowerSwitchStatus pulse;
A7319
A7320          /**
A7321           * Status of the ground fault detector. GFD_INVALID if no detector.
A7322           */
A7323          GroundFaultDetection faultDetection;
A7324      }; // end struct GroundingStatus
A7325
A7326
A7327      const unsigned long GroundingStatusSequence_MINLEN = 0;
A7328      const unsigned long GroundingStatusSequence_MAXLEN = MAX_PORTS;
A7329      /**
A7330       * A sequence of GroundingStatus.
A7331       * Minimum length: 0
A7332       * Maximum length: MAX_PORTS
A7333       */
A7334      typedef sequence<GroundingStatus, GroundingStatusSequence_MAXLEN> GroundingStatusSequence;
A7335
A7336
A7337
A7338      /**
A7339       * Design information describing a grounding circuit in a device.
A7340       */
A7341      @nested
A7342      @extensibility(APPENDABLE)
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7343 struct GroundingInfo {
A7344
A7345     /**
A7346      * Number used to represent this circuit.
A7347      */
A7348     GroundingCircuitNumber groundNumber;
A7349
A7350     /**
A7351      * Type of grounding provided by this circuit.
A7352      */
A7353     GroundingDesignType groundType;
A7354
A7355     /**
A7356      * List of ports protected by this circuit.
A7357      */
A7358     PowerPortNumberSequence protectedPorts;
A7359
A7360     /**
A7361      * Not present if no control switch. Required if control switch exists. Length 0 indicates no features.
A7362      */
A7363     SwitchFeatureSequence controlSwitchFeatures;
A7364
A7365     /**
A7366      * Not present if no pulse switch. Required if pulse switch exists. Length 0 indicates no features.
A7367      */
A7368     SwitchFeatureSequence pulseSwitchFeatures;
A7369 }; // end struct GroundingInfo
A7370
A7371
A7372 const unsigned long GroundingInfoSequence_MINLEN = 0;
A7373 const unsigned long GroundingInfoSequence_MAXLEN = MAX_PORTS;
A7374 /**
A7375  * A sequence of GroundingInfo.
A7376  * Minimum length: 0
A7377  * Maximum length: MAX_PORTS
A7378  */
A7379 typedef sequence<GroundingInfo, GroundingInfoSequence_MAXLEN> GroundingInfoSequence;
A7380
A7381
A7382
A7383 /** Maximum number of parameters that a device may contain. */
A7384 const uint32 MAX_PARAMETERS = 128;
A7385
A7386 /**
A7387  * Describe an adjustable parameter.
A7388  */
A7389 @nested
A7390 @extensibility(APPENDABLE)
A7391 struct ParameterMetadata {
A7392
A7393     /**
A7394      * Name of this parameter.
A7395      */
A7396     String1_32 name;
A7397
A7398     /**
A7399      * Units of measure. (V, A, Hz, V/Hz, ...).
A7400      */
A7401     String16 units;
A7402
A7403     /**
A7404      * Recommended lower bound. Set to -infinity if no limit, otherwise hardMinValue <= nominalMinValue <= ↩
A7405      * nominalMaxValue <= hardMaxValue.
A7406      */
A7407     float nominalMinValue;
A7408
A7409     /**
A7410      * Recommended upper bound. Set to +infinity if no limit, otherwise hardMinValue <= nominalMinValue <= ↩
A7411      * nominalMaxValue <= hardMaxValue.
A7412      */
A7413     float nominalMaxValue;
A7414
A7415     /**
A7416      * Absolute lower bound. Lower values will be rejected. Set to -infinity if no limit, otherwise hardMinValue <= ↩
A7417      * nominalMinValue <= nominalMaxValue <= hardMaxValue.
A7418      */
A7419     float hardMinValue;
A7420
A7421     /**
A7422      * Absolute upper bound. Higher values will be rejected. Set to +infinity if no limit, otherwise hardMinValue <= ↩
A7423      * nominalMinValue <= nominalMaxValue <= hardMaxValue.
A7424      */
A7425     float hardMaxValue;
A7426
A7427     /**
A7428      * Smallest change >= 0 that may have an effect. Set to 0 if no limit.
A7429      */
A7430     @min(0)
A7431     float resolution;
A7432
A7433     /**
A7434      * Enumeration labels. Each value must be at least 1 character in length. No duplicates are allowed per ↩
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7435         ParameterMetadata.
A7436     */
A7437     EnumLabelsSequence enumLabels;
A7438 }; // end struct ParameterMetadata
A7439
A7440
A7441
A7442 /**
A7443  * Provide the value for a parameter.
A7444  */
A7445 @nested
A7446 @extensibility(APPENDABLE)
A7447 struct ParameterValue {
A7448
A7449     /**
A7450      * Name of this parameter.
A7451      */
A7452     String1_32 name;
A7453
A7454     /**
A7455      * Value of this parameter, with units according to ParameterMetadata.
A7456      */
A7457     float value;
A7458 }; // end struct ParameterValue
A7459
A7460
A7461 const unsigned long ParameterValueSequence_MINLEN = 0;
A7462 const unsigned long ParameterValueSequence_MAXLEN = 128;
A7463 /**
A7464  * A sequence of ParameterValue.
A7465  * Minimum length: 0
A7466  * Maximum length: 128
A7467  */
A7468 typedef sequence<ParameterValue, ParameterValueSequence_MAXLEN> ParameterValueSequence;
A7469
A7470
A7471 const unsigned long ParameterMetadataSequence_MINLEN = 0;
A7472 const unsigned long ParameterMetadataSequence_MAXLEN = 128;
A7473 /**
A7474  * A sequence of ParameterMetadata.
A7475  * Minimum length: 0
A7476  * Maximum length: 128
A7477  */
A7478 typedef sequence<ParameterMetadata, ParameterMetadataSequence_MAXLEN> ParameterMetadataSequence;
A7479
A7480
A7481
A7482 /**
A7483  * Describe the state of a power port.
A7484  */
A7485 @nested
A7486 @extensibility(APPENDABLE)
A7487 struct PowerPortStatus {
A7488
A7489     /**
A7490      * Number shown on the device exterior to represent this port.
A7491      */
A7492     PowerPortNumber portNumber;
A7493
A7494     /**
A7495      * Indicates whether a power cable is plugged in.
A7496      */
A7497     CableSenseStatus cableStatus;
A7498
A7499     /**
A7500      * Indicates that a smart cable is attached. This value is set when a smart cable is connected and cleared when ↵
A7501      * it is disconnected.
A7502      */
A7503     @optional
A7504     Fingerprint cableId;
A7505
A7506     /**
A7507      * Indicates that this port is connected to a remote port.
A7508      */
A7509     @optional
A7510     PowerPortId connection;
A7511
A7512     /**
A7513      * Indicates the state of a switch associated with this port. Must be empty when PowerPortType.switch is NONE; ↵
A7514      * otherwise it must be full.
A7515      */
A7516     @optional
A7517     PowerSwitchStatus switchStatus;
A7518 }; // end struct PowerPortStatus
A7519
A7520
A7521 const unsigned long PowerPortStatusSequence_MINLEN = 0;
A7522 const unsigned long PowerPortStatusSequence_MAXLEN = MAX_PORTS;
A7523 /**
A7524  * A sequence of PowerPortStatus.
A7525  * Minimum length: 0
A7526  * Maximum length: MAX_PORTS
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7527  */
A7528 typedef sequence<PowerPortStatus, PowerPortStatusSequence_MAXLEN> PowerPortStatusSequence;
A7529
A7530
A7531
A7532 /**
A7533  * Specifies the operator priority for a single device or power port.
A7534  */
A7535 @nested
A7536 @extensibility(APPENDABLE)
A7537 struct OperatorPriority {
A7538
A7539     /**
A7540      * The operator priority type.
A7541      */
A7542     OperatorPriorityType priorityType;
A7543
A7544     /**
A7545      * Rank values indicate priority order, where 0 is the highest priority and increasing values are lower priority. ←
A7546      * The numericRank is used when priorityType is OPT_NUMERIC_RANK.
A7547      */
A7548     int16 numericRank;
A7549 }; // end struct OperatorPriority
A7550
A7551
A7552
A7553 /**
A7554  * Specify the device power port level operator intent.
A7555  */
A7556 @nested
A7557 @extensibility(APPENDABLE)
A7558 struct PowerPortIntent {
A7559
A7560     /**
A7561      * The device described by this structure.
A7562      */
A7563     Fingerprint deviceId;
A7564
A7565     /**
A7566      * Number shown on the device exterior to represent this port.
A7567      */
A7568     PowerPortNumber portNumber;
A7569
A7570     /**
A7571      * The operator assigned priority of this power port.
A7572      */
A7573     OperatorPriority priority;
A7574 }; // end struct PowerPortIntent
A7575
A7576
A7577
A7578 /**
A7579  * Identify a connection between power ports
A7580  */
A7581 @nested
A7582 @extensibility(APPENDABLE)
A7583 struct PowerConnectionId {
A7584
A7585     /**
A7586      * One end of the connection.
A7587      */
A7588     PowerPortId portA;
A7589
A7590     /**
A7591      * The other end of the connection.
A7592      */
A7593     PowerPortId portB;
A7594 }; // end struct PowerConnectionId
A7595
A7596
A7597
A7598 /**
A7599  * Design information for power sources.
A7600  */
A7601 @nested
A7602 @extensibility(APPENDABLE)
A7603 struct SourceInfo {
A7604
A7605     /**
A7606      * A set of features this device supports.
A7607      */
A7608     SourceFeatureSequence features;
A7609
A7610     /**
A7611      * Load sharing capabilities of this device.
A7612      */
A7613     LoadSharingInfo loadSharing;
A7614 }; // end struct SourceInfo
A7615
A7616
A7617
A7618 /**
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7619      * ActiveDiagnosticMessages
A7620      * Topic Usage:
A7621      * - TOPIC_ACTIVE_DIAGNOSTICS
A7622      */
A7623      @topic
A7624      @extensibility(APPENDABLE)
A7625      struct ActiveDiagnosticMessages {
A7626
A7627          /**
A7628           * The device described by this structure.
A7629           */
A7630          @key
A7631          Fingerprint deviceId;
A7632
A7633          /**
A7634           * Active trouble codes. A given suspect parameter can only appear once in this sequence.
A7635           */
A7636          DiagnosticTroubleCodeSequence codes;
A7637
A7638          /**
A7639           * Count of trouble codes that were truncated from the codes because the maximum sequence length was exceeded. A ↩
A7640           count of 0 indicates that all active trouble codes are listed.
A7641           */
A7642          uint32 overflow;
A7643      }; // end struct ActiveDiagnosticMessages
A7644
A7645
A7646
A7647      /**
A7648       * Report the outcome of a command that required authorization.
A7649       * Topic Usage:
A7650       * - TOPIC_AUTHORIZATION_TO_ENERGIZE_OUTCOME
A7651       */
A7652      @topic
A7653      @extensibility(APPENDABLE)
A7654      struct AuthorizationToEnergizeOutcome {
A7655
A7656          /**
A7657           * Copy of the corresponding AuthorizationToEnergizeRequest.requestId.
A7658           */
A7659          @key
A7660          DeviceRequest relatedRequestId;
A7661
A7662          /**
A7663           * Copy of the requestId from the corresponding command.
A7664           */
A7665          Fingerprint commandId;
A7666
A7667          /**
A7668           * Device that processed this authorization. Must match the corresponding command and request.
A7669           */
A7670          Fingerprint deviceId;
A7671
A7672          /**
A7673           * True if the authorization was accepted, the command will be executed, and both relatedRequestId and commandId ↩
A7674           are valid.
A7675           */
A7676          boolean accepted;
A7677
A7678          /**
A7679           * True confirms that an authorization request was received. False indicates a response due to authorization ↩
A7680           timeout, and therefore relatedRequestId is invalid.
A7681           */
A7682          boolean authReceived;
A7683
A7684          /**
A7685           * True confirms that commandId matched a pending command. False indicates the command was invalid, never ↩
A7686           received, timed out, or otherwise not pending execution.
A7687           */
A7688          boolean authCommand;
A7689
A7690          /**
A7691           * True confirms that the authorization deviceId, portNumber, and continuity matched the command.
A7692           */
A7693          boolean authLocation;
A7694
A7695          /**
A7696           * True confirms that the userId was accepted.
A7697           */
A7698          boolean authUser;
A7699
A7700          /**
A7701           * True confirms that the authorizationTime was accepted.
A7702           */
A7703          boolean authTime;
A7704
A7705          /**
A7706           * Local monotonic clock when the acceptance decision was made.
A7707           */
A7708          ClockMonotonic authMonotonic;
A7709
A7710          /**
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7711         * Local realtime clock when the acceptance decision was made.
A7712         */
A7713         ClockRealtime authRealtime;
A7714     }; // end struct AuthorizationToEnergizeOutcome
A7715
A7716
A7717
A7718 /**
A7719  * Request authorization for a manual black start command.
