

IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol to Complement the Utility Industry End Device Data Tables

Sponsored by the
IEEE Standards Coordinating Committee 31 on
Automatic Meter Reading and Energy Management

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1703™-2012

29 June 2012

IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol to Complement the Utility Industry End Device Data Tables

Sponsor
**IEEE Standards Coordinating Committee 31 on
Automatic Meter Reading and Energy Management**

Approved 29 March 2012

IEEE-SA Standards Board

Abstract: A set of application layer messaging services are provided in this standard that are applicable for the enterprise and End Device ends of an Advanced Metering Infrastructure (AMI). The application services include those useful for managing the AMI network assets defined by this standard. These messages may be transported over a wide range of underlying network transports such as TCP/IP, UDP, IEEE 802.11, IEEE 802.15.4 IEEE 802.16, PLC, and SMS over GSM, over a wide range of physical media. Additionally, interfaces are defined for a Communication Module and a Local Port (e.g., an IEEE 1701 optical port).The described protocol is tailored for, but not limited to, the transport of IEEE 1377 Table data.Also, a means by which information can be sent in a secure manner using AES-128 and the EAX' mode is provided in this standard. This standard was developed jointly with ANSI (published as ANSI C12.22) and Measurement Canada (published as MC12.22).

Keywords: ACSE, ANSI C12.22, C12.19 Device, Communication Module, End Device, IEEE 1703

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

Copyright © 2012 by The Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 29 June 2012. 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 978-0-7381-7234-7 STD97221
Print: ISBN 978-0-7381-7248-4 STDPD97221

IEEE prohibits discrimination, harassment, and bullying. For more information, visit <http://www.ieee.org/web/aboutus/whatis/policies/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.

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. 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.

Use of an IEEE Standard is wholly voluntary. IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied "AS IS."

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. Every IEEE standard is subjected to review at least every ten years. When a document is more than ten 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.

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 his or her 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.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide 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 to ensure 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 Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at <http://standards.ieee.org/develop/wg/>.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854-4141
USA

Photocopies: Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Notice to users

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 imply 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.

Copyrights

This document is copyrighted by the IEEE. It is made available 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 this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE 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. 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 the IEEE-SA Website at <http://standards.ieee.org/index.html> or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit IEEE-SA Website at <http://standards.ieee.org/index.html>.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: <http://standards.ieee.org/findstds/errata/index.html>. Users are encouraged to check this URL for errata periodically.

Patents

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 <http://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 nondiscriminatory. 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.

Participants

At the time this standard was completed, the End Device/Telemetry Interface Unit Subcommittee Working Group had the following membership:

Edward Berozet, *Chair*
Richard D. Tucker, *Vice Chair*
Avygdor Moise, *Editor*

Michael Anderson
Martin Burns
Brent Cain
Lawrence Kotewa

Vuong Nguyen
James Pace
Terry Penn

Chris Schafer
Kendall Smith
Aaron Snyder
Virginia Zinkowski

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Ali Al Awazi
Chris Bagge
Hugh Barrass
Edward Berozet
Keith Chow
Ron Greenthaler
Randall Groves
David Haynes
Brad Johnson

Piotr Karocki
Lawrence Kotewa
Jim Kulchisky
William Lumpkins
Greg Luri
Faramarz Maghsoodlou
Gary Michel
Avygdor Moise
Jose Morales
Vuong Nguyen

Terry Penn
Bartien Sayogo
Gil Shultz
Aaron Snyder
Walter Struppler
Richard D. Tucker
Scott Weikel
Oren Yuen
Virginia Zinkowski

When the IEEE-SA Standards Board approved this standard on 29 March 2012, it had the following membership:

Richard H. Hulett, *Chair*
John Kulick, *Vice Chair*
Robert Grow, *Past Chair*
Judith Gorman, *Secretary*

Satish Aggarwal
Masayuki Ariyoshi
Peter Balma
William Bartley
Ted Burse
Clint Chaplin
Wael Diab
Jean-Philippe Faure

Alexander Gelman
Paul Houzé
Jim Hughes
Young Kyun Kim
Joseph L. Koepfinger*
David J. Law
Thomas Lee
Hung Ling

Oleg Logvinov
Ted Olsen
Gary Robinson
Jon Walter Rosdahl
Mike Seavey
Yatin Trivedi
Phil Winston
Yu Yuan

*Member Emeritus

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

Richard DeBlasio, *DOE Representative*
Michael Janezic, *NIST Representative*

Julie Alessi
IEEE Standards Program Manager, Document Development

Soo H. Kim
IEEE Client Services Manager, Professional Services

Introduction

This introduction is not part of IEEE Std 1703-2012, IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol to Complement the Utility Industry End Device Data Tables.

This standard is intended to accommodate the network messaging requirements of an advanced metering infrastructure such as that identified by the Office of Electricity Delivery and Energy Reliability of the U.S. Department of Energy; the Smart Metering Initiative of the Ontario Ministry of Energy (Canada) and the stated requirements of Measurement Canada for the approval of a metering device for use in Canada.

This standard describes four different but related communication modes of operation. One is the operation of an End Device (node) over any network, a feature that all IEEE 1703 compliant nodes need to implement. The second is an exposed point-to-point interface between an IEEE 1703 Device, e.g., a meter, and an IEEE 1703 Communication Module, e.g., a network adaptor. The third is the capture, translation and transmission of one-way messages (blurts). The fourth is communication with the End Device over a dedicated IEEE 1701 ANSI Type 2 optical local port.

