IEEE SA
STANDARDS ASSOCIATION

IEEE Standard for Smart Energy Profile Application Protocol

STANDARDS

IEEE Communications Society

Power Line Communications Committee

IEEE Std 2030.5™-2023
(Revision of IEEE Std 2030.5-2018/Incorporates IEEE Std 2030.5-2023/Cor1-2024)

IEEE
AVA Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

IEEE Std 2030.5™-2023
(Revision of IEEE Std 2030.5-2018/Incorporates IEEE Std 2030.5-2023/Cor1-2024)

IEEE Standard for Smart Energy Profile Application Protocol

Developed by the

Power Line Communications Committee
of the
IEEE Communications Society

Approved 6 December 2023

IEEE SA Standards Board

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 2030.5-2023

Abstract: The application layer with TCP/IP providing functions in the transport and Internet layers to enable utility management of the end user energy environment, including demand response, load control, time of day pricing, management of distributed generation, electric vehicles, etc., is defined in this standard. Depending on the physical layer in use (e.g., IEEE 802.15.4™, IEEE 802.11™, IEEE 1901™, IEEE 1901.2™), a variety of lower layer protocols may be involved in providing a complete solution. Generally, lower layer protocols are not discussed in this standard except where there is direct interaction with the application protocol. The mechanisms for exchanging application messages, the exact messages exchanged, including error messages, and the security features used to protect the application messages are defined in this standard. With respect to the Open Systems Interconnection (OSI) network model, this standard is built using the four-layer Internet stack model. Elements from many existing standards, including IEC 61968 and IEC 61850, are sourced and a RESTful architecture (Fielding [B2]) using IETF protocols such as HTTP is followed by the defined application profile.

Keywords: adoption, application, application protocol, demand response, distributed energy resources, energy usage information, IEEE 2030.5™, load control, metering, plugin electric vehicles, prepayment, pricing communication, RESTful, SEP 2, smart energy, smart energy profile, Smart Energy Profile 2

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2024 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 6 December 2024. Printed in the United States of America.

IEEE is a registered trademark in the U.S. Patent & Trademark Office, owned by The Institute of Electrical and Electronics Engineers, Incorporated.

PDF: ISBN 979-8-8557-0599-7 STD26840
Print: ISBN 979-8-8557-0600-0 STDPD26840

IEEE prohibits discrimination, harassment, and bullying.
For more information, visit https://www.ieee.org/about/corporate/governance/p9-26.html.
No part of this publication may be reproduced in any form, in an electronic retrieval system or otherwise, without the prior written permission of the publisher.

Copyright © 2024 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 2030.5-2023

Important Notices and Disclaimers Concerning IEEE Standards Documents

IEEE Standards documents are made available for use subject to important notices and legal disclaimers. These notices and disclaimers, or a reference to this page (https://standards.ieee.org/ipr/disclaimers.html), appear in all IEEE standards and may be found under the heading "Important Notices and Disclaimers Concerning IEEE Standards Documents."

Notice and Disclaimer of Liability Concerning the Use of IEEE Standards Documents

IEEE Standards documents are developed within IEEE Societies and subcommittees of IEEE Standards Association (IEEE SA) Board of Governors. IEEE develops its standards through an accredited consensus development process, which brings together volunteers representing varied viewpoints and interests to achieve the final product. IEEE standards are documents developed by volunteers with scientific, academic, and industry-based expertise in technical working groups. Volunteers involved in technical working groups are not necessarily members of IEEE or IEEE SA and participate without compensation from IEEE. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

IEEE makes no warranties or representations concerning its standards, and expressly disclaims all warranties, express or implied, concerning all standards, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement IEEE Standards documents do not guarantee safety, security, health, or environmental protection, or compliance with law, or guarantee against interference with or from other devices or networks. In addition, IEEE does not warrant or represent that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied "AS IS" and "WITH ALL FAULTS."

Use of an IEEE standard is wholly voluntary. The existence of an IEEE standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity, nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document should rely upon their own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

IN NO EVENT SHALL IEEE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO: THE NEED TO PROCURE SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE PUBLICATION, USE OF, OR RELIANCE UPON ANY STANDARD, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE AND REGARDLESS OF WHETHER SUCH DAMAGE WAS FORESEEABLE.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Translations

The IEEE consensus balloting process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English language version published by IEEE is the approved IEEE standard.

Use by artificial intelligence systems

In no event shall material in any IEEE Standards documents be used for the purpose of creating, training, enhancing, developing, maintaining, or contributing to any artificial intelligence systems without the express, written consent of IEEE SA in advance. "Artificial intelligence" refers to any software, application, or other system that uses artificial intelligence, machine learning, or similar technologies, to analyze, train, process, or generate content. Requests for consent can be submitted using the Contact Us form.

Official statements

A statement, written or oral, that is not processed in accordance with the IEEE SA Standards Board Operations Manual is not, and shall not be considered or inferred to be, the official position of IEEE or any of its committees and shall not be considered to be, or be relied upon as, a formal position of IEEE or IEEE SA. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that the presenter's views should be considered the personal views of that individual rather than the formal position of IEEE, IEEE SA, the Standards Committee, or the Working Group. Statements made by volunteers may not represent the formal position of their employer(s) or affiliation(s). News releases about IEEE standards issued by entities other than IEEE SA should be considered the view of the entity issuing the release rather than the formal position of IEEE or IEEE SA.

Comments on standards

Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE or IEEE SA. However, **IEEE does not provide interpretations, consulting information, or advice pertaining to IEEE Standards documents**.

Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its Societies and subcommittees of the IEEE SA Board of Governors are not able to provide an instant response to comments or questions, except in those cases where the matter has previously been addressed. For the same reason, IEEE does not respond to interpretation requests. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE SA working group. You can indicate interest in a working group using the Interests tab in the Manage Profile & Interests area of the IEEE SA myProject system.[1] An IEEE Account is needed to access the application.

Comments on standards should be submitted using the Contact Us form.[2]

[1] Available at: https://development.standards.ieee.org/myproject-web/public/view.html#landing.
[2] Available at: https://standards.ieee.org/about/contact/.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not constitute compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Data privacy

Users of IEEE Standards documents should evaluate the standards for considerations of data privacy and data ownership in the context of assessing and using the standards in compliance with applicable laws and regulations.

Copyrights

IEEE draft and approved standards are copyrighted by IEEE under U.S. and international copyright laws. They are made available by IEEE and are adopted for a wide variety of both public and private uses. These include both use by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making these documents available for use and adoption by public authorities and private users, neither IEEE nor its licensors waive any rights in copyright to the documents.

Photocopies

Subject to payment of the appropriate licensing fees, IEEE will grant users a limited, non-exclusive license to photocopy portions of any individual standard for company or organizational internal use or individual, non-commercial use only. To arrange for payment of licensing fees, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400; https://www.copyright.com/. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Updating of IEEE Standards documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect.

Every IEEE standard is subjected to review at least every 10 years. When a document is more than 10 years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit IEEE Xplore or contact IEEE.[3] For more information about the IEEE SA or IEEE's standards development process, visit the IEEE SA Website.

Errata

Errata, if any, for all IEEE standards can be accessed on the IEEE SA Website.[4] Search for standard number and year of approval to access the web page of the published standard. Errata links are located under the Additional Resources Details section. Errata are also available in IEEE Xplore. Users are encouraged to periodically check for errata.

Patents

IEEE standards are developed in compliance with the IEEE SA Patent Policy.[5]

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE SA Website at https://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

IMPORTANT NOTICE

Technologies, application of technologies, and recommended procedures in various industries evolve over time. The IEEE standards development process allows participants to review developments in industries, technologies, and practices, and to determine what, if any, updates should be made to the IEEE standard. During this evolution, the technologies and recommendations in IEEE standards may be implemented in ways not foreseen during the standard's development. IEEE standards development activities consider research and information presented to the standards development group in developing any safety recommendations. Other information about safety practices, changes in technology or technology implementation, or impact by peripheral systems also may be pertinent to safety considerations during implementation of the standard. Implementers and users of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, data privacy, and interference protection practices and all applicable laws and regulations.

[3] Available at: https://ieeexplore.ieee.org/browse/standards/collection/ieee.
[4] Available at: https://standards.ieee.org/standard/index.html.
[5] Available at: https://standards.ieee.org/about/sasb/patcom/materials.html.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Participants

At the time this standard was completed, the Smart Energy Profile 2.0 Working Group had the following membership:

Robby Simpson, *Chair*
Gordon Lum, *Vice Chair*

Organization Represented	*Name of Representative*
ARM	Tom Herbst
DER Security Corp	Tom Tansy
Electric Power Research Institute, Inc.	Ajit Renjit
Energy Toolbase	James Carlson
Enetrics	Robby Simpson
General Electric	Jesse Gantz
Hydro Quebec	Amin Ketari
Itron, Inc.	Don Sturek
Kalkitech	Prasanth Gopalakrishnan
KITU Systems	Gordon Lum
Lawrence Berkeley National Laboratory	Bruce Nordman
QualityLogic	Steve Kang
Schneider Electric	Navdeep Ahuja
Southern California Edison	Warren Abatay
SunSpec Alliance	Kudrat Kaur
Xylem	Michael Cowan

The Working Group gratefully acknowledges the contributions of the following participants. Without their assistance and dedication, this standard would not have been completed.

Jithender Anandan
Matthew Brigdan
Sheldon Crow
Ed Eckert
Gene Falendysz
Prajwal Gautam
Nasif Imtiaz
James Mater
Josh McDonald
Stephen McGregor
Amir Miragha
Jorge Pineda
Sanjay Prasad
Guruprasad Ramani
Brian Seal
Jim Zuber

Copyright © 2024 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 2030.5-2023

The following members of the entity Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

DER Security Corp
Electric Power Research Institute, Inc. (EPRI)
Institute of Biomedical Engineering
Intel
Itron Inc.
Kalkitech Inc.
KITU Systems
Natural Resources Canada (NRCan)
Power Plus Communications AG
QualityLogic, Inc.
SAE Industry Technologies Consortia (SAE ITC)
Schneider Electric
Siemens Corporation
South China University of Technology
Southern California Edison
State Grid Corporation of China (SGCC)
SunSpec Alliance
WeBank Co., Ltd.
Wi-SUN Alliance

When the IEEE SA Standards Board approved this standard on 6 December 2023, it had the following membership:

David J. Law, *Chair*
Ted Burse, *Vice Chair*
Gary Hoffman, *Past Chair*
Konstantinos Karachalios, *Secretary*

Sara R. Biyabani
Doug Edward
Ramy Ahmed Fathy
Guido R. Hiertz
Yousef Kimiagar
Joseph L. Koepfinger*
Thomas Koshy
John D. Kulick
Joseph S. Levy
Howard Li
Johnny Daozhuang Lin
Gui Lin
Xiaohui Liu
Kevin W. Lu
Daleep C. Mohla
Andrew Myles
Paul Nikolich
Annette D. Reilly
Robby Robson
Lei Wang
F. Keith Waters
Karl Weber
Philip B. Winston
Don Wright

*Member Emeritus

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Participants

At the time this standard was completed, the Smart Energy Profile 2.0 Working Group had the following membership:

Robby Simpson, *Chair*
Gordon Lum, *Vice Chair*

Organization Represented	*Name of Representative*
DER Security Corp	Tom Tansy
Electric Power Research Institute, Inc.	Ajit Renjit
Enetrics	Robby Simpson
General Electric	Jesse Gantz
Hydro Quebec	Amin Ketari
Kalkitech	Prasanth Gopalakrishnan
KITU Systems	Gordon Lum
Lawrence Berkeley National Laboratory	Bruce Nordman
QualityLogic	Steve Kang
Schneider Electric	Navdeep Ahuja
Southern California Edison	Warren Abatay
SunSpec Alliance	Kudrat Kaur

The Working Group gratefully acknowledges the contributions of the following participants. Without their assistance and dedication, this standard would not have been completed.