A7720  * Topic Usage:
A7721  * - TOPIC_AUTHORIZATION_TO_ENERGIZE_REQUEST
A7722  */
A7723 @topic
A7724 @extensibility(APPENDABLE)
A7725 struct AuthorizationToEnergizeRequest {
A7726
A7727     /**
A7728      * Identity of this request.
A7729      */
A7730     @key
A7731     GridRequest requestId;
A7732
A7733     /**
A7734      * GridRequest sequence data used to associate a request and returning reply.
A7735      */
A7736     tms::RequestSequence sequenceId;
A7737
A7738     /**
A7739      * Copy of the requestId from the command that requires authorization.
A7740      */
A7741     DeviceRequest commandId;
A7742
A7743     /**
A7744      * DeviceRequest sequence data used to associate a request and returning reply.
A7745      */
A7746     tms::RequestSequence commandSequenceId;
A7747
A7748     /**
A7749      * Copy of the deviceId from the command that requires authorization.
A7750      */
A7751     Fingerprint deviceId;
A7752
A7753     /**
A7754      * All power ports that would be energized by executing this command.
A7755      */
A7756     PowerPortNumberSequence powerPorts;
A7757 }; // end struct AuthorizationToEnergizeRequest
A7758
A7759
A7760
A7761 /**
A7762  * Provide the operator's response to an authorization request.
A7763  * Topic Usage:
A7764  * - TOPIC_AUTHORIZATION_TO_ENERGIZE_RESPONSE
A7765  */
A7766 @topic
A7767 @extensibility(APPENDABLE)
A7768 struct AuthorizationToEnergizeResponse {
A7769
A7770     /**
A7771      * Copy of the corresponding AuthorizationToEnergizeRequest.requestId.
A7772      */
A7773     @key
A7774     GridRequest relatedRequestId;
A7775
A7776     /**
A7777      * Copy of the requestId from the command that requires authorization.
A7778      */
A7779     DeviceRequest commandId;
A7780
A7781     /**
A7782      * Copy of the deviceId from the command that requires authorization.
A7783      */
A7784     Fingerprint deviceId;
A7785
A7786     /**
A7787      * Copy of the AuthorizationToEnergizeRequest.ports.
A7788      */
A7789     PowerPortNumberSequence powerPorts;
A7790
A7791     /**
A7792      * True if the command is authorized to proceed.
A7793      */
A7794     boolean accept;
A7795
A7796     /**
A7797      * True if the command must not be executed.
A7798      */
A7799     boolean deny;
A7800
A7801     /**
A7802      * Identity of the user that provided this response.
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7803      */
A7804      Fingerprint userId;
A7805
A7806      /**
A7807       * Time when the user provided this response.
A7808       */
A7809      ClockRealtime authorizationTime;
A7810 }; // end struct AuthorizationToEnergizeResponse
A7811
A7812
A7813
A7814 /**
A7815  * Change the nickname associated with a Fingerprint.
A7816  * Topic Usage:
A7817  * - TOPIC_FINGERPRINT_NICKNAME_REQUEST
A7818  */
A7819 @topic
A7820 @extensibility(APPENDABLE)
A7821 struct ChangeNicknameRequest {
A7822
A7823     /**
A7824      * Identity of this request.
A7825      */
A7826     @key
A7827     DeviceRequest requestId;
A7828
A7829     /**
A7830      * Request sequence data used to associate a request and returning reply.
A7831      */
A7832     tms::RequestSequence sequenceId;
A7833
A7834     /**
A7835      * Fingerprint that should store a new nickname.
A7836      */
A7837     Fingerprint id;
A7838
A7839     /**
A7840      * New nickname to be used for this Fingerprint.
A7841      */
A7842     String32 nickname;
A7843 }; // end struct ChangeNicknameRequest
A7844
A7845
A7846
A7847 /**
A7848  * Copy the commands from one configuration index to another.
A7849  * Topic Usage:
A7850  * - TOPIC_COPY_CONFIG_REQUEST
A7851  */
A7852 @topic
A7853 @extensibility(APPENDABLE)
A7854 struct CopyConfigRequest {
A7855
A7856     /**
A7857      * Unique request identifier, including the identity of the :MC making the request.
A7858      */
A7859     @key
A7860     DeviceConfigRequest requestId;
A7861
A7862     /**
A7863      * Request sequence data used to associate a request and returning reply.
A7864      */
A7865     tms::RequestSequence sequenceId;
A7866
A7867     /**
A7868      * Configuration index to overwrite with the source values.
A7869      */
A7870     ConfigId target;
A7871 }; // end struct CopyConfigRequest
A7872
A7873
A7874
A7875 /**
A7876  * Report the present state of all the ground circuits in a device.
A7877  * Topic Usage:
A7878  * - TOPIC_DEVICE_GROUNDING_STATUS
A7879  */
A7880 @topic
A7881 @extensibility(APPENDABLE)
A7882 struct DeviceGroundingStatus {
A7883
A7884     /**
A7885      * The device described by this structure.
A7886      */
A7887     @key
A7888     Fingerprint deviceId;
A7889
A7890     /**
A7891      * Information on each ground circuit.
A7892      */
A7893     GroundingStatusSequence grounds;
A7894 }; // end struct DeviceGroundingStatus
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7895
A7896
A7897
A7898 /**
A7899  * Small image representing a device.
A7900  * Topic Usage:
A7901  * - TOPIC_DEVICE_ICON
A7902  */
A7903 @topic
A7904 @extensibility(APPENDABLE)
A7905 struct DeviceIcon {
A7906
A7907     /**
A7908      * The device illustrated by this image.
A7909      */
A7910     @key
A7911     Fingerprint deviceId;
A7912
A7913     /**
A7914      * Type of the image format. Currently, the only supported type is 'image/png'. See the IANA Media Type list - ↩
A7915      * http://www.iana.org/assignments/media-types/media-types.xhtml.
A7916      */
A7917     String32 mimeType;
A7918
A7919     /**
A7920      * Contents of the image with a resolution of 64x64 and transparent background. The image should be clearly ↩
A7921      * visible on a white or black background.
A7922      */
A7923     OctetSequence data;
A7924 }; // end struct DeviceIcon
A7925
A7926
A7927
A7928 /**
A7929  * Specify the device level operator intent.
A7930  */
A7931 @nested
A7932 @extensibility(APPENDABLE)
A7933 struct DeviceIntent {
A7934
A7935     /**
A7936      * The device described by this structure.
A7937      */
A7938     Fingerprint deviceId;
A7939
A7940     /**
A7941      * The device Battle Short state.
A7942      */
A7943     boolean battleShort;
A7944
A7945     /**
A7946      * The operator assigned priority of the device.
A7947      */
A7948     OperatorPriority priority;
A7949 }; // end struct DeviceIntent
A7950
A7951
A7952 const unsigned long DeviceIntentSequence_MINLEN = 0;
A7953 const unsigned long DeviceIntentSequence_MAXLEN = 100;
A7954 /**
A7955  * A sequence of DeviceIntent.
A7956  * Minimum length: 0
A7957  * Maximum length: 100
A7958  */
A7959 typedef sequence<DeviceIntent, DeviceIntentSequence_MAXLEN> DeviceIntentSequence;
A7960
A7961
A7962
A7963 /**
A7964  * Request for a control parameter value.
A7965  * Topic Usage:
A7966  * - TOPIC_CONTROL_PARAMETER_REQUEST
A7967  */
A7968 @topic
A7969 @extensibility(APPENDABLE)
A7970 struct ControlParameterRequest {
A7971
A7972     /**
A7973      * Identity of this request.
A7974      */
A7975     @key
A7976     DeviceConfigRequest requestId;
A7977
A7978     /**
A7979      * Request sequence data used to associate a request and returning reply.
A7980      */
A7981     tms::RequestSequence sequenceId;
A7982
A7983     /**
A7984      * Selected parameter values to update (may be partial or full list).
A7985      */
A7986     ParameterValueSequence parameters;
```



*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A7987 }; // end struct ControlParameterRequest
A7988
A7989
A7990
A7991 /**
A7992  * Status of control parameters.
A7993  * Topic Usage:
A7994  * - TOPIC_CONTROL_PARAMETER_STATUS
A7995  */
A7996 @topic
A7997 @extensibility(APPENDABLE)
A7998 struct ControlParameterStatus {
A7999
A8000     /**
A8001      * The device described by this structure.
A8002      */
A8003     @key
A8004     Fingerprint deviceId;
A8005
A8006     /**
A8007      * Configuration that these values belong to.
A8008      */
A8009     ConfigId config;
A8010
A8011     /**
A8012      * A full list of the parameters available on this device.
A8013      */
A8014     ParameterValueSequence parameters;
A8015 }; // end struct ControlParameterStatus
A8016
A8017
A8018
A8019 /**
A8020  * Report the present state of all the power ports in a device.
A8021  * Topic Usage:
A8022  * - TOPIC_DEVICE_POWER_STATUS_LIST
A8023  */
A8024 @topic
A8025 @extensibility(APPENDABLE)
A8026 struct DevicePowerPortStatuses {
A8027
A8028     /**
A8029      * The device described by this structure.
A8030      */
A8031     @key
A8032     Fingerprint deviceId;
A8033
A8034     /**
A8035      * Information on each power port.
A8036      */
A8037     PowerPortStatusSequence powerPorts;
A8038 }; // end struct DevicePowerPortStatuses
A8039
A8040
A8041
A8042 /**
A8043  * Design information for the distribution device role.
A8044  */
A8045 @nested
A8046 @extensibility(APPENDABLE)
A8047 struct DistributionInfo {
A8048
A8049     /**
A8050      * A set of features this device supports.
A8051      */
A8052     DistributionFeatureSequence features;
A8053 }; // end struct DistributionInfo
A8054
A8055
A8056
A8057 /**
A8058  * Digital load sharing configuration parameters.
A8059  */
A8060 @nested
A8061 @extensibility(APPENDABLE)
A8062 struct DLSCfg {
A8063
A8064     /**
A8065      * List of topics for receiving DLSMeasurement data. If empty, then do not receive any data. The list must ←
A8066      contain unique values.
A8067      */
A8068     TopicList sendTopics;
A8069
A8070     /**
A8071      * List of topics for sending DLSMeasurement data. If empty, then do not send any data. The list must contain ←
A8072      unique values.
A8073      */
A8074     TopicList receiveTopics;
A8075
A8076     /**
A8077      * Gain on the real-power sharing signal. Setting this to 0 makes f_{LS} equivalent to 0.
A8078      * units: per unit (p.u.)

```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8079      */
A8080      @range(min=0, max=1)
A8081      @unit("p.u.")
A8082      float gainReal;
A8083
A8084      /**
A8085       * Gain on the reactive power sharing signal. Setting this to 0 makes v_{LS} equivalent to 0.
A8086       * units: per unit (p.u.)
A8087       */
A8088      @range(min=0, max=1)
A8089      @unit("p.u.")
A8090      float gainReactive;
A8091
A8092      /**
A8093       * Bias to the real power sharing. If empty, the bias is 0.
A8094       * units: per unit watt, per unit watt (p.u. W, p.u. W)
A8095       */
A8096      @unit("p.u.⋅W,⋅p.u.⋅W")
A8097      Curve2D biasRealPu;
A8098
A8099      /**
A8100       * Bias to the reactive power sharing. If empty, the bias is 0.
A8101       * units: per unit volt ampere reactive, per unit volt ampere reactive (p.u. var, p.u. var)
A8102       */
A8103      @unit("p.u.⋅var,⋅p.u.⋅var")
A8104      Curve2D biasReactivePu;
A8105   }; // end struct DLSSConfig
A8106
A8107
A8108
A8109      /**
A8110       * Provide a short, human-readable name for a Fingerprint.
A8111       * Topic Usage:
A8112       * - TOPIC_FINGERPRINT_NICKNAME
A8113       */
A8114      @topic
A8115      @extensibility(APPENDABLE)
A8116      struct FingerprintNickname {
A8117
A8118          /**
A8119           * The Fingerprint described by this structure.
A8120           */
A8121          @key
A8122          Fingerprint id;
A8123
A8124          /**
A8125           * Nickname given to this Fingerprint.
A8126           */
A8127          String32 nickname;
A8128   }; // end struct FingerprintNickname
A8129
A8130
A8131
A8132      /**
A8133       * Retrieve all commands stored in a configuration.
A8134       * Topic Usage:
A8135       * - TOPIC_GET_CONFIG_CONTENTS_REQUEST
A8136       */
A8137      @topic
A8138      @extensibility(APPENDABLE)
A8139      struct GetConfigContentsRequest {
A8140
A8141          /**
A8142           * Unique request identifier, including the identity of the :MC making the request.
A8143           */
A8144          @key
A8145          DeviceConfigRequest requestId;
A8146
A8147          /**
A8148           * Request sequence data used to associate a request and returning reply.