This standard assigns roles to all Utility AMI network assets to enable the automated deployment and configuration of network nodes in a distributed AMI enterprise system. The roles provide for Relays, Master Relays, and Gateways; simple Hosts, Authentication Hosts, and Notification Hosts; and sensory End Devices. These devices work together to realize a Utility enterprise network in a manner that provides for a universal application framework that can operate any compliant appliance so that it may be deployed, accessed and communicated with seamlessly over any network infrastructure, hardware and available bandwidth. This is accomplished through the provision of well-defined network management services (e.g., trace, resolve, register, and de-register), data access service (e.g., read and write), session management (e.g., logon, logoff, terminate, and disconnect), message segmentation and assembly, message playback rejection, security and privacy.

The protocol is well suited for two-way and one-way communication using an extremely wide network address space (using ApTitles). It implements subscription and node discovery services so that any Utility enterprise (e.g., MDMS, DA, DR, or any other willing network appliance, such as a home energy monitoring system or a thermostat) can register themselves as Notification Hosts so that they can receive advisory messages and alerts about network asset changes or changes in the state of the network. Network subscription services may be managed through distributed Authentication Hosts.

Altogether, this standard was designed to be simple and small. Therefore it can meet the requirements of the smallest of AMI networks, while providing well-defined capacities that can grow and adapt to the largest of enterprise AMI networks and operations as needed. As such, this standard together with IEEE Std 1377™-2012 addresses the end-to-end distributed AMI network needs from the smallest to the largest of enterprise AMI systems.

The body of this standard was developed jointly with ANSI C12.22 and MC12.22. The joint agreement calls for the standards and regulatory organizations IEEE, ANSI and MC to maintain the body of this standard in step as they publish versions and revisions of the standard. A number of editorial errors and errors of omission were discovered after the publication of ANSI C12.22-2008, just before the acceptance of IEEE Std 1703-2012. These errors are listed in Annex K – Listing of Editorial Errors and Errors of Omission in ANSI C12.22-2008. All reported errors were identified in the body of this standard and highlighted to indicate that the text was in error and was corrected. In addition, errors of omission were corrected in the body of this document and similarly highlighted.

Contents

1	Overview	1
1.1	Introduction	1
1.2	Scope	1
1.3	Purpose	3
2	Normative references	4
3	Definitions and syntax	6
3.1	Definitions	6
3.2	Document syntax	10
3.3	Table syntax	10
4	Reference topology	11
5	C12.22 Node to C12.22 Network Segment details	13
5.1	C12.22 Node to C12.22 Network Segment reference	13
5.2	Data encoding rules	14
5.3	Layer 7—application layer	16
5.4	Layer 6—presentation layer	79
5.5	Layer 5—session layer	79
5.6	Layer 4—transport layer	79
5.7	Layer 3—network layer	79
5.8	Layer 2—data link layer	79
5.9	Layer 1—physical layer	79
6	Protocol Details: C12.22 Device to C12.22 Communication Module interface	80
6.1	Interface architecture	80
6.2	Interface diagram	80
6.3	Implementation guidelines	81
6.4	Layer 7—application layer	83
6.5	Layer 6—presentation layer	83
6.6	Layer 5—session layer	83
6.7	Layer 4—transport layer	83
6.8	Layer 3—network layer	100
6.9	Layer 2—data link layer	100
6.10	Layer 1—physical layer	108
7	Local Port communication protocol details	111
7.1	Protocol definition	111
7.2	C12.22 Local Port communication using a C12.18 Optical Port	112
8	Backward compatibility	114
9	Compliance	115
	Annex A (normative) Relays	116
	Annex B (informative) Routing examples	120
	Annex C (normative) Modifications and extensions to ANSI C12.19	122
	Annex D (normative) Universal Identifier	170
	Annex E (normative) One-way devices	172

Annex F (informative) APDU Response Timeout algorithm	174
Annex G (informative) Communication examples	176
Annex H (informative) CRC examples	189
Annex I (normative) The EAX' Cryptographic Mode	191
Annex J (informative) Connectionless-ACSE-1 equivalent reduced syntax for C12.22 Message transmission	208
Annex K (informative) Bibliography	210
Annex L (informative) Listing of editorial errors and errors of omission in ANSI C12.22-2008	211

IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol to Complement the Utility Industry End Device Data Tables

IMPORTANT NOTICE: IEEE Standards documents are not intended to ensure safety, health, or environmental protection, or ensure against interference with or from other devices or networks. Implementers of IEEE Standards documents are responsible for determining and complying with all appropriate safety, security, environmental, health, and interference protection practices and all applicable laws and regulations.

This IEEE document is made available for use subject to important notices and legal disclaimers. These notices and disclaimers appear in all publications containing this document and may be found under the heading “Important Notice” or “Important Notices and Disclaimers Concerning IEEE Documents.” They can also be obtained on request from IEEE or viewed at <http://standards.ieee.org/IPR/disclaimers.html>.

1 Overview

1.1 Introduction

This standard *defines* network Application Services for the exchange of Table data and control elements. These services need to be implemented by all C12.22 Nodes, including “back-office” or “Head-end” systems.

1.2 Scope

Initially, communications with electronic devices consisted of transporting memory data via proprietary protocols that were unique to each manufacturer. The desire for interoperability and support for multiple manufacturers by reading and programming systems created a need for standardization of data formats and transport protocols.

The first step was to standardize data formats. Internal data was abstracted as a set of Tables. A set of standard Table contents and formats were defined in [ANSI C12.19/MC12.19/IEEE 1377](#), “Utility Industry End Device Data Tables.”¹

¹ Information on references can be found in Clause 2.

In the “Protocol Specification for ANSI Type 2 Optical Port” Standard ([ANSI C12.18/MC12.18/IEEE 1701](#)), a point-to-point protocol was developed to transport table data over an optical connection. The [ANSI C12.18 / MC12.18/IEEE 1701](#) protocol include an application language called Protocol Specification for Electric Metering (PSEM) that allows applications to read and write Tables. The “Protocol Specification for Telephone Modem Communication” ([ANSI C12.21/MC12.18/IEEE 1702](#)) was then developed to allow devices to use PSEM to transport Tables over telephone modems.