Fatima Amara
Bob Fox
Nasif Imtiaz
James Mater
Amir Miragha
Sanjay Prasad

The following members of the entity Standards Association balloting group voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Eaton
Electric Power Research Institute, Inc. (EPRI)
Enetrics LLC
Kalkitech Inc.
KITU Systems
Power Plus Communications AG
QualityLogic, Inc.
Schneider Electric
Sensus
Siemens AG
Southern California Edison
SunSpec Alliance

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

When the IEEE SA Standards Board approved this standard on 26 September 2024, it had the following membership:

David J. Law, *Chair*
Jon Walter Rosdahl, *Vice Chair*
Gary Hoffman, *Past Chair*
Alpesh Shah, *Secretary*

Sara R. Biyabani
Ted Burse
Stephen Dukes
Doug Edwards
J. Travis Griffith
Guido R. Hiertz
Ronald W. Hotchkiss
Hao Hu
Yousef Kimiagar
Joseph L. Koepfinger*
Howard Li
Xiaohui Liu
John Haiying Lu
Kevin W. Lu
Hiroshi Mano
Paul Nikolich
Robby Robson
Lei Wang
F. Keith Waters
Sha Wei
Philip B. Winston
Don Wright

*Member Emeritus

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Introduction

This introduction is not part of IEEE Std 2030.5-2023, IEEE Standard for Smart Energy Profile Application Protocol.

The empowerment of consumers to manage their usage and generation of energy is a critical feature of the Smart Grid and is a basis of innovation for new products and services in energy management. To enable this capability, information flow between devices such as meters, smart appliances, plug-in electric vehicles, energy management systems, and distributed energy resources (including renewable energy and storage elements) must occur in an open, standardized, secure, and interoperable fashion. The following standard is intended to fulfill those needs.

The first publication of this standard (IEEE Std 2030.5-2013) was driven by, and sought to address the requirements of, many activities across the globe. Of note were the efforts within the United States by the National Institute of Standards and Technology (NIST) and the Smart Grid Interoperability Panel (SGIP) (in particular, Priority Action Plans 3, 9, 10, 11, and 18, with influence from many of the others) in fulfillment of the EISA 2007 legislation, the European Mandate on Smart Metering (M/441) (in particular, efforts within CEN/CENELEC and ETSI, and the Smart Meter Working Group), as well as similar efforts in Australia, the United Kingdom, Japan, and China, and electric vehicle standardization efforts (in particular, ISO/IEC JWG automotive EV standards and SAE EV standards), to name only a few.

This revision of IEEE Std 2030.5 was made with particular attention to the activities underway in California as part of the Rule 21 revision and the associated Smart Inverter Working Group, as well as the revision of IEEE Std 1547™. This revision also sought to address any errors and ambiguities discovered in the testing and deployment of the first publication.

After approval on 6 December 2023, technical errors were identified by the Smart Energy Profile 2.0 Working Group. A corrigendum was then submitted (IEEE Std 2030.5-2023/Cor1-2024) to address these errors and approved on 26 September 2024. All changes outlined in the corrigendum have since been made to IEEE Std 2030.5-2023.

This standard is also intended to enable communications that are link-layer agnostic and run over the Internet Protocol. Careful consideration was given to premises networks with various architectures, numbers of devices, and constraints, while maintaining flexibility, extensibility, and security.

IEEE Standards downloads and executable files

Supplemental files are available in the IEEE Std 2030.5-2023 directory located at http://standards.ieee.org/downloads/.

Acknowledgements

Dedicated to the memory of our friend, colleague, and former Chair Bob Heile.

Copyright © 2024 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 2030.5-2023

Contents

1. Overview 16
   1.1 Scope 16
   1.2 Purpose 16
   1.3 Word usage 17
   1.4 Document organization 17
   1.5 Typography conventions used 17
   1.6 Design principles 17

2. Normative references 18

3. Definitions, acronyms, and abbreviations 20
   3.1 Definitions 20
   3.2 Acronyms and abbreviations 22

4. Design pattern 23
   4.1 Protocol flexibility 23
   4.2 General rules/best practices 23
   4.3 WADL 24
   4.4 Schema 25
   4.5 Uniform resource identifiers 25
   4.6 List resources 25
   4.7 Resource design rules 29
   4.8 Resource identification 30
   4.9 Backward and forward compatibility 30

5. Application support 31
   5.1 Overview 31
   5.2 Use of TCP 32
   5.3 URI encoding 32
   5.4 HTTP headers 32
   5.5 HTTP response codes 34
   5.6 Application payload syntax 38
   5.7 Content negotiation 39

6. Security 40
   6.1 Introduction 40
   6.2 Security attributes 40
   6.3 Device credentials 47
   6.4 Resource access authentication and authorization context 49
   6.5 Resource access authentication 50
   6.6 Resource access authorization 50
   6.7 Cipher suites 50
   6.8 Default security policy 51
   6.9 Registration 52
   6.10 Security LogEvents 54
   6.11 Certificate management 54

7. Discovery 65
   7.1 Introduction 65

Copyright © 2024 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 2030.5-2023

7.2 Service instance 66
7.3 Service name 67
7.4 TXT record 67
7.5 Subtype queries 68
7.6 Discovery procedure 70

8. Support resources 70
8.1 Introduction 70
8.2 Resource section outlines 71
8.3 Device Capabilities function set 72
8.4 Self Device function set 73
8.5 End Device function set 73
8.6 Proxied Device function set 75
8.7 Aggregated Device function set 77
8.8 Function Set Assignments function set 79
8.9 Subscription/Notification function set 80
8.10 Response function set 83

9. Common resources 87
9.1 Introduction 87
9.2 Time function set 88
9.3 Device Information function set 89
9.4 Power Status function set 90
9.5 Network Status function set 91
9.6 Log Event function set 92
9.7 Configuration function set 93
9.8 File Download function set 94

10. Smart energy resources 99
10.1 Introduction 99
10.2 Common application functionality 99
10.3 Demand Response and Load Control function set 106
10.4 Metering function set 110
10.5 Pricing function set 116
10.6 Messaging function set 120
10.7 Billing function set 121
10.8 Prepayment function set 124
10.9 Flow Reservation function set 126
10.10 Distributed Energy Resources function set 127
10.11 Metering Mirror function set 135

11. Manufacturer specific proprietary extentions 138
11.1 Overview 138
11.2 mDNS/DNS-SD and xmDNS/DNS-SD 138
11.3 URIs 138
11.4 Resources 138
11.5 DeviceCapability resource 139

Annex A (informative) Web-application description language (WADL) 140
A.1 Introduction 140
A.2 Support resources section 140
A.3 Common resources section 144
A.4 Smart Energy resources section 148

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Annex B (informative) IEEE 2030.5 model 162
B.1 Introduction 162
B.2 IEEE 2030.5 package 162
B.3 DeviceCapability package 163
B.4 Common package 163
B.5 EndDevice package 185
B.6 FunctionSetAssignments package 189
B.7 Pub-Sub package 190
B.8 Response package 192
B.9 Time package 194
B.10 DeviceInformation package 195
B.11 PowerStatus package 199
B.12 NetworkStatus package 201
B.13 LogEvents package 206
B.14 Configuration package 209
B.15 SoftwareDownload package 211
B.16 DRLC package 213
B.17 Metering package 218
B.18 Pricing package 224
B.19 Messaging package 229
B.20 Billing package 230
B.21 Prepayment package 234
B.22 FlowReservation package 239
B.23 DER package 242
B.24 AggregatedDevice package 279
B.25 ProxiedDevice package 281
B.26 Links package 281

Annex C (informative) Examples and guideline 288
C.1 Introduction 288
C.2 Registration: Remote 288
C.3 Registration: Remote with AggregatedDeviceList 291
C.4 Registration: Remote with ProxiedDeviceList 293
C.5 Registration: Local 296
C.6 Discovery: Function Set Assignment 299
C.7 Discovery: Without Function Set Assignment 301
C.8 Discover: Undirected without Function Set Assignment 303
C.9 Subscription/Notification 304
C.10 Demand response: General 307
C.11 Demand response: Cancel 311
C.12 Distributed energy resource: General 313
C.13 Metering: Reading 317
C.14 Metering: Interval 322
C.15 Metering: Instantaneous 328
C.16 Metering: Mirroring 331
C.17 Pricing: Time of use 343
C.18 Billing: Billing period 348
C.19 Billing: Historical 351
C.20 Billing: Projection 354
C.21 File loading 357
C.22 Flow Reservation: General 361
C.23 Flow Reservation: Cancel 366
C.24 Event randomization 370
C.25 Event timing scenarios 374

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Annex D (informative) Guidelines 386
D.1 Pricing implementation guidelines 386
D.2 PEV implementation guidelines (subject to work with SAE and ISO/IEC) 391

Annex E (informative) Mapping to IEEE Std 1547-2018 392

Annex F (informative) Bibliography 396

Copyright © 2024 IEEE. All rights reserved.

Documents
Document Sharing Hub

Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.

Full document details: IEEE 2030.5-2023

IEEE Standard for Smart Energy Profile Application Protocol

1. Overview

1.1 Scope

This standard defines an application profile that provides an interface between the smart grid and users. It enables management of the end user energy environment, including demand response, load control, price communication, distributed generation, energy storage, and electric vehicles, as well as the support of additional commodities, including water, natural gas, and steam. This standard defines the mechanisms for exchanging application messages, the exact messages exchanged including error messages, and the security features used to protect the application messages. This standard focuses on a variety of possible architectures and usage models, including direct communications between a service provider and consumers/prosumers, communications within a premises network, and communications between a service provider and an aggregator. Lastly, this standard sources elements from many existing standards, including IEC 61968[6] and IEC 61850, and follows a RESTful architecture utilizing widely adopted protocols such as TCP/IP and HTTP. In addition, it supports all of the needs of IEEE Std 1547™-2018. This revision maintains backward compatibility with IEEE Std 2030.5™-2018, except for elimination of the requirements for mandatory DERControl modes, while providing an expanded feature set.

1.2 Purpose

The purpose of this standard is the definition of an application profile interfacing the smart grid and users. This definition is critical in enabling the management of distributed generation; electric vehicles; energy storage; the end user energy environment, including demand response, load control, and price communication; as well as the support of additional commodities, including water, natural gas, and steam. Maintaining grid stability, which is becoming a major problem in many areas, and coping with the anticipated rising cost of energy are just two of several problems requiring this kind of management capability. In an effort to facilitate global utilization, this standard sources elements from many existing standards, including IEC 61968 and IEC 61850, and follows a RESTful architecture utilizing widely adopted protocols such as TCP/IP and HTTP. It also supports all of the needs of IEEE Std 1547™-2018.

[6] Information on references can be found in Clause 2.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

1.3 Word usage

The word *shall* indicates mandatory requirements strictly to be followed in order to conform to the standard and from which no deviation is permitted (*shall* equals *is required to*).[7, 8]

The word *should* indicates that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required (*should* equals *is recommended that*).