A8149           */
A8150          tms::RequestSequence sequenceId;
A8151   }; // end struct GetConfigContentsRequest
A8152
A8153
A8154
A8155      /**
A8156       * Remotely transition a grounding circuit switch.
A8157       * Topic Usage:
A8158       * - TOPIC_GROUNDING_CIRCUIT_REQUEST
A8159       */
A8160      @topic
A8161      @extensibility(APPENDABLE)
A8162      struct GroundingCommand {
A8163
A8164          /**
A8165           * Identity of this request.
A8166           */
A8167          @key
A8168          DeviceConfigRequest requestId;
A8169
A8170          /**
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8171      * Request sequence data used to associate a request and returning reply.
A8172      */
A8173      tms::RequestSequence sequenceId;
A8174
A8175      /**
A8176      * Grounding circuit containing the switch that should change state.
A8177      */
A8178      GroundingCircuitNumber groundNumber;
A8179
A8180      /**
A8181      * Desired continuity through the grounding control switch.
A8182      */
A8183      DesiredCircuitContinuity control;
A8184
A8185      /**
A8186      * Desired continuity through the grounding pulse switch.
A8187      */
A8188      DesiredCircuitContinuity pulse;
A8189 }; // end struct GroundingCommand
A8190
A8191
A8192 const unsigned long GroundingCommandSequence_MINLEN = 0;
A8193 const unsigned long GroundingCommandSequence_MAXLEN = MAX_PORTS;
A8194 /**
A8195  * A sequence array of GroundingCommand.
A8196  * Minimum length: 0
A8197  * Maximum length: MAX_PORTS
A8198  */
A8199 typedef sequence<GroundingCommand, GroundingCommandSequence_MAXLEN> GroundingCommandSequence;
A8200
A8201
A8202
A8203 /**
A8204  * Periodic indication of device availability.
A8205  * Topic Usage:
A8206  * - TOPIC_HEARTBEAT
A8207  */
A8208 @topic
A8209 @extensibility(APPENDABLE)
A8210 struct Heartbeat {
A8211
A8212      /**
A8213      * The device described by this structure.
A8214      */
A8215      @key
A8216      Fingerprint deviceId;
A8217
A8218      /**
A8219      * A counter that starts at 0 and increments by 1 for each new heartbeat.
A8220      */
A8221      uint32 sequenceNumber;
A8222 }; // end struct Heartbeat
A8223
A8224
A8225
A8226 /**
A8227  * Design information for load devices.
A8228  */
A8229 @nested
A8230 @extensibility(APPENDABLE)
A8231 struct LoadInfo {
A8232
A8233      /**
A8234      * A set of features this device supports.
A8235      */
A8236      LoadFeatureSequence features;
A8237
A8238      /**
A8239      * Minimum step size for a LoadChangeRequest. Set to NaN if not supported.
A8240      * units: watt (W)
A8241      */
A8242      @unit("W")
A8243      float loadChangeThreshold;
A8244
A8245      /**
A8246      * Minimum step size for a LoadEstimate update.
A8247      * units: watt (W)
A8248      */
A8249      @unit("W")
A8250      float loadEstimateDeadband;
A8251 }; // end struct LoadInfo
A8252
A8253
A8254
A8255 /**
A8256  * Latest values for device metric parameters.
A8257  * Topic Usage:
A8258  * - TOPIC_METRIC_PARAMETER_STATUS
A8259  */
A8260 @topic
A8261 @extensibility(APPENDABLE)
A8262 struct MetricParameterStatus {
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8263
A8264 /**
A8265  * The device described by this structure.
A8266  */
A8267 @key
A8268 Fingerprint deviceId;
A8269
A8270 /**
A8271  * A full list of read-only parameter values on this device.
A8272  */
A8273 ParameterValueSequence metricParameters;
A8274 }; // end struct MetricParameterStatus
A8275
A8276
A8277
A8278 /**
A8279  * Specify the microgrid level load sharing intent.
A8280  */
A8281 @nested
A8282 @extensibility(APPENDABLE)
A8283 struct LoadSharingIntent {
A8284
A8285     /**
A8286      * Per unit of used capacity that triggers the MC to add additional power source to the microgrid. Per unit is in the
A8287      * range 0 - 1.
A8288      * units: per unit (p.u.)
A8289      */
A8290     @range(min=0, max=1)
A8291     @unit("p.u.")
A8292     float increaseCapacityThreshold;
A8293
A8294     /**
A8295      * Per unit of used capacity that triggers the MC to remove a power source from the microgrid. Per unit is in the
A8296      * range 0 - 1 and must be less than increaseCapacityThreshold.
A8297      * units: per unit (p.u.)
A8298      */
A8299     @range(min=0, max=1)
A8300     @unit("p.u.")
A8301     float decreaseCapacityThreshold;
A8302
A8303     /**
A8304      * The time duration the decreaseCapacityThreshold must be met before the Microgrid Controller removes a power
A8305      * source from the microgrid. This duration prevents power sources from being added and removed in rapid
A8306      * succession.
A8307     */
A8308     Duration decreaseCapacityDuration;
A8309
A8310     /**
A8311      * The priority above which the Microgrid Controller is authorized to trigger load curtailment. Load curtailment
A8312      * is accomplished by stopped a load device. A value of OPT_ALWAYS_OPERATE disables load curtailment.
A8313     */
A8314     OperatorPriorityType priorityLoadCurtailmentAllowed;
A8315
A8316     /**
A8317      * The priority above which the MC is authorized to trigger port power load shedding. Port power load shedding is
A8318      * accomplished by opening a power port on a distribution device. A value of OPT_ALWAYS_OPERATE disables
A8319      * circuit load shedding.
A8320     */
A8321     OperatorPriorityType priorityPowerPortSheddingAllowed;
A8322 }; // end struct LoadSharingIntent
A8323
A8324
A8325
A8326 /**
A8327  * Periodic indication of device availability.
A8328  */
A8329 @nested
A8330 @extensibility(APPENDABLE)
A8331 struct DeviceFingerprint {
A8332
A8333     /**
A8334      * The device described by this structure.
A8335      */
A8336     Fingerprint deviceId;
A8337 }; // end struct DeviceFingerprint
A8338
A8339
A8340 const unsigned long DeviceFingerprintSequence_MINLEN = 0;
A8341 const unsigned long DeviceFingerprintSequence_MAXLEN = MAX_PORTS;
A8342 /**
A8343  * A sequence of DeviceFingerprint.
A8344  * Minimum length: 0
A8345  * Maximum length: MAX_PORTS
A8346  */
A8347 typedef sequence<DeviceFingerprint, DeviceFingerprintSequence_MAXLEN> DeviceFingerprintSequence;
A8348
A8349
A8350
A8351 /**
A8352  * Specify the storage device intent.
A8353  */
A8354 @nested
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8355 @extensibility(APPENDABLE)
A8356 struct StorageIntent {
A8357
A8358     /**
A8359     * The desired total state of charge to hold in reserve.
A8360     * units: per unit (p.u.)
A8361     */
A8362     @range(min=0, max=1)
A8363     @unit("p.u.")
A8364     float reservedStateOfCharge;
A8365
A8366     /**
A8367     * Time till the reservedStateOfCharge is reached and maintained.
A8368     */
A8369     ClockMonotonic timeTillReservedSoc;
A8370
A8371     /**
A8372     * The priority above which the MC is authorized to trigger load shedding. Load shedding is accomplished by ↵
A8373     * stopped a smart load device. A value of OPT_ALWAYS_OPERATE disables load curtailment.
A8374     */
A8375     OperatorPriorityType priorityLoadSheddingAllowed;
A8376
A8377     /**
A8378     * The priority above which the MC is authorized to trigger power port load curtailment. Power port shedding is ↵
A8379     * accomplished by opening a power port on a distribution device. A value of OPT_ALWAYS_OPERATE disables ↵
A8380     * circuit load shedding.
A8381     */
A8382     OperatorPriorityType priorityPowerPortSheddingAllowed;
A8383
A8384     /**
A8385     * The list of storage devices that should reach the reservedStateOfCharge. At empty list indicates all ↵
A8386     * available storage devices.
A8387     */
A8388     DeviceFingerprintSequence reservedDeviceIds;
A8389 }; // end struct StorageIntent
A8390
A8391
A8392
A8393 /**
A8394 * Specify the microgrid level operator intent.
A8395 */
A8396 @nested
A8397 @extensibility(APPENDABLE)
A8398 struct MicrogridIntent {
A8399
A8400     /**
A8401     * Overall operating mode for the microgrid.
A8402     */
A8403     OperatingMode operatingModeValue;
A8404
A8405     /**
A8406     * Load sharing Microgrid Controller directives.
A8407     */
A8408     LoadSharingIntent loadSharingIntentValue;
A8409
A8410     /**
A8411     * Storage device Microgrid Controller directives.
A8412     */
A8413     StorageIntent storageIntentValue;
A8414 }; // end struct MicrogridIntent
A8415
A8416
A8417
A8418 /**
A8419 * Return the results of a MicrogridMembershipRequest.
A8420 * Topic Usage:
A8421 * - TOPIC_MICROGRID_MEMBERSHIP_OUTCOME
A8422 */
A8423 @topic
A8424 @extensibility(APPENDABLE)
A8425 struct MicrogridMembershipApproval {
A8426
A8427     /**
A8428     * Identity of this request.
A8429     */
A8430     @key
A8431     GridRequest requestId;
A8432
A8433     /**
A8434     * Copy of the corresponding MicrogridMembershipRequest.requestId.
A8435     */
A8436     GridRequest relatedRequestId;
A8437
A8438     /**
A8439     * Copy of the RequestSequence data from the processed request.
A8440     */
A8441     tms::RequestSequence requestSequenceId;
A8442
A8443     /**
A8444     * Device or platform that has been approved.
A8445     */
A8446     Fingerprint deviceId;
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8447
A8448      /**
A8449       * Requested state.
A8450       */
A8451      MicrogridMembership membership;
A8452
A8453      /**
A8454       * Indicate whether the microgrid is prepared to complete this request.
A8455       */
A8456      MicrogridMembershipResult result;
A8457
A8458      /**
A8459       * Short, human-readable text description of a blocked request. This should summarize to the operator what the ↵
A8460       * issue is. For example, blocked by interruptible load, blocked by critical load, or manual action required.
A8461       */
A8462      String32 hint;
A8463 }; // end struct MicrogridMembershipApproval
A8464
A8465
A8466
A8467 /**
A8468  * Request preparations to cleanly join or leave the microgrid.
A8469  * Topic Usage:
A8470  * - TOPIC_MICROGRID_MEMBERSHIP_REQUEST
A8471  */
A8472 @topic
A8473 @extensibility(APPENDABLE)
A8474 struct MicrogridMembershipRequest {
A8475
A8476      /**
A8477       * Identity of this request.
A8478       */
A8479      @key
A8480      GridRequest requestId;
A8481
A8482      /**
A8483       * Request sequence data used to associate a request and returning reply.
A8484       */
A8485      tms::RequestSequence sequenceId;
A8486
A8487      /**
A8488       * Device or platform requesting the change. A platform ID indicates that all devices on the platform are ↵
A8489       * affected.
A8490       */
A8491      Fingerprint requestingId;
A8492
A8493      /**
A8494       * The desired membership state.
A8495       */
A8496      MicrogridMembership membership;
A8497 }; // end struct MicrogridMembershipRequest
A8498
A8499
A8500 const unsigned long PowerPortIntentSequence_MINLEN = 0;
A8501 const unsigned long PowerPortIntentSequence_MAXLEN = 4000;
A8502 /**
A8503  * A sequence of PowerPortIntent.
A8504  * Minimum length: 0
A8505  * Maximum length: 4000
A8506  */
A8507 typedef sequence<PowerPortIntent, PowerPortIntentSequence_MAXLEN> PowerPortIntentSequence;
A8508
A8509
A8510
A8511 /**
A8512  * Specifies a complete set of operator directives used by the MC to control the microgrid.
A8513  */
A8514 @nested
A8515 @extensibility(APPENDABLE)
A8516 struct OperatorIntent {
A8517
A8518      /**
A8519       * The device that defined this structure.
A8520       */
A8521      @key
A8522      GridRequest requestId;
A8523
A8524      /**
A8525       * The operator intent type.
A8526       */
A8527      OperatorIntentType intentType;
A8528
A8529      /**
A8530       * The microgrid level directives.
A8531       */
A8532      MicrogridIntent microgrid;
A8533
A8534      /**
A8535       * The device level directives.
A8536       */
A8537      DeviceIntentSequence devices;
A8538
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8539      /**
A8540       * The power port level directives.
A8541      */
A8542      PowerPortIntentSequence powerPorts;
A8543  }; // end struct OperatorIntent
A8544
A8545
A8546
A8547  /**
A8548   * Used to request activation of an OperatorIntent.
A8549   * Topic Usage:
A8550   * - TOPIC_OPERATOR_INTENT_REQUEST
A8551   */
A8552  @topic
A8553  @extensibility(APPENDABLE)
A8554  struct OperatorIntentRequest {
A8555
A8556      /**
A8557       * The device sending this request.