This standard extends the concepts of ANSI C12.18/MC12.18/IEEE 1701, ANSI C12.21/MC12.18/IEEE 1702, and [ANSI C12.19/MC12.19/ IEEE 1377](#) standards to allow transport of Table data over any reliable networking communications system. Note that in this use of the word, “reliable” means that for every message sent, the sender receives a response at its option: either a positive acknowledgment or an error message. That is, messages cannot fail silently in a reliable network (see discussion of Reliable Stream Transport Service in [IPPA](#) [B1]).²

In addition, this standard describes an optionally exposed point-to-point interface between a C12.22 Device and a C12.22 Communications Module designed to attach to “any” network. The terms “C12.22 XXXX” (e.g., C12.22 Device) were introduced by ANSI C12.22-2008. These terms can be interchangeably replaced with the terms “IEEE 1703 XXXX”; i.e., the IEEE 1703 Device is the same as the ANSI C12.22 Device and the IEEE 1703 Communication Module is the same as the C12.22 Communication Module. However, since this standard was originally developed under the auspice of ANSI C12 SC17 WG1, the document terminology is based on C12.22 terms.

Furthermore, this standard defines a methodology to capture, translate, and transmit one-way device messages (blurts).

This standard defines interfaces between IEEE 1377 Devices (ANSI C12.19 Devices) and network protocols.

Specific goals identified by the committee in the creation of this standard were:

- a) Defining a Datagram that may convey ANSI C12.19 data Tables through any network

This was accomplished by:

- Assuming that the data source is ANSI C12.19 data Tables
- Defining the Application Layer services (language)

- b) Providing a full stack [\[ISO/IEC 7498-1\]](#) definition for interfacing a C12.22 Device to a C12.22 Communication Module

This was accomplished by:

- Defining the physical interface requirements between the C12.22 Device and the C12.22 Communication Module
- Defining the interface lower layers [ISO/IEC 7498-1]: 4 (transport), 3 (network), 2 (data link), and 1 (physical)

- c) Providing a full stack definition for point-to-point communication to be used over local ports such as optical ports or modems

This was accomplished by defining a Layer 4 (transport) and Layer 2 (data link)

- d) Providing support for efficient one-way messaging (blurts)

² The numbers in brackets correspond to those of the bibliography in Annex K.

This was accomplished by:

- Defining a compact message format that can be easily transformed into a standard ANSI C12.22 Datagram
 - Assuring that all needed layers defined in this standard can support one-way messaging
- e) Providing network architecture compatible with this protocol (some architectural concepts were derived from HCCS 1 [B5], HCCS 2 [B6], HCCS 3 [B7], DND [B4], IPPA [B1], and TCPCE [B2])

This was accomplished by:

- Defining different types of nodes such as C12.22 Relay, C12.22 Master Relay, C12.22 Host, C12.22 Authentication Host, C12.22 Notification Host, and C12.22 Gateway
 - Defining the roles and responsibilities of each of these C12.22 Nodes
- f) Providing data structure definitions in support of this protocol

This was accomplished by:

- Defining an ANSI C12.19 Decade to be used by C12.22 Nodes
- Defining an ANSI C12.19 Decade to be used by C12.22 Relays
- Defining new procedures in support of this protocol
- Defining a new Table for enhanced security

1.3 Purpose

The Utility Industry has a need for a standard that provides an operable “plug-and-play” environment for field devices (e.g., meters, communication modules, and Utility systems). The purpose of this standard is to define the network framework and means to transport the Utility End Device Data Tables via any Local area / Wide area network for use by enterprise systems in a multi-source environment.

This standard is intended to accommodate the concept of an advanced metering infrastructure such as that identified by the Office of Electricity Delivery and Energy Reliability of the U.S. Department of Energy; the Smart Metering Initiative of the Ontario Ministry of Energy (Canada); and the stated requirements of Measurement Canada for the approval of a metering device for use in Canada.

This standard is to provide a uniform, managed, adaptive, and secured network data and message delivery system for Utility End Devices and ancillary devices (e.g., home appliances and communication technology) that can operate in a “plug-and-play” and “end-to-end” multi-source enterprise AMI environment, in a manner that allows independence from the underlying network implementation (i.e., an End Device can implement this standard by utilizing a transceiver that is independent of the meter’s metrology logic and for the meter not to depend on the design of the network that is serviced by that transceiver). The independence from the underlying native network protects the End Device from premature obsolescence that may occur as networks may come and go.

This standard extends the definitions provided by IEEE Std 1377-2012 to include provisions for enterprise-level asset management, data management, and uniform data exchange interfaces, through the use of network and relay tables and services. In addition, it is to provide all the necessary support services needed to deploy, commission, notify, manage, and access End Devices in a manner that preserves the privacy, security, and integrity of the network.

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.

ANSI C12.1-2001, American National Standard for Electric Meters Code for Electricity Metering.³

ANSI C12.18, American National Standard for Protocol Specification for ANSI Type 2 Optical Port.

ANSI C12.19, American National Standard for Utility Industry End Device Data Tables.

ANSI C12.19-1997, American National Standard for Utility Industry End Device Data Tables.

ANSI C12.19-2008, American National Standard for Utility Industry End Device Data Tables.

ANSI C12.21, American National Standard for Protocol Specification for Telephone Modem Communication.

ANSI INCITS 92, Data Encryption Algorithm (R2003).