The word *may* is used to indicate a course of action permissible within the limits of the standard (*may* equals *is permitted to*).

The word *can* is used for statements of possibility and capability, whether material, physical, or causal (*can* equals *is able to*).

1.4 Document organization

The following documents comprise the definition of IEEE Std 2030.5, and all IEEE 2030.5 devices will be required to maintain compliance to the following documents:

- IEEE Std 2030.5 (this document)
- IEEE 2030.5 XML Schema Definition (XSD) (sep.xsd in the supplemental material of IEEE Std 2030.5[9])
- IEEE 2030.5 WADL (sep_wadl.xml in the supplemental material of IEEE Std 2030.5)

The IEEE 2030.5 XSD contains the definitions of the IEEE 2030.5 resources, attributes, and elements, as well as their textual descriptions. The IEEE 2030.5 WADL contains the recommended URI structures and use of HTTP methods associated with these objects. In addition, an IEEE 2030.5 UML model (also contained in the supplemental material of IEEE Std 2030.5) has been utilized in the creation of the IEEE 2030.5 XSD. The IEEE 2030.5 UML model is an informative document. Informative textual extracts of the IEEE 2030.5 XSD and the IEEE 2030.5 WADL are contained in this document for convenience.

1.5 Typography conventions used

Example URIs, protocol requests, protocol responses, and XML are presented in `fixed-width Courier New font, with a size of 8.`

1.6 Design principles

As IEEE Std 2030.5 follows a RESTful architecture (Fielding [B2][10]), several design principles are important for the standard, including:

- While devices MAY maintain state, interfaces SHOULD be stateless.
- URI structure SHOULD be clear but as efficient as possible.
- Minimize the number of transactions required to achieve a given function.

[7] The use of the word must is deprecated and cannot be used when stating mandatory requirements; must is used only to describe unavoidable situations.
[8] The use of will is deprecated and cannot be used when stating mandatory requirements; will is only used in statements of fact.
[9] Supplemental files are available in the IEEE Std 2030.5-2023 directory located at http://standards.ieee.org/downloads/.
[10] The numbers in brackets correspond to those of the bibliography in Annex F.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

2. Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEC 61850, Communication networks and systems in substations—All parts.[11]

IEC 61968, Application integration at electric utilities—System interfaces for distribution management—All parts.

IEEE Std 802.1AR™, IEEE Standard for Local and Metropolitan Area Networks—Secure Device Identity.[12, 13]

IEEE Std 1547™-2018, IEEE Standard for Interconnection and Interoperability of Distributed Energy Resources with Associated Electric Power Systems Interfaces.

IETF, Extended Multicast DNS (https://tools.ietf.org/html/draft-lynn-homenet-site-mdns-01).

IETF RFC 793, Transmission Control Protocol (http://tools.ietf.org/html/rfc793).[14]

IETF RFC 1630, Universal Resource Identifiers in WWW (http://tools.ietf.org/html/rfc1630).

IETF RFC 2818, HTTP Over TLS (http://tools.ietf.org/html/rfc2818).

IETF RFC 2863, The Interface MIB (https://tools.ietf.org/html/rfc2863).

IETF RFC 3986, Uniform Resource Identifier (URI): Generic Syntax (https://tools.ietf.org/html/rfc3986).

IETF RFC 4108, Using Cryptographic Message Syntax (CMS) to Protect Firmware (http://tools.ietf.org/html/rfc4108).

IETF RFC 4193, Unique IPv6 Unicast Addresses (http://tools.ietf.org/html/rfc4193).

IETF RFC 4291, IP Version 6 Addressing Architecture (http://tools.ietf.org/html/rfc4291).

IETF RFC 5246, Transport Layer Security (TLS) Protocol Version 1.2 (http://tools.ietf.org/html/rfc5246).

IETF RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (http://tools.ietf.org/html/rfc5280).

IETF RFC 5480, Elliptic Curve Cryptography Subject Public Key Information (http://tools.ietf.org/html/rfc5480).

IETF RFC 5646, Tags for Identifying Languages (https://datatracker.ietf.org/doc/html/rfc5646).

IETF RFC 6550, RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. (http://tools.ietf.org/html/rfc6550).

[11] IEC publications are available from the International Electrotechnical Commission (https://www.iec.ch) and the American National Standards Institute (https://www.ansi.org/).
[12] The IEEE standards or products referred to in Clause 2 are trademarks owned by The Institute of Electrical and Electronics Engineers, Incorporated.
[13] IEEE publications are available from The Institute of Electrical and Electronics Engineers (https://standards.ieee.org/).
[14] RFC documents are available through the IETF at https://www.ietf.org/standards/rfcs/.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

IETF RFC 6554, An IPv6 Routing Header for Source Routes with the Routing Protocol for Low-Power and Lossy Networks (RPL) (http://tools.ietf.org/html/rfc6554).

IETF RFC 6585, Additional HTTP Status Codes (https://www.rfc-editor.org/rfc/rfc6585.html).

IETF RFC 6762, Multicast DNS (http://tools.ietf.org/html/rfc6762).

IETF RFC 6763, DNS-Based Service Discovery (http://tools.ietf.org/html/rfc6763).

IETF RFC 6960, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—OCSP (https://tools.ietf.org/html/rfc6960).

IETF RFC 7251, AES-CCM Elliptic Curve Cryptography (ECC) Cipher Suites for TLS (http://tools.ietf.org/html/rfc7251).

IETF RFC 9110, HTTP Semantics (https://www.rfc-editor.org/rfc/rfc9110).

IETF RFC 9111, HTTP Caching (https://www.rfc-editor.org/rfc/rfc9111).

IETF RFC 9112, HTTP/1.1 (https://www.rfc-editor.org/rfc/rfc9112).

Internet Assigned Numbers Authority (IANA), Private Enterprise Number (PEN) Request.[15]

ISO 3166-1, Codes for the representation of names of countries and their subdivisions—Part 1: Country code.[16]

ISO 3166-2, Codes for the representation of names of countries and their subdivisions—Part 2: Country subdivision code.

ISO 4217, Codes for the representation of currencies.

ITU-T Recommendation X.680 (08/2015), Information technology—Abstract Syntax Notation One (ASN.1): Specification of basic notation.[17]

NIST Special Publication (SP) 800-57, Recommendation for Key Management.[18]

NIST Special Publication (SP) 800-131A, Transition: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths.

W3C, Efficient XML Interchange (EXI) Format 1.0 (http://www.w3.org/TR/exi/).[19]

W3C, Extensible Markup Language (XML) (http://www.w3.org/TR/REC-xml/).

W3C, Web Application Description Language (WADL) (http://www.w3.org/Submission/wadl/).

[15] Available online: http://pen.iana.org/pen/PenApplication.page.
[16] ISO publications are available from the International Organization for Standardization (https://www.iso.org/) and the American National Standards Institute (https://www.ansi.org/).
[17] ITU-T publications are available from the International Telecommunications Union (https://www.itu.int/).
[18] NIST publications are available from the National Institute of Standards and Technology (https://www.nist.gov/).
[19] W3C publications are available from the World Wide Web Consortium (https://www.w3.org/).

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

3. Definitions, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause. [20]

access control list: A security mechanism in which entities and authorizations (e.g., read, write, create, delete) are related to resources to determine the entities' allowed operations on the resources.

backward compatibility: The ability for data (e.g., XML instances) created according to a newer revision of IEEE Std 2030.5 to be successfully read and processed by a device compliant with an older revision of IEEE Std 2030.5.

certificate authority: An entity that issues digital certificates for use by other entities.

certificate chain: A chain of certificates, with each certificate's signature verified using the key from the next certificate in the chain. The single exception is the certificate at the end of the chain (the trust anchor), known as the root certificate authority (CA) certificate, which is self-signed.

client: The device or host that interacts with a server to obtain information related to a resource hosted by the server.

device certificate: A digital certificate installed within a device that binds the device identity to the device. Device certificates are exchanged by network access control and application protocols to authenticate devices as genuine IEEE Std 2030.5 and further to prove specific device identity.

energy services interface (ESI): A device, with multiple network interfaces, which is a member of both the home smart energy network and a service provider's private network. This is the primary mechanism for the service provider to contribute data and directives into the smart energy network and to receive responses from smart energy devices.

fingerprint: This is the result of summarizing a certificate with a secure hash function. The fingerprint is generally expressed as a hex string. It is used to confirm the integrity of a certificate obtained over an untrusted channel.

forward compatibility: The ability for data (e.g., XML instances) created according to an older revision of IEEE Std 2030.5 to be successfully read and processed by a device compliant with a newer revision of IEEE Std 2030.5.

function set: A logical grouping of resources that cooperate to implement IEEE 2030.5 features (e.g., metering, demand response, and load control).

Function Set Assignments: A logical addressing mechanism in IEEE Std 2030.5 that allows devices to be directed to use specific resources (e.g., to facilitate a device's participation in a program). Please see 8.8 for details.

function set instance: A single, top-level, instance of a function set (e.g., a single UsagePoint for the Metering function set, a single DemandResponseProgram for the Demand Response and Load Control function set).

[20]IEEE Standards Dictionary Online is available at: http://dictionary.ieee.org. An IEEE Account is required for access to the dictionary, and one can be created at no charge on the dictionary sign-in page.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

host: This is the representation of a device in its application context. Typically represented by an Internet Protocol (IP) address or domain name.

intermediate CA: A certificate authority (CA) below the root CA that issues certificates to subordinate CAs.

issuing CA: A certificate authority (CA) that issues certificates to devices or code-signers.

manufacturer's CA (MCA): An intermediate certificate authority (CA) operated by a specific manufacturer for the purpose of issuing manufacturing issuing CAs for that manufacturer.

manufacturing issuing CA (MICA): An issuing certificate authority (CA) that issues certificates to devices during the manufacturing process.

manufacturing public key infastructure (PKI): The set of certificate authorities (CAs) that issue certificates to devices during the manufacturing process. The set includes the Smart Energy Root CA, Manufacturer's CAs, and Manufacturing Issuing CAs.

master resource identifier (mRID): An mRID is the global identifier attribute of an object. For further definition and formatting, see the schema (IEEE 2030.5 supplemental material).

node: This is the representation of a device in its network context, typically represented as an Internet Protocol (IP) address.

object identifier (OID): An OID consists of a node in a hierarchically-assigned namespace, formally defined using ITU-T's ASN.1 standard (ITU-T Recommendation X.680).

privacy-enhanced electronic mail (PEM) format certificate: An X.509 certificate that has been Base64 encoded and wrapped in "-----BEGIN CERTIFICATE-----", "-----END CERTIFICATE-----" sentinels for transport as a text file or block.

public key infrastructure (PKI): A set of hardware, software, people, policies, and procedures needed to create, manage, distribute, use, store, and revoke digital certificates.

registered: The state of a device with regards to a particular server wherein an EndDevice record for the device has been populated on the server and that record contains a valid registration record. These records are typically populated with device information transmitted out-of-band to the server's owner.

resource: Uniform resource identifier (URI) addressable object that is manipulated via the RESTful uniform interface.

resource discovery: The process whereby clients identify resources being served on the network. Clients issue a request to all devices on the network requesting resource(s) of interest. Servers hosting the requested resource(s) respond with information necessary to access the server and its resource(s).

root CA: A certificate authority (CA) whose certificate or public key is a trust anchor for any other certificates in a chain of trust.

root certificate: The root certificate authority's self-signed certificate. Generally also a trust anchor.

self-signed certificate: A certificate whose issuer and subject are identical, and whose public key verifies its signature.

server: The device or host that holds a resource and exposes representations of that resource.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Smart Energy Root CA (SERCA): The top-level certificate authority (CA) for the IEEE 2030.5 manufacturing public key infastructure (PKI).

subordinate resource: A direct child resource of another resource.

trust anchor: The root of trust for a certificate chain. This is an authoritative entity represented by a public key and associated data and is generally provided in the IEEE 2030.5 hierarchy in the form of a self-signed certificate.

trusted root store: An integrity-protected location for storing root certificates.