A8558      */
A8559      @key
A8560      GridRequest requestId;
A8561
A8562      /**
A8563       * Request sequence data used to associate a request and returning reply.
A8564      */
A8565      tms::RequestSequence sequenceId;
A8566
A8567      /**
A8568       * The OperatorIntent to activate.
A8569      */
A8570      OperatorIntent desiredOperatorIntent;
A8571  }; // end struct OperatorIntentRequest
A8572
A8573
A8574
A8575  /**
A8576   * The active OperatorIntent.
A8577   * Topic Usage:
A8578   * - TOPIC_OPERATOR_INTENT_STATE
A8579   */
A8580  @topic
A8581  @extensibility(APPENDABLE)
A8582  struct OperatorIntentState {
A8583
A8584      /**
A8585       * The device described by this structure.
A8586      */
A8587      @key
A8588      Fingerprint deviceId;
A8589
A8590      /**
A8591       * The complete active OperatorIntent.
A8592      */
A8593      OperatorIntent activeOperatorIntent;
A8594  }; // end struct OperatorIntentState
A8595
A8596
A8597
A8598  /**
A8599   * Key value for a request that targets one power port of a device.
A8600   */
A8601  @nested
A8602  @extensibility(APPENDABLE)
A8603  struct PowerPortConfigRequest {
A8604
A8605      /**
A8606       * Identity of the device that sent this request.
A8607      */
A8608      Fingerprint requestingDeviceId;
A8609
A8610      /**
A8611       * Identity of the device that should received this request.
A8612      */
A8613      Fingerprint targetDeviceId;
A8614
A8615      /**
A8616       * Power device configuration that this request belongs to.
A8617      */
A8618      ConfigId config;
A8619
A8620      /**
A8621       * Power port number that should change state.
A8622      */
A8623      PowerPortNumber portNumber;
A8624  }; // end struct PowerPortConfigRequest
A8625
A8626
A8627
A8628  /**
A8629   * Identify links between devices
A8630   */
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8631 @nested
A8632 @extensibility(APPENDABLE)
A8633 struct PowerConnection {
A8634
A8635     /**
A8636      * Connection described by this object.
A8637      */
A8638     PowerConnectionId connectionId;
A8639
A8640     /**
A8641      * Technique used to detect this connection.
A8642      */
A8643     PowerConnectionDetectionType detectionType;
A8644
A8645     /**
A8646      * Device or user that detected this connection.
A8647      */
A8648     Fingerprint detectionSource;
A8649
A8650     /**
A8651      * Confidence in the link detection. -1 = confident the link does not exist, 0 = no opinion, 1 = confident the link exists.
A8652      */
A8653     @range(min=-1, max=1)
A8654     float detectionConfidence;
A8655 }; // end struct PowerConnection
A8656
A8657
A8658 const unsigned long PowerConnectionSequence_MINLEN = 0;
A8659 const unsigned long PowerConnectionSequence_MAXLEN = 200;
A8660 /**
A8661  * A sequence of PowerConnection.
A8662  * Minimum length: 0
A8663  * Maximum length: 200
A8664  */
A8665 typedef sequence<PowerConnection, PowerConnectionSequence_MAXLEN> PowerConnectionSequence;
A8666
A8667
A8668
A8669 /**
A8670  * List all power connections known to a device or user
A8671  * Topic Usage:
A8672  * - TOPIC_OPERATOR_CONNECTION_LIST
A8673  * - TOPIC_DISCOVERED_CONNECTION_LIST
A8674  * - TOPIC_MICROGRID_CONNECTION_LIST
A8675  */
A8676 @topic
A8677 @extensibility(APPENDABLE)
A8678 struct PowerConnectionList {
A8679
A8680     /**
A8681      * The reporting device or user.
A8682      */
A8683     @key
A8684     Fingerprint deviceId;
A8685
A8686     /**
A8687      * List of power connections being reported by this device or user.
A8688      */
A8689     PowerConnectionSequence connections;
A8690 }; // end struct PowerConnectionList
A8691
A8692
A8693
A8694 /**
A8695  * Remotely transition a power switch.
A8696  * Topic Usage:
A8697  * - TOPIC_POWER_SWITCH_REQUEST
A8698  */
A8699 @topic
A8700 @extensibility(APPENDABLE)
A8701 struct PowerSwitchCommand {
A8702
A8703     /**
A8704      * Identity of this request.
A8705      */
A8706     @key
A8707     PowerPortConfigRequest requestId;
A8708
A8709     /**
A8710      * Request sequence data used to associate a request and returning reply.
A8711      */
A8712     tms::RequestSequence sequenceId;
A8713
A8714     /**
A8715      * Desired continuity through the switch.
A8716      */
A8717     DesiredCircuitContinuity continuity;
A8718
A8719     /**
A8720      * Minimum synchronization RMS voltage.
A8721      * units: Volt (V)
A8722
```



*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8723      */
A8724      @unit("V")
A8725      float minV;
A8726
A8727      /**
A8728       * Maximum synchronization RMS voltage.
A8729       * units: Volt (V)
A8730       */
A8731      @unit("V")
A8732      float maxV;
A8733
A8734      /**
A8735       * Minimum synchronization frequency.
A8736       * units: hertz (Hz)
A8737       */
A8738      @unit("Hz")
A8739      float minF;
A8740
A8741      /**
A8742       * Maximum synchronization frequency.
A8743       * units: hertz (Hz)
A8744       */
A8745      @unit("Hz")
A8746      float maxF;
A8747
A8748      /**
A8749       * Maximum phase misalignment (absolute value).
A8750       * units: radian (rad)
A8751       */
A8752      @unit("rad")
A8753      float maxPhase;
A8754      }; // end struct PowerSwitchCommand
A8755
A8756
A8757      const unsigned long PowerSwitchCommandSequence_MINLEN = 0;
A8758      const unsigned long PowerSwitchCommandSequence_MAXLEN = MAX_PORTS;
A8759      /**
A8760       * A sequence of PowerSwitchCommand.
A8761       * Minimum length: 0
A8762       * Maximum length: MAX_PORTS
A8763       */
A8764      typedef sequence<PowerSwitchCommand, PowerSwitchCommandSequence_MAXLEN> PowerSwitchCommandSequence;
A8765
A8766
A8767
A8768      /**
A8769       * Topic information. Defines which communications interfaces a given device implements.
A8770       */
A8771      @nested
A8772      @extensibility(APPENDABLE)
A8773      struct TopicInfo {
A8774
A8775          /**
A8776           * Release version of data topics and types.
A8777           */
A8778          String32 dataModelVersion;
A8779
A8780          /**
A8781           * List of conditional topics.
A8782           */
A8783          TopicConditionSequence conditionalTopics;
A8784
A8785          /**
A8786           * List of optional topic names that are implemented by this device.
A8787           */
A8788          TopicConditionSequence optionalTopics;
A8789
A8790          /**
A8791           * Number of network port connectors.
A8792           */
A8793          uint16 numNetworkPorts;
A8794      }; // end struct TopicInfo
A8795
A8796
A8797
A8798      /**
A8799       * Design information for active conversion load sharing devices.
A8800       */
A8801      @nested
A8802      @extensibility(APPENDABLE)
A8803      struct ActiveConversionInfo {
A8804
A8805          /**
A8806           * Load sharing capabilities of each power port.
A8807           */
A8808          LoadSharingInfoSequence loadSharing;
A8809      }; // end struct ActiveConversionInfo
A8810
A8811
A8812
A8813      /**
A8814       * Design information for conversion taps.
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8815  */
A8816  @nested
A8817  @extensibility(APPENDABLE)
A8818  struct ConversionTapInfo {
A8819
A8820      /**
A8821       * Tap changer position.
A8822       */
A8823      TapNumber tapId;
A8824
A8825      /**
A8826       * Phase shift between this port and the internal bus.
A8827       */
A8828      float phaseShift;
A8829
A8830      /**
A8831       * Voltage ratio between this port and the internal bus.
A8832       */
A8833      float voltageRatio;
A8834  }; // end struct ConversionTapInfo
A8835
A8836
A8837  const unsigned long ConversionTapInfoSequence_MINLEN = 1;
A8838  const unsigned long ConversionTapInfoSequence_MAXLEN = MAX_TAPS;
A8839  /**
A8840   * A sequence of ConversionTapInfo.
A8841   * Minimum length: 1
A8842   * Maximum length: MAX_TAPS
A8843   */
A8844  typedef sequence<ConversionTapInfo, ConversionTapInfoSequence_MAXLEN> ConversionTapInfoSequence;
A8845
A8846
A8847
A8848  /**
A8849   * Design information for power port conversation devices.
A8850   */
A8851  @nested
A8852  @extensibility(APPENDABLE)
A8853  struct PowerPortConversionInfo {
A8854
A8855      /**
A8856       * The power port number this structure represents.
A8857       */
A8858      PowerPortNumber portNumber;
A8859
A8860      /**
A8861       * Sequence to support tap changers
A8862       */
A8863      ConversionTapInfoSequence conversionTaps;
A8864
A8865      /**
A8866       * False requires (or causes) circuit interrupt to change, true can change under rated load.
A8867       */
A8868      boolean changeUnderLoad;
A8869  }; // end struct PowerPortConversionInfo
A8870
A8871
A8872  const unsigned long PowerPortConversionInfoSequence_MINLEN = 1;
A8873  const unsigned long PowerPortConversionInfoSequence_MAXLEN = MAX_PORTS;
A8874  /**
A8875   * A sequence of PowerPortConversionInfo.
A8876   * Minimum length: 1
A8877   * Maximum length: MAX_PORTS
A8878   */
A8879  typedef sequence<PowerPortConversionInfo, PowerPortConversionInfoSequence_MAXLEN> PowerPortConversionInfoSequence;
A8880
A8881
A8882
A8883  /**
A8884   * Design information for a passive energy converter.
A8885   */
A8886  @nested
A8887  @extensibility(APPENDABLE)
A8888  struct PassiveConversionInfo {
A8889
A8890      /**
A8891       * Information for each port on the device.
A8892       */
A8893      PowerPortConversionInfoSequence portConversion;
A8894  }; // end struct PassiveConversionInfo
A8895
A8896
A8897
A8898  /**
A8899   * Design information for power conversion devices.
A8900   */
A8901  @nested
A8902  @extensibility(APPENDABLE)
A8903  struct ConversionInfo {
A8904
A8905      /**
A8906       * A set of features this device supports.
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8907      */
A8908      ConversionFeatureSequence features;
A8909
A8910      /**
A8911       * Power regulation for active devices.
A8912       */
A8913      @optional
A8914      ActiveConversionInfo activeConversion;
A8915
A8916      /**
A8917       * Power regulation for passive devices.
A8918       */
A8919      @optional
A8920      PassiveConversionInfo passiveConversion;
A8921  }; // end struct ConversionInfo
A8922
A8923
A8924
A8925  /**
A8926   * Design information for the device.
A8927   */
A8928  @nested
A8929  @extensibility(APPENDABLE)
A8930  struct ProductInfo {
A8931
A8932      /**
A8933       * Supply type of this device.
A8934       */
A8935      NatoStockNumber nsn;
A8936
A8937      /**
A8938       * Product type of this device.
A8939       */
A8940      GlobalTradeItemNumber gtin;
A8941
A8942      /**
A8943       * Name of the device manufacturer.
A8944       */
A8945      String32 manufacturerName;
A8946
A8947      /**
A8948       * Name of the device model. Optional.
A8949       */
A8950      String32 modelName;
A8951
A8952      /**
A8953       * Manufacturer number for the device model. Optional.
A8954       */
A8955      String32 modelNumber;
A8956
A8957      /**
A8958       * Unique production number for a specific device. Large enough to contain a variety of formats, including the ↵
A8959       *   GS1 serial number (AI 21). Optional.
A8960       */
A8961      String32 serialNumber;
A8962
A8963      /**
A8964       * Software Version.
A8965       */
A8966      String32 softwareVersion;
A8967  }; // end struct ProductInfo
A8968
A8969
A8970
A8971  /**
A8972   * Design information for a microgrid dashboard.
A8973   */
A8974  @nested
A8975  @extensibility(APPENDABLE)
A8976  struct MicrogridDashboardInfo {
A8977
A8978      /**
A8979       * A set of features this device supports.
A8980       */
A8981      MicrogridDashboardFeatureSequence features;
A8982  }; // end struct MicrogridDashboardInfo
A8983
A8984
A8985
A8986  /**
A8987   * Design information for a microgrid controller.
A8988   */
A8989  @nested
A8990  @extensibility(APPENDABLE)
A8991  struct MicrogridControllerInfo {
A8992
A8993      /**
A8994       * A set of features this device supports.