ATIS T1.667, Intelligent Network.⁴

EAX, Authenticated Encryption with Associated Data (AEAD) Algorithm Designed to Simultaneously Protect both Authentication and Privacy of Messages, as Described in “A Conventional Authenticated-Encryption Mode,” M. Bellare, P. Rogaway and D. Wagner, April 13, 2003.⁵

EAX MO, The EAX Mode of Operation, A Two-Pass Authenticated-Encryption Scheme Optimized for Simplicity and Efficiency, M. Bellare, P. Rogaway, and D. Wagner, January 18 2004.⁶

FIPS Pub 197, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/N.I.S.T, Springfield, Virginia, November 26, 2001.⁷

IEEE Std C37.90.1™, IEEE Standard for Surge Withstand Capability (SWC) Tests for Relays and Relay Systems Associated with Electric Power Apparatus.^{8, 9}

IEEE Std C62.41™, IEEE Recommended Practice on Surge Voltages in Low-voltage AC Power Circuits.

ISO/IEC 646, ASCII character set.¹⁰

ISO/IEC 7498-1, Information Technology—Open Systems Interconnection—Basic Reference Model: The Basic Model.

ISO/IEC 8824-1, Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation.

³ ANSI publications are available from the American National Standards Institute (<http://www.ansi.org/>).

⁴ ATIS publications are available from the Alliance for Telecommunications Industry Solutions (<http://www.atis.org/>).

⁵ EAX publications are available from the National Technical Information Service (<http://www.ntis.gov/>). This document is available from <http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/eax/eax-spec.pdf> and described in [EAX MO 2004].

⁶ This document is available from <http://www.cs.ucdavis.edu/~rogaway/papers/eax.pdf>.

⁷ FIPS publications are available from the National Technical Information Service (<http://www.ntis.gov/>).

⁸ IEEE publications are available from The Institute of Electrical and Electronics Engineers (<http://standards.ieee.org/>).

⁹ The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

¹⁰ ISO/IEC publications are available from the ISO Central Secretariat (<http://www.iso.org/>). ISO publications are also available in the United States from the American National Standards Institute (<http://www.ansi.org/>).

IEEE Std 1703-2012
IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol
to Complement the Utility Industry End Device Data Tables

ISO/IEC 8824-2, Information Technology—Abstract Syntax Notation One (ASN.1): Information Object Specification.

ISO/IEC 8824-3, Information Technology—Abstract Syntax Notation One (ASN.1): Constraint Specification.

ISO/IEC 8824-4, Information Technology—Abstract Syntax Notation One (ASN.1): Parameterization of ASN.1 Specifications.

ISO/IEC 8825-1, Information Technology—ASN.1 Encoding Rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER).

ISO/IEC 10035-1, Information Technology—Open Systems Interconnection—Connectionless Protocol for the Association Control Service Element: Protocol Specification.

ISO/IEC 13239, Information Technology—Telecommunications and Information Exchange between Systems—High-level Data Link Control (HDLC) Procedures—Frame Structure, Annex A, Explanatory Notes On Implementation of the Frame Checking Sequence.

ISO/IEC 15954, Information Technology—Open Systems Interconnection—Connection-mode Protocol for the Application Service Object Association Control Service Element.

ISO/IEC 15955, Information Technology—Open Systems Interconnection—Connectionless Protocol for the Application Service Object Association Control Service.

NIST SP 800-38A, Recommendation for Block Cipher Modes of Operation - Methods and Techniques. NIST Special Publication 800-38A. US Department of Commerce/N.I.S.T, Springfield, Virginia.¹¹

NIST SP 800-38B, Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. NIST Special Publication 800-38B. US Department of Commerce/N.I.S.T, Springfield, Virginia.¹²

¹¹ NIST publications are available from the National Institute of Standards and Technology (<http://www.nist.gov/>). This document is available from http://csrc.nist.gov/publications/nistpubs/800-38A/sp_800-38A.pdf.

¹² This document is available from http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf.

3 Definitions and syntax

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. The *IEEE Standards Dictionary Online* [B8] should be consulted for terms not defined in this clause.¹³

Absolute Universal Identifier (UID): Absolute encoding of a UID according to the “Encoding of an object identifier value” clause in ISO/IEC 10035-1.

Association Control Service Element (ACSE): Association Control Service Element encoded per “Connectionless protocol for the association control service element: Protocol specification” in ISO/IEC 10035-1. Connectionless ACSE defines the Datagram encapsulation protocol used in this standard.

Application Association: *See: Association.*

Application Context: A set of service elements and supporting information used on the Application Association. It includes a description of the relationships and dependencies of the C12.22 Application service elements and a description of the logical structure of information to be exchanged between cooperating Application Entities.

Application Entity: The system-independent application activities that are made available as application services to the application agent; e.g., a set of application service elements that together perform all or part of the communication aspects of an application process (see ATIS T1.667).

Application Process: *See: Application Entity.*

Application Protocol Data Unit (APDU): A Datagram that is transferred error-free between network nodes. The C12.22 standard encodes APDUs using Association Control Service Element (ACSE) to carry Extended Protocol Specification for Electric Metering (EPSEM) services and C12.19 payloads between C12.22 Nodes.

Application Protocol Data Unit (APDU) Segment: An APDU that is constructed using C12.22 Segmentation as the process of breaking a C12.22 Datagram into smaller units before transmission. *See also: C12.22 Datagram segmentation and reassembly.*

ApTitle: In addition to the addressing constructs (transport address and possibly session and presentation selectors), the communicating application entities have names—application-entity titles (AeTitle). These are carried by Association Control Service Element (ACSE) as two fields—the Application-process titles (ApTitle) and the application-entity qualifier (AeQualifier).

C12.22 ApTitles may be encoded absolutely or relatively. Relative Universal Identifiers (UIDs) shall be unique only within the context of a C12.22 Network or inside C12.22 Nodes. C12.22 Relays, C12.22 Communication Modules, or C12.22 Transport Layers shall map Relative UIDs to other Relative UIDs or Absolute UIDs to ensure the uniqueness of the ApTitle within the context of any network, a C12.22 Network, or a C12.22 Network Segment as needed.