3.2 Acronyms and abbreviations

ACL	access control list
CA	certificate authority
CRL	certificate revocation list
DER	distributed energy resource
DNS	Domain Name System
DNS-SD	DNS-based Service Discovery
DRLC	demand response and load control
ECDSA	Elliptic Curve Digital Signature Algorithm
EDC	EndDeviceControl
EMS	energy management system
ESI	energy services interface
EVSE	electric vehicle supply equipment
EXI	Efficient XML Interchange
HEMS	home energy management system
HTTP	Hypertext Transfer Protocol
IETF	Internet Engineering Task Force
IP	Internet Protocol
LFDI	long form device identifier
MCA	manufacturer’s certificate authority
MICA	manufacturing issuing certificate authority

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

OCSP	Online Certificate Status Protocol
OID	object identifier
PKI	public key infrastructure
REST	representational state transfer
SERCA	smart energy root certificate authority
SFDI	short form device identifier
TCP	Transmission Control Protocol
UOM	units of measure
URI	uniform resource identifier
WADL	Web Application Description Language
XML	Extensible Markup Language

4. Design pattern

4.1 Protocol flexibility

IEEE Std 2030.5 is designed to implement a REST architecture. It is built around the core actions of GET, HEAD, PUT, POST, and DELETE (as used in (Fielding [B2]), with the addition of a lightweight subscription mechanism as discussed in 8.9. Any application protocol that can implement a RESTful command set could likely be used with IEEE Std 2030.5, but Hypertext Transfer Protocol (HTTP) is a required baseline for interoperable IEEE 2030.5 implementations. HTTP utilizes Transmission Control Protocol (TCP) as its transport protocol. As a result, TCP manages the session providing delivery assurance and windowing.

IEEE 2030.5 servers and clients SHALL be compliant with IETF RFC 9110, IETF RFC 9111, and IETF RFC 9112.

4.2 General rules/best practices

This standard shall not make distinctions between servers, clients, or devices when defining interfaces and uniform resource identifiers (URIs). The goal is to avoid having a resource that has one behavior on a server and a different behavior on a client or different type of server. The distinction between the server and client role depends on whether a device exposes a resource (server) or interacts with the resource (client).

The default mechanism for obtaining a resource representation is a pull mechanism, implemented with GET. A client requests and retrieves data from a server or creates, modifies, or deletes data on a server.

The default polling rate for a given function set is specified at the top-level resource of that function set in the pollRate attribute.

Copyright © 2024 IEEE. All rights reserved.
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

The use of a subscription mechanism for retrieving a resource representation is also optionally supported, where convenient and appropriate. Resources that support subscription are denoted by their subscribable attribute.

Objects have a defined granularity and whole objects (not partial objects) are to be updated with the granularity defined in the schema.

Clients that expect to have intermittent connections to the network (e.g., battery-powered sleepy devices, mobile devices, etc.) use a pull mechanism as their default behavior for resource retrieval, as a Subscription/Notification mechanism may not be reliable. It should be noted that clients that expect to have intermittent connections to the network may still POST, PUT, and DELETE resources, provided they have the appropriate security permissions.

The TCP ports used for HTTP or Hypertext Transfer Protocol Secure (HTTPS) SHALL be specified in the mDNS and xmDNS service advertisements for the service.

Content SHALL be transferred with either one of the content types: “application/sep+xml” or “application/sep-exi.”

Devices do not assume the use of the URIs and their structures given throughout this standard. All resources are self-describing as it is acknowledged that URIs, schemas, and resources might change in the future. All resources SHALL contain links to their subordinate resources to support flexibility in URIs and future extensibility. Thus, to allow for extensibility and granularity, all objects are described in schemas and referenced via URIs. The URIs presented throughout this standard are recommendations. Thus, clients do not assume that URIs for resources are fixed on all servers or even on a given server (over time), but rather retrieve the appropriate URIs through resource discovery and links within resources. For network efficiency, devices MAY assume URIs are fixed on a particular server over time. If a URI returns an unexpected result, the client SHOULD execute resource discovery to determine the new URI value.

Version information should not be presented in the URI unless that version information is inherent to the name of that resource. If necessary and for reasons of extensibility, version information is provided within the associated resources and/or schemas.

All values in XML and EXI SHOULD be represented as compactly as possible. Decimal values SHOULD be represented without leading zeros. Hexadecimal values SHALL be represented with one leading zero, if needed, to ensure an even number of digits.

4.3 WADL

The IEEE 2030.5 RESTful interface is defined using the Web Application Description Language (WADL) (IEEE 2030.5 supplemental material).

IEEE 2030.5 devices SHALL conform to the interface specifications contained in the WADL as follows:

a) Devices SHALL conform to the WADL specification as per W3C WADL.

b) Devices SHALL conform to the WADL definition in the supplemental material of IEEE Std 2030.5. By implication, all resource representations SHALL validate per the schema (IEEE 2030.5 supplemental material) within the standardized IEEE 2030.5 XML namespace (`urn:ieee:std:2030.5:ns`).

c) Compliance (MODE) designations are interpreted as follows:

   1) Mandatory ("M"): Devices SHALL implement and conform.

   2) Optional ("O"): Devices MAY implement, and if implemented, SHALL conform.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

3) Discouraged ("D"): Devices SHOULD NOT implement, but if implemented, SHALL conform.

4) Error ("E"): Devices SHALL return one of the specified response status codes (e.g., 400 Bad Request or 405 – Method Not Allowed).

4.4 Schema

Resources located at URIs returned in the href attribute of "Link" specializations (e.g., EndDeviceListLink, SelfDeviceLink) SHALL conform to the schema definition for that object, which is the tag name with the "Link" suffix removed. For example, the resource at SelfDeviceLink href follows the definition for the SelfDevice resource.

If a client PUTs or POSTs a resource to a server containing attributes or elements that instead are to be populated by the server (e.g., href), the server SHALL return an HTTP 400 error.

If a function set is not implemented, Link elements to resources in that function set SHALL NOT be included.

4.5 Uniform resource identifiers

HTTP uses ASCII text for transferring URIs between clients and servers, as well as for including options and details regarding the message content. These transfers occur with every transaction. If the naming scheme used for URIs is overly verbose, these transactions become needlessly inefficient on constrained networks. Of course, if the naming scheme used is overly shortened, the advantages of a text-based protocol are lost, and it becomes difficult for someone troubleshooting to decipher a transaction.

The following conventions are used for URI naming:

a) URI elements are at most four characters, but recognizable to a knowledgeable engineer. Element names as short as one character are acceptable provided their meaning is clear.

b) URI elements are constructed of consonants only, unless inclusion of a vowel adds clarity, such as a leading vowel or well-known abbreviation.

c) URI elements are in all lower case.

d) URIs SHALL NOT be greater than 255 bytes in length. In practice, URIs SHOULD be much smaller than 80 bytes.

4.6 List resources

4.6.1 Introduction

Many resources within this standard are derived from the <List> object. Throughout this standard, these resources will collectively be referred to as *list resources*.

The following attributes are defined for list resources:

— `all` SHALL indicate the total number of items (subordinate resources) that exist in the list resource before any query string parameters are applied. This number may vary according to the client's access privileges.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

— `results` SHALL indicate the number of items (subordinate resources) included in a specific subset of the list (result from a paged GET query to the list, etc.). This value will always be less than or equal to "all."

Clients and servers use these attributes, combined with the query string parameters described below, to implement paged access to lists. Client control of list paging is important for resource-constrained devices.

List items (subordinate resources) are read using one of the two idioms described below:

1) Ordinal access to the first, second, nth, etc. item in the list is supported via query string parameters included with a GET to the list resource URI.

2) Random access to specific list items is supported via a GET directly to the URI of the list item.

Some list items may be created by POSTing a list item representation to the URI of the list (e.g., Notifications), while others may be created using private interfaces over the provider network. Ordinal placement of the new resource within the list is determined by the list sort order (defined by each list).

Each function set defined in this standard that contains list resources includes a list ordering table. Each list resource has an entry in the corresponding table that describes the details of one or more unique sort keys and the precedence of those keys. A list resource's elements SHALL be ordered according to this specified list ordering.

4.6.2 Query string parameters

Query string parameters are parameters added to a URI to provide filtering/paging of list items returned in query results.

The list paging mechanism allows GET requests to specify the range of list items to be returned in a query result set. The general syntax of a paged query is as follows:

```
{URI}?s={x}&a={y}&l={z}
```

Where `{URI}` represents a URI used to address a list resource, (`s | a | l`) represent query string parameters (further defined below), and `{x}`, `{y}`, and `{z}` represent the respective query string parameter values.

The query string parameters are defined as:

— `s` ("start") is used to indicate the first ordinal position in the list to be returned in the query result list as determined by the list's ordering. The value is specified in decimal. The first ordinal position of the list SHALL be designated with a value of "0" and the maximum possible value is "4294967295." If this query string parameter is not specified, the default start value SHALL be "0."

— `a` ("after") is used to indicate that only items whose primary key occurs after the given date/time parameter should be included in the query result list. This query string parameter is only applied to list resources that are ascending ordered using a time-based primary key. The parameter SHALL be ignored if the primary key is not time-based or if the ordering is not ascending. The format of the parameter SHALL be a 64-bit decimal number with identical semantics as that of the TimeType (see subclause 12.2 and the XML XSD in the supplemental material of IEEE Std 2030.5).

— `l` ("limit") is used to set the maximum number of list items (up to 4 294 967 295) to be included in the query result list. The value is specified in decimal. If this query string parameter is not specified, the default limit SHALL be "1." Servers MAY return a result list smaller than that specified by the client-provided limit.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

If both a “start” and “after” query string parameter are used simultaneously, the “after” query string parameter SHALL have precedence. The “start” position 0 SHALL be relative to the position specified by the “after” parameter.

If a query string requests a List element that does not exist (e.g., `s=3` when there are two items in the list), servers SHALL return an empty list representation.

Readers should note that the “after” query string parameter SHOULD NOT be used alone for paging through a list. As some list resources MAY contain multiple subordinate resources with the same time-based primary key, clients wishing to paginate a list resource while using the “after” query string parameter SHOULD keep the value for the “after” query string parameter constant while changing the “start” query string parameter.

If a particular query string parameter appears more than once, then the first occurrence of the query string parameter SHALL be used (in left-to-right order) and subsequent occurrences SHALL be ignored.

Server receipt of a query parameter unknown to the server SHALL be ignored by the server and SHALL NOT generate an HTTP error. Servers SHALL NOT generate resource representations containing `href` attributes that contain query parameters. Clients SHALL ignore query parameters contained in resource `hrefs`, but SHOULD NOT remove them if the URI is used for subsequent RESTful exchanges.

Should an empty list representation be requested (either through the use of query string parameters such as `l=0` or when the list itself is empty), the server SHALL return no subordinate representations, but SHALL return any other elements that may be defined for the list.