A8995       */
A8996      MicrogridControllerFeatureSequence features;
A8997  }; // end struct MicrogridControllerInfo
A8998
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A8999
A9000
A9001 /** The request has succeeded. */
A9002 const uint32 REPLY_OK = 200;
A9003
A9004 /** Some value in the request was invalid. */
A9005 const uint32 REPLY_BAD_REQUEST = 400;
A9006
A9007 const uint32 REPLY_METHOD_NOT_ALLOWED = 405;
A9008
A9009 const uint32 REPLY_CONFLICT = 409;
A9010
A9011 const uint32 REPLY_GONE = 410;
A9012
A9013 const uint32 REPLY_PRECONDITION_FAILED = 412;
A9014
A9015 const uint32 REPLY_REQUEST_ENTITY_TOO_LARGE = 413;
A9016
A9017 const uint32 REPLY_INTERNAL_SERVER_ERROR = 500;
A9018
A9019 const uint32 REPLY_NOT_IMPLEMENTED = 501;
A9020
A9021 const uint32 REPLY_SERVICE_UNAVAILABLE = 503;
A9022
A9023 /** Request is valid, and authorization is required before processing. */
A9024 const uint32 REPLY_PENDING_AUTHORIZATION = 600;
A9025
A9026 /**
A9027  * Indicate the type of success or failure of a request.
A9028  */
A9029 @nested
A9030 @extensibility(APPENDABLE)
A9031 struct ReplyStatus {
A9032
A9033     /**
A9034      * Indicator of success or failure. Intended to support automatic handling. Values from 100 to 599 are as defined↵
A9035      * in the IETF RFCs. Values outside that range are reserved for future versions of this standard. Selected ↵
A9036      * values are defined as constants.
A9037      */
A9038     uint32 code;
A9039
A9040     /**
A9041      * Short textual description of the status code intended for human operators.
A9042      */
A9043     String32 reason;
A9044 }; // end struct ReplyStatus
A9045
A9046
A9047
A9048 /**
A9049  * The outcome of a request. Supported requests include GridRequest, DeviceRequest, DeviceConfigRequest, and ↵
A9050  * PortConfigRequest.
A9051  * Topic Usage:
A9052  * - TOPIC_REQUEST_RESPONSE
A9053  */
A9054 @topic
A9055 @extensibility(APPENDABLE)
A9056 struct Reply {
A9057
A9058     /**
A9059      * Identity of the device that sent this request.
A9060      */
A9061     @key
A9062     Fingerprint requestingDeviceId;
A9063
A9064     /**
A9065      * Identity of the device that received the request. Default to NULL Fingerprint.
A9066      */
A9067     @key
A9068     Fingerprint targetDeviceId;
A9069
A9070     /**
A9071      * Power device configuration that this request belongs to. Default to 0.
A9072      */
A9073     @key
A9074     ConfigId config;
A9075
A9076     /**
A9077      * Power port number that should change state. Default to 0.
A9078      */
A9079     @key
A9080     PowerPortNumber portNumber;
A9081
A9082     /**
A9083      * Copy of the RequestSequence data from the processed request.
A9084      */
A9085     RequestSequence requestSequenceId;
A9086
A9087     /**
A9088      * Indication of success or failure.
A9089      */
A9090     ReplyStatus status;
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9091 }; // end struct Reply
A9092
A9093
A9094
A9095 /**
A9096  * Change source state.
A9097  * Topic Usage:
A9098  * - TOPIC_SOURCE_TRANSITION_REQUEST
A9099  */
A9100 @topic
A9101 @extensibility(APPENDABLE)
A9102 struct SourceTransitionRequest {
A9103
A9104     /**
A9105      * Identity of this request.
A9106      */
A9107     @key
A9108     DeviceConfigRequest requestId;
A9109
A9110     /**
A9111      * Request sequence data used to associate a request and returning reply.
A9112      */
A9113     tms::RequestSequence sequenceId;
A9114
A9115     /**
A9116      * Transition the device should complete.
A9117      */
A9118     SourceTransition desiredTransition;
A9119 }; // end struct SourceTransitionRequest
A9120
A9121
A9122
A9123 /**
A9124  * Report the progress of transitions.
A9125  * Topic Usage:
A9126  * - TOPIC_SOURCE_TRANSITION_STATE
A9127  */
A9128 @topic
A9129 @extensibility(APPENDABLE)
A9130 struct SourceTransitionState {
A9131
A9132     /**
A9133      * The device described by this structure.
A9134      */
A9135     @key
A9136     Fingerprint deviceId;
A9137
A9138     /**
A9139      * Present location in the state transition model.
A9140      */
A9141     SourceState presentState;
A9142
A9143     /**
A9144      * State that the device is transitioning to, or presentState if no transition.
A9145      */
A9146     SourceState futureState;
A9147
A9148     /**
A9149      * How long the device has been in the present state or transition.
A9150      * units: second (s)
A9151      */
A9152     @unit("s")
A9153     float elapsedTime;
A9154
A9155     /**
A9156      * Estimates when the present transition will complete. It is zero (0.0) when the device is not in transition.
A9157      * units: second (s)
A9158      */
A9159     @unit("s")
A9160     float remainingTime;
A9161
A9162     /**
A9163      * Copy of the most recent SourceTransitionRequest.requestId.
A9164      */
A9165     DeviceConfigRequest relatedRequestId;
A9166 }; // end struct SourceTransitionState
A9167
A9168
A9169
A9170 /**
A9171  * Identify which :MC currently has write access to the standard configurations.
A9172  * Topic Usage:
A9173  * - TOPIC_STANDARD_CONFIG_MASTER
A9174  */
A9175 @topic
A9176 @extensibility(APPENDABLE)
A9177 struct StandardConfigMaster {
A9178
A9179     /**
A9180      * The device described by this structure.
A9181      */
A9182     @key
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9183     Fingerprint deviceId;
A9184
A9185     /**
A9186      * The MC that currently controls this device. If no MC is selected or available, use the NULL Fingerprint value ←
A9187      * (all 0s)
A9188      */
A9189     Fingerprint masterId;
A9190 }; // end struct StandardConfigMaster
A9191
A9192
A9193 module ac {
A9194
A9195     const unsigned long PhaseAnglesSequence_MINLEN = 0;
A9196     const unsigned long PhaseAnglesSequence_MAXLEN = 3;
A9197     /**
A9198      * A sequence of floats representing phase angles.
A9199      * Minimum length: 0
A9200      * Maximum length: 3
A9201      */
A9202     typedef sequence<float, PhaseAnglesSequence_MAXLEN> PhaseAnglesSequence;
A9203
A9204
A9205
A9206     /**
A9207      * Measurements for a single line.
A9208      */
A9209     @nested
A9210     @extensibility(APPENDABLE)
A9211     struct PowerMeasurementLine {
A9212
A9213         /**
A9214          * RMS voltage.
A9215          * units: Volt (V)
A9216          */
A9217         @unit("V")
A9218         float v;
A9219
A9220         /**
A9221          * RMS current.
A9222          * units: Ampere (A)
A9223          */
A9224         @unit("A")
A9225         float i;
A9226
A9227         /**
A9228          * Power factor angle (phase offset) between the voltage and current.
A9229          * units: degrees (degrees)
A9230          */
A9231         @range(min=-180, max=180)
A9232         @unit("degrees")
A9233         float phi;
A9234
A9235         /**
A9236          * Active RMS power.
A9237          * units: watt (W)
A9238          */
A9239         @unit("W")
A9240         float p;
A9241
A9242         /**
A9243          * Reactive RMS power.
A9244          * units: volt ampere reactive (var)
A9245          */
A9246         @unit("var")
A9247         float q;
A9248
A9249         /**
A9250          * Dominant line frequency.
A9251          * units: hertz (Hz)
A9252          */
A9253         @min(0)
A9254         @unit("Hz")
A9255         float f;
A9256     }; // end struct PowerMeasurementLine
A9257
A9258
A9259     const unsigned long PowerMeasurementLineSequence_MINLEN = 1;
A9260     const unsigned long PowerMeasurementLineSequence_MAXLEN = 4;
A9261     /**
A9262      * A sequence of PowerMeasurementLine.
A9263      * Minimum length: 1
A9264      * Maximum length: 4
A9265      */
A9266     typedef sequence<PowerMeasurementLine, PowerMeasurementLineSequence_MAXLEN> PowerMeasurementLineSequence;
A9267
A9268
A9269
A9270     /**
A9271      * All measurements for an AC system.
A9272      */
A9273     @nested
A9274     @extensibility(APPENDABLE)
```

# TMS Data Model

## DRAFT 19 Apr 2021

```

A9275 struct PowerMeasurement {
A9276
A9277     /**
A9278      * Measured values for all applicable AC phases. The actual length depends on the power ports ↩
A9279      * PowerConnectorPhases. PHASE_SINGLE=1, PHASE_SPLIT=2, PHASE_3WYE=4, and PHASE_3DELTA=3.
A9280      */
A9281     PowerMeasurementLineSequence phases;
A9282
A9283     /**
A9284      * Measured phase difference between the voltages for all applicable phase pairs. The actual length depends ↩
A9285      * on the power ports PowerConnectorPhases.
A9286      * units: degrees (degrees)
A9287      */
A9288     @unit("degrees")
A9289     PhaseAnglesSequence phaseAngles;
A9290 }; // end struct PowerMeasurement
A9291
A9292
A9293
A9294 /**
A9295  * Power measurement for an alternating current port.
A9296  */
A9297 @nested
A9298 @extensibility(APPENDABLE)
A9299 struct PowerPortMeasurement {
A9300
A9301     /**
A9302      * Port being measured.
A9303      */
A9304     tms::PowerPortNumber portNumber;
A9305
A9306     /**
A9307      * Power measurement for this port.
A9308      */
A9309     PowerMeasurement measurement;
A9310 }; // end struct PowerPortMeasurement
A9311
A9312
A9313 const unsigned long PowerPortMeasurementSequence_MINLEN = 0;
A9314 const unsigned long PowerPortMeasurementSequence_MAXLEN = tms::MAX_PORTS;
A9315 /**
A9316  * A sequence of ac::PowerPortMeasurement.
A9317  * Minimum length: 0
A9318  * Maximum length: tms::MAX_PORTS
A9319  */
A9320 typedef sequence<PowerPortMeasurement, PowerPortMeasurementSequence_MAXLEN> PowerPortMeasurementSequence;
A9321
A9322
A9323
A9324 /**
A9325  * Report power monitoring for an alternating current device
A9326  * Topic Usage:
A9327  * - TOPIC_DEVICE_POWER_MEASUREMENT_LIST
A9328  */
A9329 @topic
A9330 @extensibility(APPENDABLE)
A9331 struct DevicePowerMeasurements {
A9332
A9333     /**
A9334      * The device described by this structure.
A9335      */
A9336     @key
A9337     tms::Fingerprint deviceId;
A9338
A9339     /**
A9340      * Time of these measurements.
A9341      */
A9342     tms::ClockMonotonic timeMeasured;
A9343
A9344     /**
A9345      * Measurements for all power ports on this device.
A9346      */
A9347     PowerPortMeasurementSequence powerPorts;
A9348 }; // end struct DevicePowerMeasurements
A9349
A9350
A9351
A9352 /**
A9353  * Represents the load sharing parameters for alternating current devices.
A9354  */
A9355 @nested
A9356 @extensibility(APPENDABLE)
A9357 struct LoadSharing {
A9358
A9359     /**
A9360      * Real power to frequency relationship curve, f_P.
A9361      * units: watt, hertz (W, Hz)
A9362      */
A9363     @unit("W, Hz")
A9364     tms::ControlCurve frequencyPowerCurve;
A9365
A9366     /**

```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9367      * Reactive power to voltage relationship curve, v_Q.
A9368      * units: volt ampere reactive, volts (var, V)
A9369      */
A9370      @unit("var, V")
A9371      tms::ControlCurve voltageReactiveCurve;
A9372
A9373      /**
A9374      * Digital load sharing configuration.
A9375      */
A9376      tms::DLSCConfig dls;
A9377  }; // end struct LoadSharing
A9378
A9379
A9380
A9381  /**
A9382  * Sets load sharing parameters for alternating current devices.
A9383  * Topic Usage:
A9384  * - TOPIC_LOAD_SHARING_REQUEST
A9385  */
A9386  @topic
A9387  @extensibility(APPENDABLE)
A9388  struct LoadSharingRequest {
A9389
A9390      /**
A9391      * Identity of this request.
A9392      */
A9393      @key
A9394      tms::PowerPortConfigRequest requestId;
A9395
A9396      /**
A9397      * Request sequence data used to associate a request and returning reply.
A9398      */
A9399      tms::RequestSequence sequenceId;
A9400
A9401      /**
A9402      * New value to use.
A9403      */
A9404      LoadSharing parameters;
A9405  }; // end struct LoadSharingRequest
A9406
A9407
A9408
A9409  /**
A9410  * Report the present value of the load sharing parameters for alternating current devices.