Association: A cooperative relationship among peer (utilizing the same protocol) Application Entities that enables the communication of information and the coordination of their joint operation for an instance of communication. This relationship may be formed by the transfer of application protocol control information during the establishment of a connection, or transitionally, during a single invocation through a connectionless service. Associations can also be predefined and longstanding.

¹³ The *IEEE Standards Dictionary Online* subscription is available at http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

Basic Encoding Rules (BER): As defined by ISO/IEC 8825-1, describing methods used to identify and encode data elements for transport.

Bit: A Binary Digit. It is the unit of information of a computational quantity that can take on one of two values, such as false and true or 0 and 1. A bit is said to be “set” if its value is true or 1, and “reset” or “clear” if its value is false or 0. One speaks of setting and clearing bits. To toggle or “invert” a bit is to change it, either from 0 to 1 or from 1 to 0.

Byte: A group of 8 bits of data. When expressed in this standard, bit 0 is the least significant bit and it is written at the right-most position of a bit sequence. Bit 7 is the most significant bit and it is written at the left-most position of the bit sequence. The actual bit-signal transmission sequence and bit polarity, which represents ones or zeroes, is determined by the characteristics of the appropriate OSI Physical Layer used to transmit the byte.

C12.19 Device: A C12.22 Node that contains Tables.

C12.19 Device Class: A Relative Universal Identifier (UID) that uniquely identifies a C12.19 Device Table set per the MANUFACTURER field defined in Table 0 of ANSI C12.19-1997 or the DEVICE_CLASS field defined in ANSI C12.19-2008.

C12.22 Application: An Application Entity that implements a set of services and procedures as defined in this standard permitting one or more well-defined devices (C12.22 Host, C12.22 Relay, C12.22 Device, C12.22 Communication Module, etc.) to interact within the framework of a C12.22 Network. It may also contain C12.19 Tables.

C12.22 Authentication Host: A C12.22 Host that is an authoritative administrative host for a registering C12.22 Node in the C12.22 Master Relay domain. The C12.22 Authentication Host may be embedded inside a C12.22 Master Relay or it may be a separate C12.22 Node on the network. There may be one or more C12.22 Authentication Hosts operating under the domain of a single C12.22 Master Relay. Registration with C12.22 Master Relays can only succeed if at least one C12.22 Authentication Host accepts registration on behalf of a C12.22 Node by a C12.22 Master Relay.

C12.22 Client: A C12.22 Node that initiates a Logon service request for the purpose of establishing a session with a C12.22 Server.

C12.22 Communication Module: A hardware module that attaches a C12.22 Device to a C12.22 Network Segment. A C12.22 Communication Module can be physically located inside or outside the C12.22 Device enclosure. However, it is physically and logically distinct from the C12.22 Device. The interface between the C12.22 Communication Module and the C12.22 Device is completely defined by this standard. The combination of a C12.22 Device and a C12.22 Communication module is a C12.22 Node. If a C12.22 Communication Module contains Tables, it is also a C12.22 Node.

C12.22 Datagram segmentation and reassembly: The process of breaking a C12.22 Datagram into smaller units before transmission and then reassembling it into the proper order at the receiving C12.22 Node. C12.22 Datagrams are made smaller specifically because of specified packet size restrictions in a given path across a channel. The transport protocol determines the size of the smallest maximum Protocol Data Unit (PDU) supported by the underlying C12.22 Network Segment for the purpose of transmission to the target C12.22 Node.

C12.22 Device: A module that hosts C12.22 Application(s) and provides at least one Interface to a C12.22 Communication Module.

C12.22 Gateway: A C12.22 Node that translates the ANSI C12.22 protocol to/from other protocols. Gateways are required when a C12.22 Node needs to communicate with non-C12.22 Nodes. C12.22 Gateways can be attached directly to the non-C12.22 Devices or they can provide their translation services through any network segment.

C12.22 Host: A C12.22 Node that may be a C12.22 Authentication Host or C12.22 Notification Host or both. A Host typically runs on a computer instead of within an embedded system.

C12.22 Master Relay: A C12.22 Relay that operates at the top of a hierarchy of relays. It provides registration services of all devices in its domain. It is also responsible for issuing registration service queries to C12.22 Authentication Hosts and Deregistration service requests and notifications to C12.22 Notification Hosts when registering a C12.22 Node. A C12.22 Master Relay can also act as a C12.22 Host.

C12.22 Message: Any notice, service request, service response, or device status sent from one C12.22 Node to another C12.22 Node for the purpose of communication across a C12.22 Network. The detailed encoding of C12.22 Messages is defined by the appropriate encoding rules of the OSI Layer from which they are issued.

C12.22 Network: A C12.22 communication infrastructure that is composed of one or more C12.22 Network Segments. A C12.22 Network shall include at least one C12.22 Master Relay.

C12.22 Network Segment: A collection of C12.22 Nodes that can communicate with each other without forwarding messages through a C12.22 Relay. C12.22 Network Segments are interconnected using C12.22 Relays.

C12.22 Node: A point on the network that attaches to a C12.22 Network Segment. C12.22 Nodes contain one or more C12.22 Applications. Each C12.22 Node shall have a unique ApTitle on a C12.22 Network.

C12.22 Notification Host: A C12.22 Host that contains an application that needs to be notified when C12.22 Nodes are registered for the first time (“first” here means an actual registration request as contrasted with the reuse of the register service as keep-alive) or deregistered. Each C12.22 Notification Host may add the registered C12.22 Node to its active client list for subsequent processing by the C12.22 Host application.