Clients SHALL NOT assume any index semantics for list URIs. For example, a client desiring to read the third item from the list at `http://some-host/somelist` SHALL NOT assume a GET to `http://some-host/somelist/2` will return the third item. The correct access is supported by the client issuing a GET to `http://some-host/somelist?s=2&l=1`.

The following examples demonstrate the use of query string parameters with a list resource. Consider the MyTypeList resource, depicted below:

```
<MyTypeList href="http://host1/the/list" all="7" results="7">
                <MyType href="http://host1/instance/of/type/red">
                                <timeStamp>100</timeStamp>
                </MyType>
                <MyType href="http://host2/instance/of/type/green">
                                <timeStamp>200</timeStamp>
                </MyType>
                <MyType href="http://host3/instance/of/type/blue">
                                <timeStamp>300</timeStamp>
                </MyType>
                <MyType href="http://host4/instance/of/type/yellow">
                                <timeStamp>400</timeStamp>
                </MyType>
                <MyType href="http://host5/instance/of/type/black">
                                <timeStamp>500</timeStamp>
                </MyType>
                <MyType href="http://host6/instance/of/type/white">
                                <timeStamp>600</timeStamp>
                </MyType>
                <MyType href="http://host7/instance/of/type/orange">
                                <timeStamp>700</timeStamp>
                </MyType>
</MyTypeList>
```

This list is sorted in ascending order per the `<timeStamp/>` element of `MyType`.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

A GET to `http://host1/the/list?s=0&l=1` will return:

```
<MyTypeList href="http://host1/the/list" all="7" results="1">
                <MyType href="http://host1/instance/of/type/red">
                                <timeStamp>100</timeStamp>
                </MyType>
</MyTypeList>
```

A GET to `http://host1/the/list?s=0&l=5` will return:

```
<MyTypeList href="http://host1/the/list" all="7" results="5">
                <MyType href="http://host1/instance/of/type/red">
                                <timeStamp>100</timeStamp>
                </MyType>
                <MyType href="http://host2/instance/of/type/green">
                                <timeStamp>200</timeStamp>
                </MyType>
                <MyType href="http://host3/instance/of/type/blue">
                                <timeStamp>300</timeStamp>
                </MyType>
                <MyType href="http://host4/instance/of/type/yellow">
                                <timeStamp>400</timeStamp>
                </MyType>
                <MyType href="http://host5/instance/of/type/black">
                                <timeStamp>500</timeStamp>
                </MyType>
</MyTypeList>
```

A GET to `http://host1/the/list?s=5&l=1` will return:

```
<MyTypeList href="http://host1/the/list" all="7" results="1">
                <MyType href="http://host6/instance/of/type/white">
                                <timeStamp>600</timeStamp>
                </MyType>
</MyTypeList>
```

A GET to `http://host1/the/list?s=5&l=5` will return:

```
<MyTypeList href="http://host1/the/list" all="7" results="2">
                <MyType href="http://host6/instance/of/type/white">
                                <timeStamp>600</timeStamp>
                </MyType>
                <MyType href="http://host7/instance/of/type/orange">
                                <timeStamp>700</timeStamp>
                </MyType>
</MyTypeList>
```

A GET to `http://host1/the/list?s=12&l=2` will return a list representation containing zero items:

```
<MyTypeList href="http://host1/the/list" all="7" results="0">
</MyTypeList>
```

or `<MyTypeList href="http://host1/the/list" all="7" results="0" />`

A GET to `http://host6/instance/of/type/white` will return:

```
<MyType href="http://host6/instance/of/type/white">
                <timeStamp>600</timeStamp>
</MyType>
```

A GET to `http://host1/the/list?a=400&l=4` will return:

```
<MyTypeList href="http://host1/the/list" all="7" results="3">
                <MyType href="http://host5/instance/of/type/black">
                                <timeStamp>500</timeStamp>
```


Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

```
                </MyType>
                <MyType href="http://host6/instance/of/type/white">
                                <timeStamp>600</timeStamp>
                </MyType>
                <MyType href="http://host7/instance/of/type/orange">
                                <timeStamp>700</timeStamp>
                </MyType>
</MyTypeList>
```

A GET to `http://host1/the/list?a=400&s=0&l=2` will return:

```
<MyTypeList href="http://host1/the/list" all="7" results="2">
                <MyType href="http://host5/instance/of/type/black">
                                <timeStamp>500</timeStamp>
                </MyType>
                <MyType href="http://host6/instance/of/type/white">
                                <timeStamp>600</timeStamp>
                </MyType>
</MyTypeList>
```

A GET to `http://host1/the/list?a=400&s=2&l=2` will return:

```
<MyTypeList href="http://host1/the/list" all="7" results="1">
                <MyType href="http://host7/instance/of/type/orange">
                                <timeStamp>700</timeStamp>
                </MyType>
</MyTypeList>
```

4.7 Resource design rules

The following rules apply to the design and use of list resources:

- List resources SHALL support “start” and “limit” query string parameters, thus always supporting paging.
- List resources that have a time-based primary key SHALL support the “after” query string parameters.
- All subordinate resources of list resources SHALL include an `href` attribute containing the URI of the subordinate resource.
- Each list resource described in this standard includes a list ordering that defines how the list is ordered, including the details of one or more keys used to order the list and the precedence of these keys.
- When queried, list resources SHALL return subordinate resources in the order defined by the list ordering.
- All subordinate resources of list resources that support multiple types, for example, NotificationList, SHALL include an `xsi:type` attribute. In this case, the XML Schema Instance Namespace must also be declared. See Annex C for examples.

The following rules apply to the design and use of non-list resources:

- Non-list resources SHALL NOT support the defined query string parameters. Query string parameters applied to non-list resource URIs SHOULD be ignored.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

4.8 Resource identification

Several resources defined in the schema (IEEE 2030.5 supplemental material) contain an mRID and a version to uniquely identify the resource. While the schema contains more information, it should be noted that if these resources are modified, their corresponding versions are incremented.

The following best practices for using mRID and version can lead to greater efficiency:

— Clients SHOULD only check the mRID and version of a resource, if applicable, to determine if the resource has been modified.

— For resources (e.g., DERProgram) containing links to other non-list resources (e.g., DefaultDERControl) that are not populated and are not expected to be populated by a client, servers SHOULD NOT include a link to an empty representation. This way, only the first resource needs to be polled to determine when the linked resource becomes available.

4.9 Backward and forward compatibility

It is desired to have both syntactic and semantic backward and forward compatibility between revisions of IEEE Std 2030.5. The XML schema (IEEE 2030.5 supplemental material) contains elements to support extensions, including manufacturer-specific proprietary extensions (see Clause 11), future IEEE 2030.5 elements, and future IEEE 2030.5 attributes, all while allowing schema validation using the XML schema. In the following rules, *unknown* is defined as contents that are not defined in the revision of IEEE Std 2030.5 being used by a device.

To ensure forward compatibility with future revisions of IEEE Std 2030.5:

— Devices SHALL ignore any XML elements defined for future extensibility (those that are named “r2_3”) or any unknown XML elements in a different namespace as allowed by the xs:any element.

— Devices SHALL ignore any unknown XML attributes allowed by the xs:anyAttribute element.

— Devices SHALL ignore any unknown query string parameters.

— Devices SHALL ignore any unknown enumeration values in an XML enumeration.

— Devices SHALL ignore any unknown bits in an XML bitmap.

— Devices SHALL ignore any XML elements with empty values, if the meaning of an empty value is undefined and the element has no mandatory (i.e., minOccurs > 0) child elements.

— Devices logging information or presenting information to a user SHOULD consider logging or presenting unknown XML elements, attributes, etc.

— Devices acting as a proxy SHOULD NOT remove unknown XML elements, attributes, etc. from resources when they are sent to the proxied device(s).

To ensure backward compatibility with older revisions of IEEE Std 2030.5, the following guidelines are to be considered for revisions of IEEE Std 2030.5:

— No XML elements are to be removed (deprecation is allowed).

— No XML attributes are to be removed (deprecation is allowed).

— No query string parameters are to be removed (deprecation is allowed).

— No XML components’ or query string parameters’ semantics are to be modified.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

- No changes are to be made that further restrict an XML component (e.g., no changing a choice to a sequence).
- No XML ranges are to be expanded (i.e., minOccurs cannot be decreased and maxOccurs cannot be increased).
- XML elements added to existing XML elements are to be placed only in XML elements defined for extensibility (those that are named “r2_3”).
- XML elements added to existing XML elements are to be optional but may contain their own mandatory components. These optional elements may be mandatory going forward, recognizing they will not be present in older revisions.
- XML attributes are to be added only to existing XML elements that allow additional attributes as denoted with the xs:anyAttribute element.
- XML attributes added to existing XML elements are to be optional.
- Enumeration values added to existing XML enumerations are to be optional.
- Bitmap values added to existing XML bitmaps are to be optional.
- XML datatypes are to be changed with care.
- Query string parameters added are to be optional.
- The structure of XML elements are not to be changed (but can be expanded).

Should it be desired to break backward compatibility in the future, a new XML namespace is to be considered.

5. Application support

5.1 Overview

This clause details the standards-based application transport protocol and other lower layer requirements required for interoperability of IEEE 2030.5 devices.

The application support layer provides the following services:

- RESTful HTTP/1.1 as the application data exchange semantics, as described in IETF RFC 9110 and (Fielding [B2])
- XML (W3C XML) and/or EXI (W3C EXI) encoding as the data payload of the RESTful operations
- Transport authentication and encryption using HTTP over TLS (IETF RFC 2818 and IETF RFC 5246)

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

5.2 Use of TCP

The choice of HTTP/1.1 as the application data exchange protocol leads directly to the use of TCP as the transport protocol. IETF RFC 9112 states that:

"HTTP messaging is independent of the underlying transport- or session-layer connection protocol(s). HTTP only presumes a reliable transport with in-order delivery of requests and the corresponding in-order delivery of responses. The mapping of HTTP request and response structures onto the data units of an underlying transport protocol is outside the scope of this specification."

Hence, albeit that other transport protocols may be used to carry HTTP, they are required to provide reliable transport. Clearly, the de facto choice is TCP (IETF RFC 793).

5.3 URI encoding

The address of the resources presented by an IEEE 2030.5 host will use the standard URI syntax specific to HTTP/1.1 (i.e., `http://`) as per IETF RFC 1630 and IETF RFC 3986.

Given the constrained nature of many IEEE 2030.5 hosts, it is critical to devise the URL namespace scheme (hierarchy) that is both descriptive and compact. For more information, see 4.5.

URIs generated by IEEE 2030.5 hosts SHOULD use only mDNS domain names, xmDNS domain names, or Internet Protocol (IP) addresses in fully qualified URIs, as IEEE 2030.5 devices may not have access to name servers.

5.4 HTTP headers

5.4.1 Introduction

HTTP/1.1 defines a variety of header types that have the potential to be verbose. As an example, the Accept header is illustrated below showing lengths varying from 35 octets to 106 octets:

- `Accept: audio/*; q=0.2, audio/basic` (35 octets)
- `Accept: text/*; q=0.3, text/html; q=0.7, text/html; level=1, text/html; level=2; q=0.4, */*; q=0.5` (92 octets)
- `Accept: text/xml, application/xml, application/xhtml+xml, text/html; q=0.9, text/plain; q=0.8, image/png, */*; q=0.5` (106 octets)

Given the packet size constraints for many IEEE 2030.5 networks, the set of HTTP headers supported must be curtailed through best practices recommendations realizing that a host cannot prevent a verbose message from being sent to it.