A9411  * Topic Usage:
A9412  * - TOPIC_LOAD_SHARING_STATUS
A9413  */
A9414  @topic
A9415  @extensibility(APPENDABLE)
A9416  struct LoadSharingStatus {
A9417
A9418      /**
A9419      * The device described by this structure.
A9420      */
A9421      @key
A9422      tms::Fingerprint deviceId;
A9423
A9424      /**
A9425      * Configuration that these values belong to.
A9426      */
A9427      tms::ConfigId config;
A9428
A9429      /**
A9430      * Parameter values.
A9431      */
A9432      LoadSharing parameters;
A9433  }; // end struct LoadSharingStatus
A9434
A9435  }; //end module ac
A9436
A9437  module dc {
A9438
A9439
A9440      /**
A9441      * Power measurement for direct current devices.
A9442      */
A9443      @nested
A9444      @extensibility(APPENDABLE)
A9445      struct PowerMeasurement {
A9446
A9447          /**
A9448          * Average power.
A9449          * units: watt (W)
A9450          */
A9451          @unit("W")
A9452          float power;
A9453
A9454          /**
A9455          * Voltage from DC+ to DC-.
A9456          * units: Volt (V)
A9457          */
A9458          @unit("V")
```



*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9459         float v;
A9460
A9461         /**
A9462          * Current from DC+ to DC-.
A9463          * units: Ampere (A)
A9464          */
A9465         @unit("A")
A9466         float i;
A9467
A9468         /**
A9469          * Voltage from DC- to ground.
A9470          * units: Volt (V)
A9471          */
A9472         @unit("V")
A9473         float vGround;
A9474     }; // end struct PowerMeasurement
A9475
A9476
A9477
A9478     /**
A9479     * Power measurement for a direct current power port.
A9480     */
A9481     @nested
A9482     @extensibility(APPENDABLE)
A9483     struct PowerPortMeasurement {
A9484
A9485         /**
A9486          * Port being measured.
A9487          */
A9488         tms::PowerPortNumber portNumber;
A9489
A9490         /**
A9491          * Power measurement for this power port.
A9492          */
A9493         PowerMeasurement measurement;
A9494     }; // end struct PowerPortMeasurement
A9495
A9496
A9497     const unsigned long PowerPortMeasurementSequence_MINLEN = 0;
A9498     const unsigned long PowerPortMeasurementSequence_MAXLEN = tms::MAX_PORTS;
A9499     /**
A9500     * A sequence of dc:PowerPortMeasurement.
A9501     * Minimum length: 0
A9502     * Maximum length: tms::MAX_PORTS
A9503     */
A9504     typedef sequence<PowerPortMeasurement, PowerPortMeasurementSequence_MAXLEN> PowerPortMeasurementSequence;
A9505
A9506
A9507
A9508     /**
A9509     * Report power monitoring for a direct current device
A9510     * Topic Usage:
A9511     * - TOPIC_DC_DEVICE_POWER_MEASUREMENT_LIST
A9512     */
A9513     @topic
A9514     @extensibility(APPENDABLE)
A9515     struct DevicePowerMeasurements {
A9516
A9517         /**
A9518          * The device described by this structure.
A9519          */
A9520         @key
A9521         tms::Fingerprint deviceId;
A9522
A9523         /**
A9524          * Time of these measurements.
A9525          */
A9526         tms::ClockMonotonic timeMeasured;
A9527
A9528         /**
A9529          * Measurements for all ports on this device.
A9530          */
A9531         PowerPortMeasurementSequence powerPorts;
A9532
A9533         /**
A9534          * Measurements for the DC balance resistor.
A9535          */
A9536         tms::CircuitContinuity balanceResistorEnabled;
A9537     }; // end struct DevicePowerMeasurements
A9538
A9539
A9540
A9541     /**
A9542     * Represents the load sharing parameters for direct current devices.
A9543     */
A9544     @nested
A9545     @extensibility(APPENDABLE)
A9546     struct LoadSharing {
A9547
A9548         /**
A9549          * Minimum output voltage. Saturate if droop would go lower.
A9550          * units: Volt (V)
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9551      */
A9552      @unit("V")
A9553      float minV;
A9554
A9555      /**
A9556       * Maximum output voltage. Saturate if droop would go higher.
A9557       * units: Volt (V)
A9558       */
A9559      @unit("V")
A9560      float maxV;
A9561
A9562      /**
A9563       * Minimum output current. Saturate if droop would go lower.
A9564       * units: Ampere (A)
A9565       */
A9566      @unit("A")
A9567      float minI;
A9568
A9569      /**
A9570       * Maximum output current. Saturate if droop would go higher.
A9571       * units: Ampere (A)
A9572       */
A9573      @unit("A")
A9574      float maxI;
A9575
A9576      /**
A9577       * Voltage to current droop curve.
A9578       * units: volt, ampere (V, A)
A9579       */
A9580      @unit("V, A")
A9581      tms::ControlCurve voltageDroop;
A9582 }; // end struct LoadSharing
A9583
A9584
A9585      /**
A9586      * Sets load sharing parameters for direct current devices.
A9587      * Topic Usage:
A9588      * - TOPIC_DC_LOAD_SHARING_REQUEST
A9589      */
A9590      @topic
A9591      @extensibility(APPENDABLE)
A9592      struct LoadSharingRequest {
A9593
A9594          /**
A9595           * Identity of this request.
A9596           */
A9597          @key
A9598          tms::PowerPortConfigRequest requestId;
A9599
A9600          /**
A9601           * Request sequence data used to associate a request and returning reply.
A9602           */
A9603          tms::RequestSequence sequenceId;
A9604
A9605          /**
A9606           * New value to use.
A9607           */
A9608          LoadSharing parameters;
A9609      }; // end struct LoadSharingRequest
A9610
A9611
A9612      /**
A9613      * Report the present value of the load sharing parameters for direct current devices.
A9614      * Topic Usage:
A9615      * - TOPIC_DC_LOAD_SHARING_STATUS
A9616      */
A9617      @topic
A9618      @extensibility(APPENDABLE)
A9619      struct LoadSharingStatus {
A9620
A9621          /**
A9622           * The device described by this structure.
A9623           */
A9624          @key
A9625          tms::Fingerprint deviceId;
A9626
A9627          /**
A9628           * Configuration that these values belong to.
A9629           */
A9630          tms::ConfigId config;
A9631
A9632          /**
A9633           * Parameter values.
A9634           */
A9635          LoadSharing parameters;
A9636      }; // end struct LoadSharingStatus
A9637
A9638 }; //end module dc
A9639
A9640 module stor {
A9641
A9642
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9643
A9644
A9645 /**
A9646  * Identifies an energy storage cell within a device.
A9647  */
A9648 typedef uint16 StorageCellNumber;
A9649
A9650 /** Number to use for a single-cell device. */
A9651 const StorageCellNumber ONLY_CELL = 0;
A9652
A9653 /** Maximum number of cells that a device may report. */
A9654 const StorageCellNumber MAXLEN_storageCells = 50;
A9655
A9656
A9657 /**
A9658  * Power measurement for direct current devices.
A9659  */
A9660 @nested
A9661 @extensibility(APPENDABLE)
A9662 struct StorageCellState {
A9663
A9664     /**
A9665      * Identifies the energy storage cell being reported on within a device.
A9666      */
A9667     StorageCellNumber cellNumber;
A9668
A9669     /**
A9670      * Cumulative estimate of the energy storage wear, measured as the number of operating charge/discharge cycles↵
A9671      * . Set to NaN if not available.
A9672      */
A9673     float cycleCounter;
A9674
A9675     /**
A9676      * Estimate of present charge as a fraction of rated capacity. 0=empty and 1=full. Set to NaN if not ↵
A9677      * available.
A9678      * units: per unit (p.u.)
A9679      */
A9680     @range(min=0, max=1)
A9681     @unit("p.u.")
A9682     float stateOfCharge;
A9683
A9684     /**
A9685      * Temperature measurement of the individual storage cell. Set to NaN if not available.
A9686      * units: Celsius (C)
A9687      */
A9688     @range(min=-273.15, max=10000)
A9689     @unit("C")
A9690     float temperature;
A9691
A9692     /**
A9693      * DC bus voltage across the energy storage cell terminals. Set to NaN if not available.
A9694      * units: Volt (V)
A9695      */
A9696     @unit("V")
A9697     float voltage;
A9698 }; // end struct StorageCellState
A9699
A9700
A9701 const unsigned long StorageCellStateSequence_MINLEN = 1;
A9702 const unsigned long StorageCellStateSequence_MAXLEN = 999;
A9703 /**
A9704  * A sequence of StorageCellState.
A9705  * Minimum length: 1
A9706  * Maximum length: 999
A9707  */
A9708 typedef sequence<StorageCellState, StorageCellStateSequence_MAXLEN> StorageCellStateSequence;
A9709
A9710
A9711
A9712 /**
A9713  * Report key parameters about the state of the ESU.
A9714  * Topic Usage:
A9715  * - TOPIC_STORAGE_STATE
A9716  */
A9717 @topic
A9718 @extensibility(APPENDABLE)
A9719 struct StorageState {
A9720
A9721     /**
A9722      * The device described by this structure.
A9723      */
A9724     @key
A9725     tms::Fingerprint deviceId;
A9726
A9727     /**
A9728      * How much power the ESU is consuming for its present internal operation, whether charging, discharging, or ↵
A9729      * idle. Equals the difference between the external ESU power output and the internal energy storage power↵
A9730      * output.
A9731      * units: watt (W)
A9732      */
A9733     @unit("W")
A9734     float operatingPower;
```

# TMS Data Model

## DRAFT 19 Apr 2021

```

A9735
A9736 /**
A9737  * Cumulative time that the ESU has been actively charging or discharging. Usually displayed as an hour meter↵
A9738  .
A9739  * units: second (s)
A9740  */
A9741 @unit("s")
A9742 float runTime;
A9743
A9744 /**
A9745  * Cumulative estimate of the energy storage wear, measured as the number of operating charge/discharge cycles↵
A9746  .
A9747  */
A9748 float cycleCounter;
A9749
A9750 /**
A9751  * Measurement indicating the DC bus voltage between the energy storage and power converter.
A9752  * units: Volt (V)
A9753  */
A9754 @unit("V")
A9755 float internalVoltage;
A9756
A9757 /**
A9758  * Estimate of present charge as a fraction of rated capacity. 0=empty and 1=full. TODO: show how reserve ↵
A9759  thresholds interact, note that >1 may be possible, do not saturate.
A9760  */
A9761 @range(min=0, max=1)
A9762 float stateOfCharge;
A9763
A9764 /**
A9765  * Estimate of how long the ESU can operate at the present power rate.
A9766  * units: second (s)
A9767  */
A9768 @unit("s")
A9769 float holdTime;
A9770
A9771 /**
A9772  * Estimate of how long the ESU can charge at different power levels, without exceeding the highStateOfCharge.
A9773  * units: watt, second (W, s)
A9774  */
A9775 @unit("W,␣s")
A9776 tms::Curve2D chargeTime;
A9777
A9778 /**
A9779  * Estimate of how long the ESU can discharge at different power levels, without exceeding the ↵
A9780  lowStateOfCharge.
A9781  * units: watt, second (W, s)
A9782  */
A9783 @unit("W,␣s")
A9784 tms::Curve2D dischargeTime;
A9785
A9786 /**
A9787  * Estimate of how long the ESU can charge at different power levels, with battle short enabled. Must be ↵
A9788  greater than or equal to the chargeTime.
A9789  * units: watt, second (W, s)
A9790  */
A9791 @unit("W,␣s")
A9792 tms::Curve2D maxChargeTime;
A9793
A9794 /**
A9795  * Estimate of how long the ESU can discharge at different power levels, with battle short enabled. Must be ↵
A9796  greater than or equal to the dischargeTime.
A9797  * units: watt, second (W, s)
A9798  */
A9799 @unit("W,␣s")
A9800 tms::Curve2D maxDischargeTime;
A9801
A9802 /**
A9803  * Estimate of temperature control load as a fraction of capacity. 0=no load, 1=full rated load (hot or cold)↵
A9804  . Values above 1 indicate overload conditions.
A9805  * units: per unit (p.u.)
A9806  */
A9807 @min(0)
A9808 @unit("p.u.")
A9809 float thermalLoad;
A9810
A9811 /**
A9812  * Temperature measurement indicating the state of the power converter.
A9813  * units: Celsius (C)
A9814  */
A9815 @unit("C")
A9816 float powerConverterTemperature;
A9817
A9818 /**
A9819  * Temperature measurement indicating the overall state of the energy storage.
A9820  * units: Celsius (C)
A9821  */
A9822 @unit("C")
A9823 float energyStorageTemperature;
A9824
A9825 /**
A9826  * Information on each storage cell.

```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9827     */
A9828     StorageCellStateSequence storageCells;
A9829 }; // end struct StorageState
A9830
A9831
A9832 @extensibility(APPENDABLE)
A9833 enum StorageOperatingState {
A9834     EOS_UNKNOWN, // Uninitialized value.