C12.22 Relay: A C12.22 Node that provides address resolution, Datagram segmentation, and optionally message forwarding services to other C12.22 Nodes. Address resolution services consist of mapping Layer 7 addresses (ApTitle) to lower layer addresses (Network Entity Title).

C12.22 Server: A C12.22 Node that is a recipient of a Logon service request from a C12.22 Client for the purpose of establishing a session with that client.

Called ApTitle: The Called ApTitle is encoded as an Absolute Universal Identifier (UID) or Relative UID. It uniquely identifies the target of an ACSE C12.22 Message.

Calling ApTitle: The Calling ApTitle is encoded as an Absolute Universal Identifier (UID) or Relative UID. It uniquely identifies the source of an ACSE C12.22 Message.

Channel: A single path for transmitting signals, usually in distinction from other parallel paths. Multiple channels may coexist on the same physical media. The term “channel” may signify either a one-way path (providing transmission in one direction only) or a two-way path (providing transmission in two directions).

Cipher: An algorithm for performing encryption.

Cipher, Inverse: An algorithm for performing decryption.

Ciphertext: The data output from the Cipher or input to the Inverse Cipher.

Cleartext: Data sent without transformation even when privacy is being used.

Connection: A logical and physical binding between two or more users of a service.

Datagram: A self-contained, independent entity of application data carrying sufficient information to be routed from the source Application Layer to the destination Application Layer. This standard encapsulates each Datagram as one or more ACSE Protocol Data Units (PDUs). For more discussion on Datagrams, see pp. 3, 8, 9, 14, 15, and 26–52 of **HCCS 3** [B7].

Extended PSEM (EPSEM): Extended structures and services enabling transportation of multiple requests and responses at the same time for use by devices such as gas, water, electricity, and related electronic modules. There are also provisions for response control and C12.19 Device Class identification. EPSEM messages are encapsulated within ACSE PDUs.

Fragment: *See:* **APDU Segment**.

Interface: The C12.22 Device hardware components used to manifest a C12.22 Node on a C12.22 Network Segment.

Local Port: A physical interface that is directly attached to the C12.22 Node or a physical interface that is located in the immediate vicinity of the C12.22 Node and attached to it by means of a dedicated short signal path (e.g., cable). The main purpose of the Local Port is to provide direct access to the application process of the C12.22 Node. The C12.22 Node application process may redirect C12.22 Messages that originate from a Local Port to other Local Ports or other C12.22 Node interfaces. Similarly, the C12.22 Node application process may redirect incoming C12.22 Messages to Local Ports. The C12.22 Communication Module interface of a C12.22 Device is not a Local Port. All Local Ports of a C12.22 Node shall access to the same C12.22 Application. The physical Local Port characteristics (OSI Layer 1) are not defined by this standard; however, the Data Link through Application Layers (Layers 2-7) are fully defined by this standard.

Octet: *See:* **Byte**.

Other Device: A device that does not implement the ANSI C12.22 protocol.

Plaintext: A data input to the Cipher or output from the Inverse Cipher. This should not be confused with Cleartext, to which a Cipher is not applied.

PSEM: *See* ANSI C12.18.

Relative UID: Relative encoding of a UID according to “Encoding of a relative object identifier value” in ISO/IEC 10035-1. The Relative UID is a subset of an Absolute UID; e.g., the absolute object identifier 2.100.3.8571.3.2 contains the relative object identifiers 8571.3.2.

Segment: In the context of a C12.22 Network, a C12.22 Network Segment. In the context of a C12.22 APDU, an APDU Segment.

Segmentation: *See:* **C12.22 Datagram segmentation and reassembly**.

Session: An association context that is maintained while the two C12.22 Nodes are communicating back and forth in a conversation of some duration. Managing a session includes setting up and taking down the connection between two communicating C12.22 Nodes. Some session associations last only long enough to send a message in one direction. However, other session associations may last longer, usually with one or both of the communicating parties able to terminate it.

Transaction: A unit of interaction that occurs individually and coherently.

Universal Identifier (UID): Universally unique identifiers that are encoded using BER. A UID may be formulated as an Absolute UID or as a Relative UID and can be used to specify C12.22 object identifiers such as Calling ApTitle, Called ApTitle.

3.2 Document syntax

Document syntax is identical to that described in ANSI C12.18.

Describing data definitions is usually accomplished within the confines of a given language and the grammar rules of that language. Since the data definitions embodied within this document are meant to be independent of specific language and, hopefully, capable of being implemented within the confines of any language, a method for describing the data definitions must be developed. A modified form of the Backus-Naur Form (BNF) serves as the basis for building the descriptions used to construct the data definitions.

The modified form of BNF has the following definitions:

Symbol Meaning:

- < > A string contained inside angle brackets is called a nonterminal. That is, while it may be viewed as a single unit, it can and should be redefined as consisting of one or more simpler elements.
- ::= This symbol is read as “is defined as”. The nonterminal that occurs on the left-hand side (LHS) of this symbol consists of the elements (nonterminals, terminals, or a combination of the two) found on the right-hand side (RHS). A line containing a left-hand side, ::, and a right-hand side is known as a production rule.
- | The vertical bar is an “OR” symbol. The OR symbol always occurs on the RHS of a production where the LHS can be defined in more than one way. The OR bar separates valid alternative right-hand sides.
- [] A symbol enclosed in square brackets is optional. The production is valid whether or not it is included.
- * The superscript asterisk is known as the Kleene star. A symbol followed by the Kleene star may occur zero or more times without violating the grammar.
- + The superscript plus sign is known as the Kleene cross. A symbol followed by the Kleene cross must occur one or more times.
- +n A symbol followed by the Kleene cross and any superscript number “n” represents “n” occurrences of the symbol.
- { } The curly braces are used to enclose comments within the descriptions. Comments have no impact on the productions.