IEEE Std 2030.5 provides a set of *mandatory, optional,* and *discouraged* recommendations for each HTTP header. IEEE 2030.5 implementations SHOULD employ only mandatory HTTP headers, minimize the use of optional HTTP headers, and avoid the use of discouraged headers.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

5.4.2 HTTP header field recommended usage

In Table 1, the HTTP/1.1 header fields have been annotated with the following labels:

— MANDATORY: Support for the field is REQUIRED.

— OPTIONAL: Support for this field is left to the implementer's discretion.

— DISCOURAGED: To conserve code space and/or bandwidth, support for this field, while not explicitly forbidden, is not recommended.

Table 1 summarizes the recommended use of HTTP/1.1 headers in IEEE Std 2030.5. This table is provided as informative guidance, particularly to suit constrained networks and devices, and IETF RFC 9110, IETF RFC 9111, and IETF RFC 9112 are the authoritative references for compliance.

Table 1—HTTP headers

Header	Used in message type		IETF RFC required/optional	IEEE 2030.5 use
Accept	Request		Optional	Mandatory
Accept-Charset	Request		Optional	Discouraged
Accept-Encoding	Request		Optional	Discouraged
Accept-Language	Request		Optional	Discouraged
Accept-Ranges	Request	Response	Optional	Discouraged
Age		Response	Optional (required for a cache)	Discouraged
Allow		Response	Required	Mandatory
Authorization	Request	Response	Optional	Discouraged
Cache-Control	Request	Response	Optional	Discouraged
Connection	Request		Optional (required in some situations [e.g., HTTP/1.1 applications that do not support persistent connections])	Optional (mandatory in some situations)
Content-Encoding		Response	Optional (required when an encoding is applied)	Discouraged
Content-Language		Response	Optional	Discouraged
Content-Length	Request	Response	Optional (required in many situations, see Section 8.6 of IETF RFC 9110)	Optional (required in many situations, see Section 8.6 of IETF RFC 9110)
Content-Location		Response	Optional	Discouraged
Content-MD5		Response	Optional	Discouraged
Content-Range		Response	Optional	Optional (see 9.8.2.3)
Content-Type	Request	Response	Required	Mandatory
Cookies	Request		Optional	Discouraged
Date	Request	Response	Mandatory	Mandatory
Etag		Response	Optional	Optional (see 9.8.2.3)
Expect	Request		Optional	Discouraged
Expires		Response	Optional	Discouraged
From	Request		Optional	Discouraged
Host	Request		Required	Mandatory
If-Match	Request		Optional	Discouraged
If-Modified-Since	Request		Optional	Discouraged

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Header	Used in message type		IETF RFC required/optional	IEEE 2030.5 use
If None-Match	Request		Optional	Discouraged
If-Range	Request		Optional	Discouraged
If-Unmodified-Since	Request		Optional	Discouraged
Last-Modified		Response	Optional	Discouraged
Location		Response	Optional	Mandatory in many situations (e.g., POST responses)
Max-Forwards	Request		Optional	Discouraged
Pragma	Request	Response	Optional	Discouraged
Proxy-Authenticate		Response	Optional	Discouraged
Proxy-Authorization	Request		Optional	Discouraged
Range	Request		Optional	Optional (see 9.8.2.3)
Referer	Request		Optional	Discouraged
Retry-After		Response	Optional	Discouraged
Server		Response	Optional	Discouraged
Set-Cookie		Response	Optional	Discouraged
TE	Request		Discouraged	Discouraged
Trailer		Response	Discouraged	Discouraged
Transfer-Encoding		Response	Optional	Discouraged
Upgrade	Request		Optional	Discouraged
User-Agent	Request		Optional	Discouraged
Vary		Response	Discouraged	Discouraged
Via	Request	Response		Discouraged
Warning	Request	Response	Discouraged	Discouraged
WWW-Authenticate		Response	Optional	Discouraged

5.5 HTTP response codes

5.5.1 Introduction

Response codes are expected to be generalized across RESTful platforms. The specific uses detailed below are likely to be generalized. In the interest of clarity and completeness, they are included here. Please note that these response codes follow general best practices for RESTful interfaces, though they are tuned to address some of the limitations of the embedded space.

This subclause attempts to highlight HTTP response codes that are felt to be more important or that need special attention from developers. However, IEEE 2030.5 clients may encounter various HTTP response codes and, all use of, and response to HTTP response codes SHALL be standard and RFC compliant.

5.5.2 Common responses

5.5.2.1 Introduction

The following HTTP response codes are those considered to be of utmost importance for this standard.

Copyright © 2024 IEEE. All rights reserved.
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

5.5.2.2 1xx (Informational)

These response codes are informational in purpose and are used to indicate that the server is continuing to process in some fashion.

IETF RFC 9110 states, “A client MUST be able to parse one or more 1xx responses received prior to a final response, even if the client does not expect one. A user agent MAY ignore unexpected 1xx responses.”

5.5.2.3 200 ("OK")

This response code is sent to indicate a successful transaction.

This response code is often used in response to a successful GET request, with the entity-body containing a representation of the requested resource. Use of this response code in response to PUT, POST, or DELETE requests is discouraged, to avoid the potentially unnecessary traffic generated by returning the resource representation in the entity-body (see 201 [“Created”] and 204 [“No Content”]).

5.5.2.4 201 ("Created")

This response code is sent to indicate a new resource has been created, at the client’s request with a PUT or POST.

The Location header SHALL be used in conjunction with this response code to indicate the URI of the newly created resource. The inclusion of a representation of the newly created resource in the entity-body of the response is discouraged, to conserve bandwidth.

IETF RFC 9110 states, “If the target resource does not have a current representation and the PUT successfully creates one, then the origin server MUST inform the user agent by sending a 201 (Created) response.”

5.5.2.5 204 ("No Content")

This response code is sent to indicate a successful transaction, but one where the response does not include an entity-body.

This response code is often used in response to a successful PUT or POST request, where the resource is modified, not created. This response code is also sent in response to a successful DELETE request. This response code is also sent in response to a successful GET request, where the resource exists but has an empty representation.

Further, when there are URIs that point to a resource that does not yet have content (an “empty representation”), this response code SHOULD be returned. For instance, if a client created a new resource with a POST and that new resource contains URIs pointing to resources that were not yet created and then a client were to request those linked resources, this response code would be the best response. When those resources are created (via a PUT, for instance), this response code (204) SHOULD be returned (in response to the PUT, for instance).

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

5.5.2.6 206 ("Partial Content")

This response code is sent to indicate the server has fulfilled the partial GET request (as specified by the Range header) for a resource. Note that IETF RFC 9110 requires the Content-Range and Date headers SHALL be present in the response.

5.5.2.7 301 ("Moved Permanently")

This response code is sent to indicate that the requested resource has a new URI. The Location header SHOULD be used in conjunction with this response code to indicate the new URI of the requested resource. The entity-body of the response SHOULD be empty. Upon unexpected receipt of this response code, clients SHOULD perform resource discovery to determine which resources have changed location.

5.5.2.8 302 ("Redirect")

The Location header SHALL be used in conjunction with this response code to indicate the new URI of the requested resource. The entity-body of the response SHOULD be empty.

This response code is often used to redirect URIs requested as HTTP to HTTPS.

5.5.2.9 400 ("Bad Request")

This response code is used to indicate a client-side error and is used when no other 4xx response code is appropriate. Often, this response code indicates that the representation sent by a client with a PUT or POST is not appropriate or is malformed.

IETF RFC 9112 states, “A server MUST respond with a 400 (Bad Request) status code to any HTTP/1.1 request message that lacks a Host header field and to any request message that contains more than one Host header field line or a Host header field with an invalid field value.”

5.5.2.10 401 ("Unauthorized")

This response code is used when a client does not have proper authorization to perform the requested action on a resource.

Note, if a server did not wish a client to know of the existence of the resource, it should instead send a 404 (“Not Found”) response code.

IETF RFC 9110 states, “A server generating a 401 (Unauthorized) response MUST send a WWW-Authenticate header field containing at least one challenge.”

5.5.2.11 404 ("Not Found")

This response code is used to indicate that no resource can be found at the specified URI.

This response code MAY also be used in lieu of a 401 response code.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

5.5.2.12 405 ("Method Not Allowed")

This response code is used to indicate that the resource does not allow the HTTP method used by the client.

IETF RFC 9110 states, “The origin server MUST generate an Allow header field in a 405 response containing a list of the target resource’s currently supported methods.”

5.5.2.13 406 ("Not Acceptable")

This response code is used to indicate that a server is unable to generate a response that is acceptable according to the Accept headers sent in the request.

5.5.2.14 413 ("Request Entity Too Large")

This response code is used to indicate that a server is refusing to process a request, as the request is larger than the server is willing or able to process.

5.5.2.15 416 ("Requested Range Not Satisfiable")

This response code is used to indicate that a server has received a Range request that does not overlap any of the resource content.

5.5.2.16 417 ("Expectation Failed")

This response code is used to indicate that an expectation given in an Expect request-header field cannot be met by the server or that the server does not support the given expectation.

5.5.2.17 429 ("Too Many Requests")

This response code is used to indicate that a user has sent too many requests within a given amount of time.

5.5.2.18 500 ("Internal Server Error")

This response code is used to indicate that the server has an internal problem and is a generic server error response.

5.5.2.19 501 ("Not Implemented")

This response code is used when a client attempts to use a feature of HTTP (such as a method) that the server does not support.

5.5.2.20 503 ("Service Unavailable")

This response code is used when a server, due to a temporary overload condition, is unable to service a request.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

5.5.3 Minimal understanding

Should a client wish to operate with minimal understanding of HTTP response codes, it need only examine the first digit of the response code to understand the general category of the response and "treat an unrecognized status code as being equivalent to the x00 status code of that class" (IETF RFC 9110).

5.6 Application payload syntax

5.6.1 Introduction

Application payload message encoding using both XML (W3C XML) and EXI (W3C EXI) SHALL be supported by all servers. Application payload message encoding using either XML (W3C XML) or EXI (W3C EXI) SHALL be supported by all clients. Certain function sets may require additional encodings and, if so, are specified in their respective sections.

5.6.2 XML encoding

The XML declaration is optional as per W3C XML and SHOULD NOT be included in IEEE 2030.5 transactions, to reduce packet sizes. The XML version used SHALL be 1.0. For XML payloads, the encoding SHALL be UTF-8.

All XML payloads SHALL include a schemaVer attribute in the top-level XML element equal to the version of the IEEE 2030.5 schema (IEEE 2030.5 supplemental material) used (e.g., `schemaVer="2.2"`). The version used for this revision SHALL be 2.2. It should be noted that previous revisions of IEEE Std 2030.5 did not require this schemaVer attribute and thus are assumed by newer devices to use the default version of 2.1.

5.6.3 EXI encoding

The options for encoding EXI documents SHALL be as follows, and transactions will likely fail if different options are declared in the EXI option header. Options marked as (default) are EXI specification (W3C EXI) defaults and SHOULD NOT be specified explicitly.