A9835     EOS_STANDALONE, // Standalone operation as only source.
A9836     EOS_PARALLEL, // Parallel operation with other sources.
A9837     EOS_AUTOMATIC // Automatic detection of standalone or parallel status.
A9838 }; // enum StorageOperatingState
A9839
A9840
A9841
A9842 /**
A9843  * Stop discharging the ESU if the available generation is motoring.
A9844  */
A9845 @nested
A9846 @extensibility(APPENDABLE)
A9847 struct OverFrequencyStopDischarge {
A9848
A9849     /**
A9850      * Maximum frequency threshold. Set to NaN to disable this threshold.
A9851      * units: hertz (Hz)
A9852      */
A9853     @unit("Hz")
A9854     float frequency;
A9855
A9856     /**
A9857      * Time to wait before stopping discharge.
A9858      */
A9859     tms::Duration delay;
A9860 }; // end struct OverFrequencyStopDischarge
A9861
A9862
A9863
A9864 /**
A9865  * Stop charging the ESU if the available generation is overloaded.
A9866  */
A9867 @nested
A9868 @extensibility(APPENDABLE)
A9869 struct UnderFrequencyStopCharge {
A9870
A9871     /**
A9872      * Minimum frequency threshold. Set to NaN to disable this threshold.
A9873      * units: hertz (Hz)
A9874      */
A9875     @unit("Hz")
A9876     float frequency;
A9877
A9878     /**
A9879      * Time to wait before stopping charge.
A9880      */
A9881     tms::Duration delay;
A9882 }; // end struct UnderFrequencyStopCharge
A9883
A9884
A9885
A9886 /**
A9887  * Parameters for on-platform capabilities of the ESU controller.
A9888  */
A9889 @nested
A9890 @extensibility(APPENDABLE)
A9891 struct StorageControlParameters {
A9892
A9893     /**
A9894      * The high state of charge during continuous operations.
A9895      * units: per unit (p.u.)
A9896      */
A9897     @range(min=0, max=1)
A9898     @unit("p.u.")
A9899     float highStateOfCharge;
A9900
A9901     /**
A9902      * The low state of charge during continuous operations.
A9903      * units: per unit (p.u.)
A9904      */
A9905     @range(min=0, max=1)
A9906     @unit("p.u.")
A9907     float lowStateOfCharge;
A9908
A9909     /**
A9910      * If the ESU has not been required to discharge by more than 1kW for this period of time, then it may ↵
A9911      * automatically initiate a charge cycle to nominalCharge. Similarly, if the ESU has not been required to ↵
A9912      * discharge by more than 1kW for this period of time, then it may automatically initiate a discharge cycle ↵
A9913      * to nominalCharge. Set to +infinity to disable.
A9914      */
A9915     tms::Duration nominalChargeDelay;
A9916
A9917     /**
A9918      * (>=0) If the ESU initiates a charge/discharge cycle to nominalCharge, then the charge/discharge power must ↵
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A9919         stay below this value.
A9920         * units: watt (W)
A9921         */
A9922         @unit("W")
A9923         float nominalChargeRate;
A9924
A9925         /**
A9926         * Limit how fast the power output changes for on-platform capabilities that can be planned in advance. These
A9927         * include state of charge maintenance and power curtailment based on state of charge. This limit does
A9928         * not apply to LoadPolicy commands, UPS, overload protection, or motoring protection. Set to NaN for no
A9929         * limit.
A9930         * units: watt per second (W/s)
A9931         */
A9932         @unit("W/s")
A9933         float rampRate;
A9934
A9935         /**
A9936         * Force the ESU to operate in a fixed state, or enable automatic detection.
A9937         */
A9938         StorageOperatingState modeSelection;
A9939
A9940         /**
A9941         * Stop charging when there is sudden, severe overload.
A9942         */
A9943         UnderFrequencyStopCharge stopChargeInrush;
A9944
A9945         /**
A9946         * Stop charging when there is persistent, light overload.
A9947         */
A9948         UnderFrequencyStopCharge stopChargeSteady;
A9949
A9950         /**
A9951         * Stop discharging when there is sudden, severe motoring.
A9952         */
A9953         OverFrequencyStopDischarge stopDischargeInrush;
A9954
A9955         /**
A9956         * Stop discharging when there is persistent, light motoring.
A9957         */
A9958         OverFrequencyStopDischarge stopDischargeSteady;
A9959     }; // end struct StorageControlParameters
A9960
A9961     /**
A9962     * Set the storage control parameters.
A9963     * Topic Usage:
A9964     * - TOPIC_STORAGE_CONTROL_REQUEST
A9965     */
A9966     @topic
A9967     @extensibility(APPENDABLE)
A9968     struct StorageControlRequest {
A9969
A9970         /**
A9971         * Identity of this request.
A9972         */
A9973         @key
A9974         tms::DeviceConfigRequest requestId;
A9975
A9976         /**
A9977         * Request sequence data used to associate a request and returning reply.
A9978         */
A9979         tms::RequestSequence sequenceId;
A9980
A9981         /**
A9982         * New value to use.
A9983         */
A9984         StorageControlParameters parameters;
A9985     }; // end struct StorageControlRequest
A9986
A9987     /**
A9988     * Report the present value of the storage control parameters.
A9989     * Topic Usage:
A9990     * - TOPIC_STORAGE_CONTROL_STATUS
A9991     */
A9992     @topic
A9993     @extensibility(APPENDABLE)
A9994     struct StorageControlStatus {
A9995
A9996         /**
A9997         * The device described by this structure.
A9998         */
A9999         @key
A9999         tms::Fingerprint deviceId;
A9999
A9999         /**
A9999         * Configuration that this command belongs to.
A9999         */
A9999         tms::ConfigId config;
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A10011      /**
A10012      * Parameter values.
A10013      */
A10014      StorageControlParameters parameters;
A10015  }; // end struct StorageControlStatus
A10016
A10017  }; //end module stor
A10018
A10019
A10020  /**
A10021  * Specified a single power device configuration.
A10022  */
A10023  @nested
A10024  @extensibility(APPENDABLE)
A10025  struct PowerDeviceConfig {
A10026
A10027      /**
A10028      * A list of power switch commands.
A10029      */
A10030      PowerSwitchCommandSequence powerSwitchCommands;
A10031
A10032      /**
A10033      * A source transition request.
A10034      */
A10035      @optional
A10036      SourceTransitionRequest sourceTransition;
A10037
A10038      /**
A10039      * A control parameter request.
A10040      */
A10041      @optional
A10042      ControlParameterRequest controlParameter;
A10043
A10044      /**
A10045      * A list of grounding commands.
A10046      */
A10047      GroundingCommandSequence groundingCommands;
A10048
A10049      /**
A10050      * An alternating current load sharing request.
A10051      */
A10052      @optional
A10053      ac::LoadSharingRequest acLoadSharingRequest;
A10054
A10055      /**
A10056      * A direct current load sharing request.
A10057      */
A10058      @optional
A10059      dc::LoadSharingRequest dcLoadSharingRequest;
A10060
A10061      /**
A10062      * A storage control request.
A10063      */
A10064      @optional
A10065      stor::StorageControlRequest storageControlRequest;
A10066  }; // end struct PowerDeviceConfig
A10067
A10068
A10069
A10070  /**
A10071  * Design information for control services on microgrid controllers and dashboards.
A10072  */
A10073  @nested
A10074  @extensibility(APPENDABLE)
A10075  struct ControlServiceInfo {
A10076
A10077      /**
A10078      * Microgrid Controller information.
A10079      */
A10080      @optional
A10081      MicrogridControllerInfo mc;
A10082
A10083      /**
A10084      * Microgrid Dashboard information.
A10085      */
A10086      @optional
A10087      MicrogridDashboardInfo md;
A10088  }; // end struct ControlServiceInfo
A10089
A10090
A10091
A10092  /**
A10093  * Design information for a power device which is any TMS device role that generates, distributes or consumes power. ↵
A10094      SRC, STOR, LOAD, DIST and CONV.
A10095  */
A10096  @nested
A10097  @extensibility(APPENDABLE)
A10098  struct PowerDeviceInfo {
A10099
A10100      /**
A10101      * 1 for SRC and STOR and LOAD, 2+ for DIST and CONV
A10102      */
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A10103     PowerPortInfoSequence powerPorts;
A10104
A10105     /**
A10106      * Grounding port information.
A10107      */
A10108     GroundingInfoSequence grounds;
A10109
A10110     /**
A10111      * Conversion information.
A10112      */
A10113     @optional
A10114     ConversionInfo conversion;
A10115
A10116     /**
A10117      * Distribution device information.
A10118      */
A10119     @optional
A10120     DistributionInfo distribution;
A10121
A10122     /**
A10123      * Power source device information.
A10124      */
A10125     @optional
A10126     SourceInfo source;
A10127
A10128     /**
A10129      * Power storage device information.
A10130      */
A10131     @optional
A10132     StorageInfo storage;
A10133
A10134     /**
A10135      * Power load information.
A10136      */
A10137     @optional
A10138     LoadInfo load;
A10139 }; // end struct PowerDeviceInfo
A10140
A10141
A10142
A10143 /**
A10144  * Design information for the device.
A10145  * Topic Usage:
A10146  * - TOPIC_DEVICE_ANNOUNCEMENT
A10147  */
A10148 @topic
A10149 @extensibility(APPENDABLE)
A10150 struct DeviceInfo {
A10151
A10152     /**
A10153      * The device described by this structure.
A10154      */
A10155     @key
A10156     Fingerprint deviceId;
A10157
A10158     /**
A10159      * A platform is a group of TMS devices that are assembled into a larger physical package. Platforms may ↔
A10160      * integrate multiple device roles. Set to FINGERPRINT_INVALID if this device is not permanently mounted on a ↔
A10161      * platform.
A10162      */
A10163     Fingerprint platformId;
A10164
A10165     /**
A10166      * TMS role implemented by this device.
A10167      */
A10168     DeviceRole role;
A10169
A10170     /**
A10171      * Product information.
A10172      */
A10173     ProductInfo product;
A10174
A10175     /**
A10176      * communications interfaces implemented by this device
A10177      */
A10178     TopicInfo topics;
A10179
A10180     /**
A10181      * Information about the device controller. May apply to both control services and power devices.
A10182      */
A10183     @optional
A10184     ControlHardwareInfo controlHardware;
A10185
A10186     /**
A10187      * Information about the power hardware, if applicable.
A10188      */
A10189     @optional
A10190     PowerHardwareInfo powerHardware;
A10191
A10192     /**
A10193      * A full list of read-write control parameters available on the ControlParameterStatus topic.
A10194      */
```



# TMS Data Model

## DRAFT 19 Apr 2021

```
A10195     ParameterMetadataSequence controlParameters;
A10196
A10197     /**
A10198      * A full list of read-only metric parameters available on the MetricParameterStatus topic.
A10199      */
A10200     ParameterMetadataSequence metricParameters;
A10201
A10202     /**
A10203      * Control role-specific information.
A10204      */
A10205     @optional
A10206     ControlServiceInfo controlService;
A10207
A10208     /**
A10209      * Power device role-specific information.
A10210      */
A10211     @optional
A10212     PowerDeviceInfo powerDevice;
A10213 }; // end struct DeviceInfo
A10214
A10215
A10216
A10217 /**
A10218  * Response to a GetConfigContentsRequest.
A10219  * Topic Usage:
A10220  * - TOPIC_GET_CONFIG_CONTENTS_RESPONSE
A10221  */
A10222 @topic
A10223 @extensibility(APPENDABLE)
A10224 struct GetConfigContentsResponse {
A10225
A10226     /**
A10227      * Copy of the corresponding requestId.
A10228      */
A10229     @key
A10230     DeviceConfigRequest relatedRequestId;
A10231
A10232     /**
A10233      * Indication of success or failure.
A10234      */
A10235     ReplyStatus status;
A10236
A10237     /**
A10238      * The requested power device configuration.
A10239      */
A10240     PowerDeviceConfig powerDeviceConfigValue;
A10241 }; // end struct GetConfigContentsResponse
A10242
A10243 }; // module tms
```

## A10244 E.2 Quality of Service XML

A10245 The following is a syntax-highlighted, verbatim copy of a OMG DDS QoS XML file. This  
A10246 file is informative. It contains definitions of all the non-default policies.

A10247 Automatic line wraps are indicated by a ↵. A separate, plain-text XML file should be  
A10248 distributed with the standard.