3.3 Table syntax

Table document form syntax is identical to that described in ANSI C12.19-2008.

4 Reference topology

This standard defines components of a C12.22 Network. Each component is defined with enough flexibility to allow multiple components to be incorporated into one physical device. Figure 4-1 describes the C12.22 Network and protocol topology within which C12.22 Nodes are expected to operate.

This network topology accommodates interconnections among C12.22 Nodes that can be located on the same or on different networks. C12.22 Messages are forwarded between C12.22 Network Segments using C12.22 Relays.

The network topology also accommodates C12.22 Gateways that translate the C12.22 protocol to other protocols. C12.22 Devices and non-C12.22 Devices may be collocated on the same C12.22 Network Segment and optionally provide C12.22 Gateway functionality. The blocks labeled Other Device on the topology diagram are logical constructs and may physically include one or more devices (e.g., a non-C12.22 network of devices).

The interface between a C12.22 Device and C12.22 Communication Module(s) is fully defined in this standard. However, the Standard general definition of the interface of a C12.22 Node to a C12.22 Network Segment is limited to the Application, Presentation, and Session Layers only (Layers 7-5). The interface between a C12.22 Device and a C12.22 Communication Module (Layers 1–4) is shown in Figure 4-1 using triple lines.

In the case where a C12.22 Node is connected to more than one C12.22 Network Segment, communication through those segments shall be to the same C12.22 Application(s). Access to the C12.22 Node through a Local Port shall also be to the same C12.22 Application(s).

When a Local Port is attached to a C12.22 Device, this port provides access to the C12.22 Application(s) of this C12.22 Device and may optionally provide access to the attached C12.22 Communication Modules. Similarly, when a Local Port is attached to a C12.22 Communication Module, this port provides access to the C12.22 Application(s) of this C12.22 Communication Module and may optionally provide access to the attached C12.22 Device.