- Non-strict schema-informed grammar with the schema (IEEE Std 2030.5 supplemental material)
- Alignment is bit-packed (default)
- Compression is false (default)
- Strict is false (default)
- Fragment is false (default)
- Preserve options are all false (default)
- selfContained is false (default)
- schemaId is "S2" (two bytes: 0x53, 0x32, without quotes). This schemaId corresponds to the normative schema of IEEE Std 2030.5-2023 (supplemental material).
- datatypeRepresentationMap is not used (default)
- valueMaxLength is unbounded (default)
- valuePartitionCapacity is unbounded (default)
- No user defined meta-data

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

The following XML document describes the EXI option header that SHALL be used to encode the messages:

```
<header xmlns="http://www.w3.org/2009/exi">
                <common><schemaId>S2</schemaId></common>
</header>
```

5.7 Content negotiation

5.7.1 General

A client SHALL declare acceptable media types using the HTTP Accept header.

5.7.2 Schema version negotiation

When specifying an “application/sep-exi” media type, an extensibility level (“level”) SHALL be specified in the HTTP Accept header for schema version negotiation. The extensibility level specified using an HTTP Accept header supersedes an extensibility declaration discovered during resource discovery (see 7.4).

The extensibility level defines the base schema and its capability for arbitrary extension. The extensibility level is one of “-S2” or “+S2”. The S2 indicates the base schema version: IEEE Std 2030.5-2023. “-S2” indicates the node does not accept arbitrary tags that are not defined in the base schema, and “+S2” indicates it accepts arbitrary tags. A node with “-S2” will likely fail on an EXI document using arbitrary types, elements, and attributes that are not defined in the schema used for encoding. Devices SHALL NOT send messages to nodes that declare “-S2” using arbitrary types, elements, and attributes.

The grammar used for EXI SHALL be generated as a non-strict grammar only, as having both strict and non-strict grammars would put a large burden on storage requirements for certain devices. The use of a non-strict grammar allows for extensions without schema modification. An invalid (i.e., not defined in the schema) part of an EXI document is allowed in a non-strict grammar and can carry arbitrary tags, attributes, and text encoded using the built-in grammar.

Due to strict memory constraints, some nodes may not be able to parse invalid parts of an EXI document encoded using the built-in grammar. To avoid such errors, a node may declare its inability to receive arbitrary extensions using the “-” (minus) prefix in the extensibility level. Alternatively, nodes that declare the “+” (plus) prefix in the extensibility level will be able to parse extended parts of an EXI document.

Note that the extensibility level does not indicate whether the node can process the data, but only whether it can parse the data.

The format of the extensibility level is “(-/+)Sn” where n is a character to describe base schema version (currently “2”). As extensions of IEEE 2030.5 schemas are intended to be backward compatible, a node that declares schemaId “S[i]” is intended to be compatible with all versions between “S2” and “S[i]”. However, an “S2” implementation is not backward compatible with an “S0” (IEEE Std 2030.5-2013) implementation but is backward compatible with an “S1” (IEEE Std 2030.5-2018) implementation.

For example:

```
Accept: application/sep-exi; level=-S2
```

indicates that the client wishes to receive content encoded using EXI where the base schema of the client is IEEE Std 2030.5-2023 and that the client does not accept arbitrary tags not defined in the schema.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

A client SHOULD use the extensibility level discovered during resource discovery to determine if a server accepts non-strict parts of an EXI document prior to initiating PUT/POST operations where the content contains extended attributes/elements. A server SHOULD use the extensibility level specified in the Accept header to determine if a client accepts non-strict parts of an EXI document prior to responding to GET operations where the content contains extended attributes/elements.

6. Security

6.1 Introduction

Depending on the underlying physical network, messages may be encrypted at lower layers, in addition to the security features provided specifically for the application layer. This clause describes the security features that are provided at the application layer and that SHALL be used over all networks.

Securing transactions between clients and servers is based on using HTTP over TLS (IETF RFC 2818) (also known as HTTPS) using TLS version 1.2 (IETF RFC 5246). The TLS records are then transported using TCP. The TLS handshake mechanism provides mutual authentication based on device certificates or self-signed certificates and TLS records provide encryption and message authentication using the AES-CCM mode of operation. Access control lists allow or deny use of resources based on authentication level and address information. A registration list is used for authorizing clients.

This standard does not mandate a specific security policy or specific access controls as they may vary depending on the device, network, and use case. Readers should note that many resources are available for guidance in creating a security policy and for access control, including those from NIST and OWASP. A default security policy is described in 6.8.

6.2 Security attributes

6.2.1 Introduction

In this subclause, we define some abstract data structures for managing registration (see 6.9) and access control. How this functionality is accomplished is left to the implementer. No access to these data structures is defined in this standard.

6.2.2 Local registration attributes

Local registration attributes represent the data which would be used to hold information passed out-of-band as part of the registration process prior to resources being established (Table 2 and Table 3).

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Table 2—Local registration attributes

Attribute	Identifier	Type	Range	Description	Default
aclLocalRegistration	0x00	List	-	A table of Registration Descriptors each with information for a specific registration	(empty)
aclLocalRegistrationEntries	0x01	Integer	Implementation specific	Number of entries in *aclRegistration*	0

Table 3—Local registration descriptor entry

Name	Type	Range	Description	Default
PIN	Integer	0 to 999999	6-digit personal identification number (PIN) (5 plus check digit) for basic server validation. The PIN is also reflected in the Registration resource linked to the EndDevice resource.	-
SFDI	SFDI	-	SFDI of registering device. The SFDI is also reflected in the EndDevice resource.	-
DeviceType	Integer	0 to 3	Minimum required device type	0
HardwareModuleName	String	-	Optional additional hardware module information	-

6.2.3 Access control list (ACL) attributes

6.2.3.1 Introduction

Access control list (ACL) attributes represent the data that would be used to hold information to determine whether access to a particular resource by a particular client is allowed or denied (Table 4, Table 5, and Table 6). An ACL can enforce more granular access control based on various criteria (e.g., client identity). Conceptually, an ACL exists for every single accessible resource; however, in practice, it is likely only certain resources with more complex access policies would require a representation of all the data specified in this subclause.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Table 4—ACL attributes

Attribute	Identifier	Type	Range	Description	Default
aclDefaultAccess	0x02	Access descriptor		Default access to resource	-
aclSpecificID	0x03	List		A list of specificIDDescriptors for each specific client access to resource	-
aclSpecificIDEntries	0x04	Integer	Implementation specific	Number of entries in *aclSpecificID*	0

Table 5—AccessDescriptorEntry

Name	Type	Range	Description	Default
Method	Bitmap	0x0 to 0xf	Bitmap of which methods are supported: 0x1: GET 0x2: PUT 0x4: POST 0x8: DELETE 0x10: HEAD	0x0
AuthType	Integer	0x0 to 0x0f	Bitmap of which authentication types are allowed: 0x1: No authentication 0x2: User authentication 0x4: Self-signed certificate 0x8: Device certificate Remaining bits are reserved for authentication types not defined by this standard but by an additional security policy.	0x0
DeviceType	Integer	0 to 3	DeviceType: 0: Any device type	0

Table 6—SpecificIDDescriptor entry

Name	Type	Range	Description	Default
Access	Access descriptor	-	Access levels required for client	-
IPAddr	IPAddr	-	IP address of client	-
Port	Integer	0x0000 to 0xffff	Port of client 0: Wildcard (any port)	0

ACL attributes provide a mechanism for granting and revoking privileges to use specified methods with a particular resource, applicable to all resources described in this standard. Access control that is more granular than a resource is out of scope for this standard. The mechanisms for authentication and the binding of that authentication to a specified identity (such as an IP address) are specified in 6.5 and 6.6.

ACLs are used to set default privileges and grant additional privileges, not to deny privileges. Thus, if a given client's request does not explicitly meet the required privilege in the associated ACL, then that client does not have that privilege. The default configuration of an ACL for a given resource means that resource is not accessible to clients. In practice, this state only exists ephemerally and all ACLs will be initialized appropriately at startup according to the security policy and will be subsequently modified according to registration and authentication.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Subordinate resources created dynamically will initially inherit their ACL from their parent resource.

ACLs may be statically fixed with a default operation for some resources and may be dynamic and extensible for others. If a resource does not have an ACL, access is granted to the resource unconditionally.

Initialization of ACLs, beyond minimal requirements, is out of scope for this standard and is governed by overall security policy. Default settings for ACLs for particular function sets are described in 6.8.

6.2.3.2 aclDefaultAccess

aclDefaultAccess in the ACL is used for settings irrespective of the client identity of an incoming request and is used initially to authorize an incoming HTTP request.

6.2.3.3 aclSpecificIDList

A SpecificIDDescriptor entry in the *aclSpecificIDList* part of the ACL is an additional entry used to allow specific additional checks to be done to authorize based on client identity (IP address and port).

6.2.3.4 Access authorization

6.2.3.4.1 Introduction

These are controls that are independent of the source identity (IP address and port) and thus can be configured in both *aclDefaultAccess* and a SpecificIDDescriptor entry in *aclSpecificIDList* as shown in Table 5 and Table 6.

In the following, *corresponding ACL entry* means:

- The first SpecificIDDescriptor entry in *aclSpecificIDList* which matches the incoming client HTTP request's source IP address and port
- *aclDefaultAccess* otherwise

6.2.3.4.2 Method attribute

The Method attribute is used to control which HTTP request method is allowed for client access.

- GET
- PUT
- POST
- DELETE
- HEAD

The HTTP method of an incoming HTTP request is checked and Method authorization will be TRUE if the method is allowed in the corresponding ACL entry.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

6.2.3.4.3 AuthType attribute

The AuthType attribute is used to control the required authentication types the client can use in its incoming HTTP/HTTPS request.

- 0x1: No authentication
- 0x2: User authentication
- 0x4: Self-signed certificate
- 0x8: Device certificate

If an incoming HTTP request is destined for the port associated with HTTPS (typically port 443), the incoming authentication type will be set to the corresponding TLS session authentication type. If an incoming HTTP request is destined for the port associated with HTTP (typically port 80), the authentication type will be set to 0x1 (no authentication). The user authentication AuthType (0x02) can be used if additional user authentication outside of the scope of this standard takes place.

The incoming authentication type is compared with the bitmap in AuthType in the corresponding ACL entry and AuthType authorization will be TRUE if the corresponding authentication type is set in AuthType in the corresponding ACL entry. Representing this logically:

```
if (authentication type & AuthType) != 0:
                AuthType authorization = TRUE
else
                AuthType authorization = FALSE
```

If AuthType authorization is FALSE, the incoming HTTP/HTTPS request is expected to be rejected. The use of HTTPS is strongly recommended, and more detailed recommendations regarding a default security policy can be found in 6.8.

6.2.3.4.4 DeviceType attribute

The DeviceType attribute is used to control the device type required of the client's incoming HTTP request. It is based on the deviceType OID in the certificate (see 6.11.7.2).

If an incoming HTTP request is destined for the port associated with HTTPS, the incoming device type will be set to the corresponding TLS session device type based on the deviceType OID in the certificate. If an incoming HTTP request is destined for the port associated with HTTP, the device type will be set to 0 (any device type).

If the DeviceType in the ACL is set to 0, device type authorization will be TRUE unconditionally. Otherwise the incoming device type is compared with the DeviceType in the corresponding ACL entry and DeviceType authorization will be TRUE if the device type is equal to the DeviceType in the corresponding ACL entry.