```
A10249 <?xml version="1.0" encoding="UTF-8"?>
A10250 <dds xmlns="http://www.omg.org/spec/DDS-XML" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="↵
A10251     http://www.omg.org/spec/DDS-XML/../../xsd/omgdds/dds-xml_dds_system_definitions.xsd">
A10252     <qos_library name="TMS_3.0">
A10253         <qos_profile name="PublishLast" base_name="tms_base">
A10254             <datareader_qos>
A10255                 <deadline>
A10256                     <period>
A10257                         <sec>DURATION_INFINITE_SEC</sec>
A10258                         <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10259                     </period>
A10260                 </deadline>
A10261                 <durability>
A10262                     <kind>TRANSIENT_LOCAL_DURABILITY_QOS</kind>
A10263                 </durability>
A10264                 <history>
A10265                     <kind>KEEP_LAST_HISTORY_QOS</kind>
A10266                     <depth>1</depth>
A10267                 </history>
A10268                 <liveliness>
A10269                     <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10270                     <lease_duration>
A10271                         <sec>DURATION_INFINITE_SEC</sec>
A10272                         <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10273                     </lease_duration>
A10274                 </liveliness>
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A10275         <ownership>
A10276             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10277         </ownership>
A10278         <reliability>
A10279             <kind>RELIABLE_RELIABILITY_QOS</kind>
A10280         </reliability>
A10281     </datareader_qos>
A10282     <datawriter_qos>
A10283         <deadline>
A10284             <period>
A10285                 <sec>DURATION_INFINITE_SEC</sec>
A10286                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10287             </period>
A10288         </deadline>
A10289         <durability>
A10290             <kind>TRANSIENT_LOCAL_DURABILITY_QOS</kind>
A10291         </durability>
A10292         <history>
A10293             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10294             <depth>1</depth>
A10295         </history>
A10296         <liveliness>
A10297             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10298             <lease_duration>
A10299                 <sec>DURATION_INFINITE_SEC</sec>
A10300                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10301             </lease_duration>
A10302         </liveliness>
A10303         <ownership>
A10304             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10305         </ownership>
A10306         <ownership_strength>
A10307             <value>1</value>
A10308         </ownership_strength>
A10309         <reliability>
A10310             <kind>RELIABLE_RELIABILITY_QOS</kind>
A10311         </reliability>
A10312     </datawriter_qos>
A10313 </qos_profile>
A10314 <qos_profile name="Rare" base_name="tms_base">
A10315     <datareader_qos>
A10316         <deadline>
A10317             <period>
A10318                 <sec>2000</sec>
A10319                 <nanosec>0</nanosec>
A10320             </period>
A10321         </deadline>
A10322         <durability>
A10323             <kind>VOLATILE_DURABILITY_QOS</kind>
A10324         </durability>
A10325         <history>
A10326             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10327             <depth>1</depth>
A10328         </history>
A10329         <liveliness>
A10330             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10331             <lease_duration>
A10332                 <sec>DURATION_INFINITE_SEC</sec>
A10333                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10334             </lease_duration>
A10335         </liveliness>
A10336         <ownership>
A10337             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10338         </ownership>
A10339         <reliability>
A10340             <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10341         </reliability>
A10342     </datareader_qos>
A10343     <datawriter_qos>
A10344         <deadline>
A10345             <period>
A10346                 <sec>2000</sec>
A10347                 <nanosec>0</nanosec>
A10348             </period>
A10349         </deadline>
A10350         <durability>
A10351             <kind>VOLATILE_DURABILITY_QOS</kind>
A10352         </durability>
A10353         <history>
A10354             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10355             <depth>1</depth>
A10356         </history>
A10357         <liveliness>
A10358             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10359             <lease_duration>
A10360                 <sec>DURATION_INFINITE_SEC</sec>
A10361                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10362             </lease_duration>
A10363         </liveliness>
A10364         <ownership>
A10365             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10366         </ownership>
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A10367         <ownership_strength>
A10368             <value>1</value>
A10369         </ownership_strength>
A10370         <reliability>
A10371             <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10372         </reliability>
A10373     </datawriter_qos>
A10374 </qos_profile>
A10375 <qos_profile name="Slow" base_name="tms_base">
A10376     <datareader_qos>
A10377         <deadline>
A10378             <period>
A10379                 <sec>20</sec>
A10380                 <nanosec>0</nanosec>
A10381             </period>
A10382         </deadline>
A10383         <durability>
A10384             <kind>VOLATILE_DURABILITY_QOS</kind>
A10385         </durability>
A10386         <history>
A10387             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10388             <depth>1</depth>
A10389         </history>
A10390         <liveliness>
A10391             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10392             <lease_duration>
A10393                 <sec>DURATION_INFINITE_SEC</sec>
A10394                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10395             </lease_duration>
A10396         </liveliness>
A10397         <ownership>
A10398             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10399         </ownership>
A10400         <reliability>
A10401             <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10402         </reliability>
A10403     </datareader_qos>
A10404     <datawriter_qos>
A10405         <deadline>
A10406             <period>
A10407                 <sec>20</sec>
A10408                 <nanosec>0</nanosec>
A10409             </period>
A10410         </deadline>
A10411         <durability>
A10412             <kind>VOLATILE_DURABILITY_QOS</kind>
A10413         </durability>
A10414         <history>
A10415             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10416             <depth>1</depth>
A10417         </history>
A10418         <liveliness>
A10419             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10420             <lease_duration>
A10421                 <sec>DURATION_INFINITE_SEC</sec>
A10422                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10423             </lease_duration>
A10424         </liveliness>
A10425         <ownership>
A10426             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10427         </ownership>
A10428         <ownership_strength>
A10429             <value>1</value>
A10430         </ownership_strength>
A10431         <reliability>
A10432             <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10433         </reliability>
A10434     </datawriter_qos>
A10435 </qos_profile>
A10436 <qos_profile name="Medium" base_name="tms_base">
A10437     <datareader_qos>
A10438         <deadline>
A10439             <period>
A10440                 <sec>3</sec>
A10441                 <nanosec>0</nanosec>
A10442             </period>
A10443         </deadline>
A10444         <durability>
A10445             <kind>VOLATILE_DURABILITY_QOS</kind>
A10446         </durability>
A10447         <history>
A10448             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10449             <depth>1</depth>
A10450         </history>
A10451         <liveliness>
A10452             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10453             <lease_duration>
A10454                 <sec>DURATION_INFINITE_SEC</sec>
A10455                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10456             </lease_duration>
A10457         </liveliness>
A10458         <ownership>
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A10459         <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10460     </ownership>
A10461     <reliability>
A10462         <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10463     </reliability>
A10464 </datareader_qos>
A10465 <datawriter_qos>
A10466     <deadline>
A10467         <period>
A10468             <sec>3</sec>
A10469             <nanosec>0</nanosec>
A10470         </period>
A10471     </deadline>
A10472     <durability>
A10473         <kind>VOLATILE_DURABILITY_QOS</kind>
A10474     </durability>
A10475     <history>
A10476         <kind>KEEP_LAST_HISTORY_QOS</kind>
A10477         <depth>1</depth>
A10478     </history>
A10479     <liveliness>
A10480         <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10481         <lease_duration>
A10482             <sec>DURATION_INFINITE_SEC</sec>
A10483             <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10484         </lease_duration>
A10485     </liveliness>
A10486     <ownership>
A10487         <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10488     </ownership>
A10489     <ownership_strength>
A10490         <value>1</value>
A10491     </ownership_strength>
A10492     <reliability>
A10493         <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10494     </reliability>
A10495 </datawriter_qos>
A10496 </qos_profile>
A10497 <qos_profile name="Continuous" base_name="tms_base">
A10498     <datareader_qos>
A10499         <deadline>
A10500             <period>
A10501                 <sec>2</sec>
A10502                 <nanosec>0</nanosec>
A10503             </period>
A10504         </deadline>
A10505         <durability>
A10506             <kind>VOLATILE_DURABILITY_QOS</kind>
A10507         </durability>
A10508         <history>
A10509             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10510             <depth>1</depth>
A10511         </history>
A10512         <liveliness>
A10513             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10514             <lease_duration>
A10515                 <sec>DURATION_INFINITE_SEC</sec>
A10516                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10517             </lease_duration>
A10518         </liveliness>
A10519         <ownership>
A10520             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10521         </ownership>
A10522         <reliability>
A10523             <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10524         </reliability>
A10525     </datareader_qos>
A10526     <datawriter_qos>
A10527         <deadline>
A10528             <period>
A10529                 <sec>2</sec>
A10530                 <nanosec>0</nanosec>
A10531             </period>
A10532         </deadline>
A10533         <durability>
A10534             <kind>VOLATILE_DURABILITY_QOS</kind>
A10535         </durability>
A10536         <history>
A10537             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10538             <depth>1</depth>
A10539         </history>
A10540         <liveliness>
A10541             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10542             <lease_duration>
A10543                 <sec>DURATION_INFINITE_SEC</sec>
A10544                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10545             </lease_duration>
A10546         </liveliness>
A10547         <ownership>
A10548             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10549         </ownership>
A10550         <ownership_strength>
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```
A10551         <value>1</value>
A10552     </ownership_strength>
A10553     <reliability>
A10554         <kind>BEST_EFFORT_RELIABILITY_QOS</kind>
A10555     </reliability>
A10556 </datawriter_qos>
A10557 </qos_profile>
A10558 <qos_profile name="Command" base_name="tms_base">
A10559     <datareader_qos>
A10560         <deadline>
A10561             <period>
A10562                 <sec>DURATION_INFINITE_SEC</sec>
A10563                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10564             </period>
A10565         </deadline>
A10566         <durability>
A10567             <kind>VOLATILE_DURABILITY_QOS</kind>
A10568         </durability>
A10569         <history>
A10570             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10571             <depth>1</depth>
A10572         </history>
A10573         <liveliness>
A10574             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10575             <lease_duration>
A10576                 <sec>DURATION_INFINITE_SEC</sec>
A10577                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10578             </lease_duration>
A10579         </liveliness>
A10580         <ownership>
A10581             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10582         </ownership>
A10583         <reliability>
A10584             <kind>RELIABLE_RELIABILITY_QOS</kind>
A10585         </reliability>
A10586     </datareader_qos>
A10587     <datawriter_qos>
A10588         <deadline>
A10589             <period>
A10590                 <sec>DURATION_INFINITE_SEC</sec>
A10591                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10592             </period>
A10593         </deadline>
A10594         <durability>
A10595             <kind>VOLATILE_DURABILITY_QOS</kind>
A10596         </durability>
A10597         <history>
A10598             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10599             <depth>1</depth>
A10600         </history>
A10601         <liveliness>
A10602             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10603             <lease_duration>
A10604                 <sec>DURATION_INFINITE_SEC</sec>
A10605                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10606             </lease_duration>
A10607         </liveliness>
A10608         <ownership>
A10609             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10610         </ownership>
A10611         <ownership_strength>
A10612             <value>1</value>
A10613         </ownership_strength>
A10614         <reliability>
A10615             <kind>RELIABLE_RELIABILITY_QOS</kind>
A10616         </reliability>
A10617     </datawriter_qos>
A10618 </qos_profile>
A10619 <qos_profile name="Response" base_name="tms_base">
A10620     <datareader_qos>
A10621         <deadline>
A10622             <period>
A10623                 <sec>DURATION_INFINITE_SEC</sec>
A10624                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10625             </period>
A10626         </deadline>
A10627         <durability>
A10628             <kind>VOLATILE_DURABILITY_QOS</kind>
A10629         </durability>
A10630         <history>
A10631             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10632             <depth>1</depth>
A10633         </history>
A10634         <liveliness>
A10635             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10636             <lease_duration>
A10637                 <sec>DURATION_INFINITE_SEC</sec>
A10638                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10639             </lease_duration>
A10640         </liveliness>
A10641         <ownership>
A10642             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
```

*TMS Data Model*  
*DRAFT 19 Apr 2021*

```

A10643         </ownership>
A10644         <reliability>
A10645             <kind>RELIABLE_RELIABILITY_QOS</kind>
A10646         </reliability>
A10647     </datareader_qos>
A10648     <datawriter_qos>
A10649         <deadline>
A10650             <period>
A10651                 <sec>DURATION_INFINITE_SEC</sec>
A10652                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10653             </period>
A10654         </deadline>
A10655         <durability>
A10656             <kind>VOLATILE_DURABILITY_QOS</kind>
A10657         </durability>
A10658         <history>
A10659             <kind>KEEP_LAST_HISTORY_QOS</kind>
A10660             <depth>1</depth>
A10661         </history>
A10662         <liveliness>
A10663             <kind>AUTOMATIC_LIVELINESS_QOS</kind>
A10664             <lease_duration>
A10665                 <sec>DURATION_INFINITE_SEC</sec>
A10666                 <nanosec>DURATION_INFINITE_NSEC</nanosec>
A10667             </lease_duration>
A10668         </liveliness>
A10669         <ownership>
A10670             <kind>EXCLUSIVE_OWNERSHIP_QOS</kind>
A10671         </ownership>
A10672         <ownership_strength>
A10673             <value>1</value>
A10674         </ownership_strength>
A10675         <reliability>
A10676             <kind>RELIABLE_RELIABILITY_QOS</kind>
A10677         </reliability>
A10678     </datawriter_qos>
A10679 </qos_profile>
A10680 </qos_library>
A10681 </dds>

```