6.2.3.5 Authorization logic

Authorization is granted if Method authorization, AuthType authorization, and DeviceType authorization are all TRUE. Access to the resource can then take place.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

If Method authorization is not granted, the server MAY respond with either:

```
HTTP/1.1 400 Bad Request
```

or

```
HTTP/1.1 405 Method Not Allowed
```

If AuthType or DeviceType authorization is not granted, the server SHOULD immediately respond:

```
HTTP/1.1 404 Not Found
```

The server MAY respond:

```
HTTP/1.1 401 Unauthorized
```

Otherwise, processing will continue.

6.2.3.6 ACL examples

6.2.3.6.1 Introduction

This subclause contains two informative examples to illustrate the use of ACLs, using EndDeviceList and EndDevice.

6.2.3.6.2 EndDeviceList

The EndDeviceList resource is usually accessible by any client to find its own entry in the list. Therefore, the ACL will be as shown in Table 7.

Table 7—Example ACL for EndDeviceList

Attribute	Comment
aclDefaultAccess	Method: 0x01 (GET only) AuthType: 0x1 (no authentication) DeviceType: 0 (any device type)
aclSpecificID	Empty
aclSpecificIDEntries	0

In this example, there will never be any SpecificIDDescriptors required, as this resource needs to be accessible to any device.

6.2.3.6.3 EndDevice

In this example, an EndDevice resource for a client does not exist prior to registration.

The earliest point it can be created is at the point of registration and the ACL would be set as shown in Table 8.

Copyright © 2024 IEEE. All rights reserved.
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Table 8—Example ACL entry for EndDevice at point of registration

Attribute	Comment
aclDefaultAccess	Method: 0x00 (no default access) AuthType: 0x1 (no authentication) DeviceType: 0 (any device type)
aclSpecificID	Empty
aclSpecificIDEntries	0

In this example, there is no default access as only the client device associated with the EndDevice is able to access the EndDevice. Also, at this point, there has been no client communication, therefore there would be no SpecificIDDescriptor entry in *aclSpecificIDList*.

There would also be an entry placed in the local registration list corresponding to the client, as shown in Table 9.

Table 9—Example local registration entry for registering device

Name	Type	Range	Description	Default
PIN	Integer	0 to 999 999	6-digit PIN (5 plus check digit) of registering device	-
SFDI	SFDI	0 to 68 719 476 735, 2^{36}–1	SFDI of registering device	-
DeviceType	Integer	0 to 3	Registering device type	0
HardwareModuleName	String	-	Optional additional hardware module information	-

In this state, it is primed to be populated with a SpecificIDDescriptor when the device actually accesses the EndDeviceList resource.

When the client attempts access to any resource on a function set server using TLS, it will start performing the TLS handshake, which involves transferring the client's certificate. At the point of access, if there is a pending registration, it will be checked against the client's device. If there is a match, and the validated certificate's SFDI matches, the ACL for the EndDevice will be populated with an additional SpecificIDDescriptor as shown in Table 10 and Table 11.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

Table 10—Example ACL entry for EndDevice at point of client access

Attribute	Comment
aclDefaultAccess	Method: 0x00 (no default access) AuthType: 0x1 (no authentication) DeviceType: 0 (Any device type)
aclSpecificID	One SpecificIDDescriptor for client
aclSpecificIDEntries	1

Table 11—Example SpecificIDDescriptor for EndDevice at point of client access

Name	Type	Range	Description	Default
Access	AccessDescriptor	-	Access levels required for client: Method: 0xF (GET, PUT, POST, DELETE) AuthType: 0x8 (device certificate) DeviceType: (as appropriate for device)	-
IPAddr	IPAddr	-	IP address of client	-
Port	Integer	0x0000 to 0xffff	Port of client	0

6.3 Device credentials

6.3.1 Introduction

There are three credentials per device:

— Short form device identifier (SFDI)

— Long form device identifier (LFDI)

— PIN

6.3.2 Certificate fingerprint

The certificate fingerprint is the result of performing a SHA256 operation over the whole DER-encoded certificate and is used to derive the SFDI and LFDI. A certificate fingerprint is not confidential and is never used to derive subsequent keying material.

An example certificate fingerprint used for illustration in the following examples is:

3E4F-45AB-31ED-FE5B-67E3-43E5-E456-2E31-984E-23E5-349E-2AD7-4567-2ED1-45EE-213A

6.3.3 Short-form device identifier (SFDI)

The SFDI SHALL be the certificate fingerprint left-truncated to 36 bits. For display purposes, the SFDI SHALL be expressed as 11 decimal (base 10) digits, with an additional sum-of-digits checksum digit right-concatenated. Based on the example in 6.3.2, this would be 167-261-211-391.

Left truncation to 36 bits: 0x3E4F45AB3

Expressed as a decimal: 16726121139

Right-concatenation of check digit and hyphenation: 167-261-211-391

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

For input validation purposes, the sum of the digits of the SFDI including the checksum digit, modulo 10, SHALL be zero. The SFDI has sufficient entropy (2^{36}) to uniquely identify the device in the context of its usage and is used to identify a device within a home area network (HAN) or site domain. It should not be used in a truly global context (i.e., where the identity of the device cannot be qualified with the domain it is in).

For a device with a device certificate, the SFDI can be printed on the device packaging.

6.3.4 Long-form device identifier (LFDI)

The LFDI SHALL be the certificate fingerprint left-truncated to 160 bits (20 octets). For display purposes, this SHALL be expressed as 40 hexadecimal (base 16) digits in groups of four. Based on the example in 6.3.2, this would be "3E4F-45AB-31ED-FE5B-67E3-43E5-E456-2E31-984E-23E5." The LFDI is used when a globally unique identity is required, for example in sending an event back to a service provider that is associated with a particular device.

For proxied devices or DERComponents that do not natively communicate using IEEE Std 2030.5, the LFDI SHALL be composed of a random 128-bit number with the 32-bit PEN of the proxy or end device right-concatenated, traceable by the proxy or end device to the associated device or DERComponent. The proxy or end device SHALL ensure that random number generators maintain a distribution of values over the population of proxied devices or DERComponents (e.g., by using different seeds). Note that the use of a random number is to ensure global uniqueness without the need for centralized coordination. The SFDI of these proxied devices SHALL be this LFDI left-truncated to 36 bits.

6.3.5 6-digit PIN code

The SFDI and LFDI are derived from public information (i.e., a Certificate), therefore can potentially be recreated by an eavesdropper. Therefore, a device MAY also have an additional 6-digit PIN code, which can be shared out-of-band with a service provider in conjunction with the SFDI or LFDI. For display purposes, this SHALL be expressed as 5 decimal (base 10) digits, with an additional sum-of-digits checksum digit right-concatenated:

Original PIN: 12345

Right-concatenation of check digit and hyphenation: 123-455

For input validation purposes, the sum of the digits of the PIN including the checksum digit, modulo 10, SHALL be zero. The PIN MAY be obtainable from the EndDevice server through the Registration resource to validate that the client is in communication with the correct server. The PIN SHOULD be configurable on a device where possible for registration purposes, otherwise SHOULD be a random 5-digit value plus check digit pre-programmed into the device and printed on the device. The PIN is not overly secure and therefore SHALL NOT be used in any way to derive keys for actual data encryption.

6.3.6 Registration code

The SFDI and PIN are usually presented separately. However, in certain cases it may be convenient to provide a single registration code, which is simply the concatenation of the SFDI and the PIN expressed as a decimal (base 10) number:

SFDI || PIN

From the examples above, this would be 167-261-211-391-123-455.

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023

6.4 Resource access authentication and authorization context

A node is able to perform network layer communication once it has been authenticated and authorized to join the network. However, application layer authentication and authorization MAY be required before clients and servers can exchange application layer messaging. Registration (see 6.9) with a utility or third party service provider MAY also be needed to provide explicit device and user authorization at the application layer.

Resource access requiring application layer authentication, data confidentiality, and integrity checking SHALL occur through requests from a client to the server using HTTP over TLS (IETF RFC 2818) (also known as HTTPS) using TLS version 1.2 (IETF RFC 5246). Resource access not requiring application layer authentication, data confidentiality, or integrity checking SHALL occur through requests from a client to the host server using HTTP (IETF RFC 9110 and IETF RFC 9112).

If a request is made to the port associated with HTTPS, it is considered an HTTPS request and authentication SHALL have taken place. If authentication has not taken place, authentication SHALL be initiated as described in 6.5. When authenticated, the request is then passed to the ACL associated with the resource. Ancillary information about the request obtained from the secure session, notably the level of client authentication, will also be compared with the ACL.

If the request is made to the port associated with HTTP, it is considered an HTTP request and authentication SHALL NOT be REQUIRED and the request is then passed to the ACL associated with the resource. Ancillary information about the request stating that the client is unauthenticated will also be compared with the ACL (see 6.2.3.4.3).

Authorization on a request-by-request basis is determined by the ACL settings for the resource, which may be set up at the end of the authentication based on the level of client authentication. The Local Registration List (*aclLocalRegistrationList*) may be additionally used to authorize on a device-by-device basis.

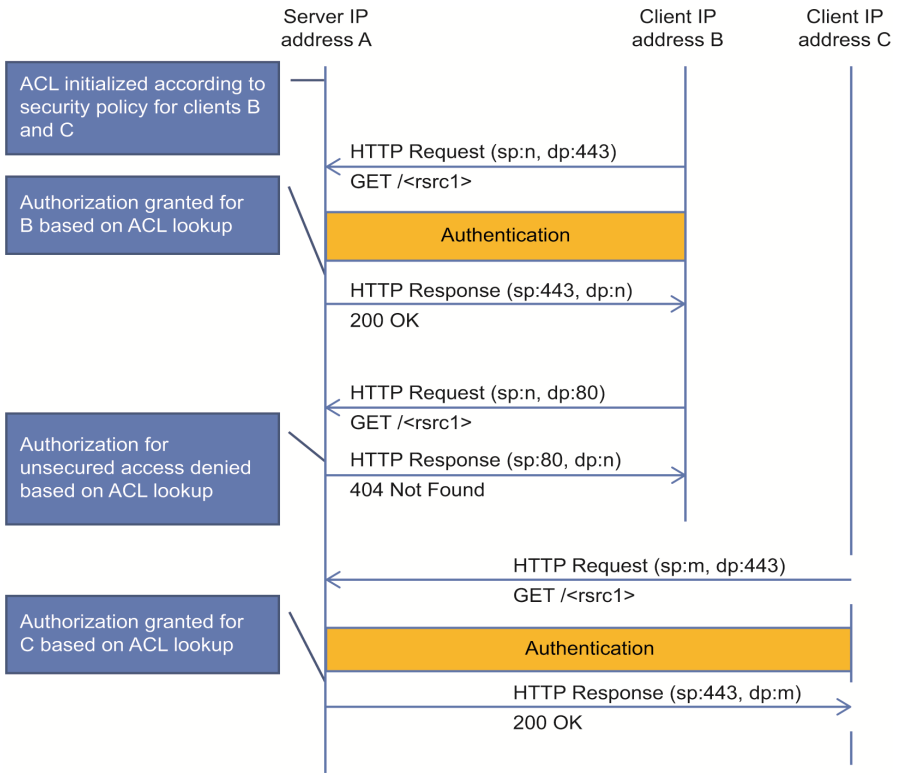


Figure 1—Example device authentication procedure using HTTPS port 443

Copyright © 2024 IEEE. All rights reserved.
Documents
Document Sharing Hub
Authorized licensed use limited to: Tsinghua University. Downloaded on February 25,2026 at 04:04:01 UTC from IEEE Xplore. Restrictions apply.
Full document details: IEEE 2030.5-2023