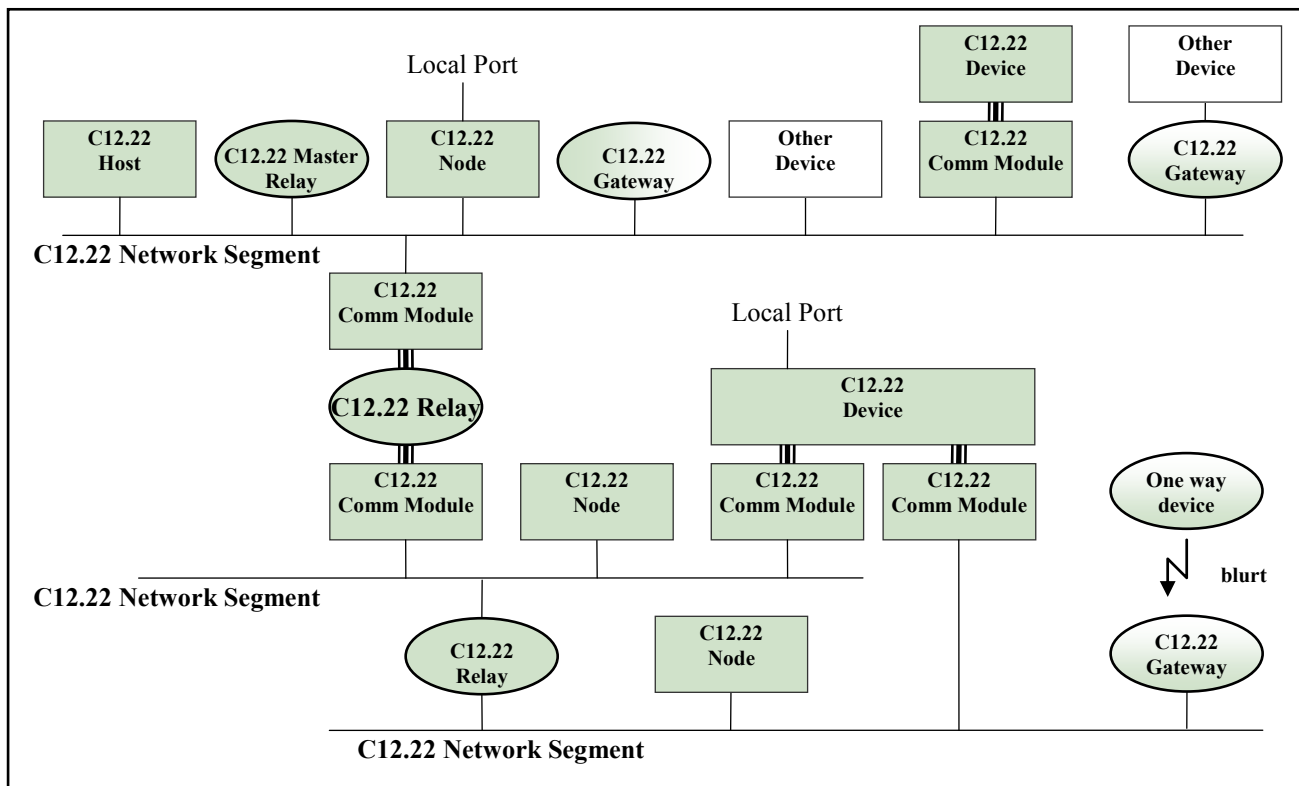


Figure 4-1—Reference topology

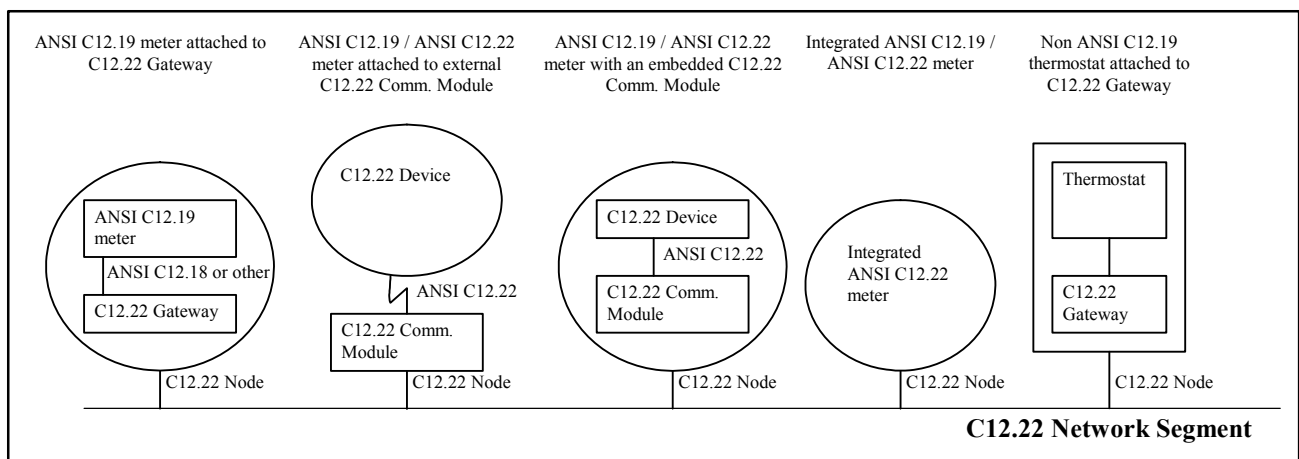
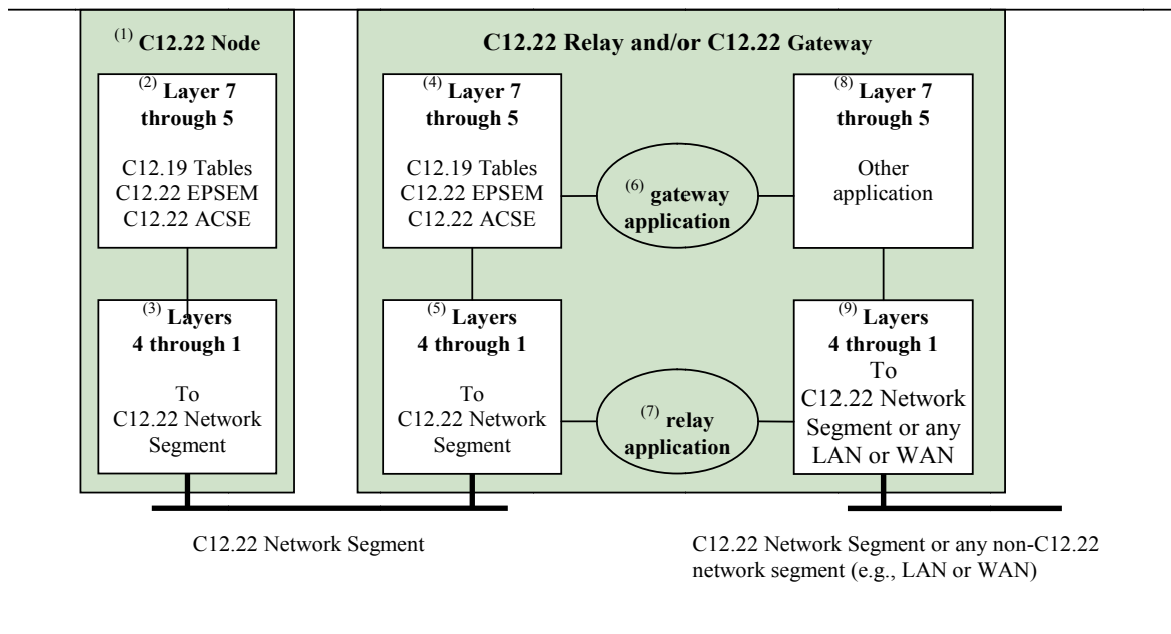


Figure 4-2—C12.22 Node implementation examples

5 C12.22 Node to C12.22 Network Segment details

5.1 C12.22 Node to C12.22 Network Segment reference

Figure 5-1 shows a C12.22 Node that is attached to a C12.22 Network Segment. It also shows the relay and gateway interface requirements needed to access remote networks (C12.22 Relay) and the translation services that may exist in these networks (C12.22 Gateway). The figure also shows how a C12.22 Network Segment can be connected to a non-C12.22 network segment.



Annotations:

- 1) C12.22 Node attached to a C12.22 Network Segment
- 2) C12.22 Application communicating using C12.19 Tables and EPSEM/ACSE encapsulation
- 3) Layers 4 through 1 interfacing a C12.22 Application to an unspecified native network infrastructure
- 4) C12.22 Application of a C12.22 Gateway
- 5) Layers 4 through 1 of a C12.22 Relay
- 6) C12.22 Gateway performing translation of the C12.22 Application to and from another application
- 7) C12.22 Relay performing layer 4 through 1 translation
- 8) Other application of a C12.22 Gateway
- 9) Remote (other) network interface side of a C12.22 Gateway and/or C12.22 Relay

Figure 5-1—C12.22 reference network model

5.2 Data encoding rules

5.2.1 Data order

The data order is identical to that defined in ANSI C12.18 and ANSI C12.19.

Within the syntax definitions, multiple parameters shall be encoded in the order as shown, from left to right.

Unless otherwise noted, parameters in all layers within the protocol definition are encoded most significant byte first. The order of data fields within Tables is dictated by ANSI C12.19.

```
<word24>      ::= <msbyte> <byte> <lsbyte>
<word16>      ::= <msbyte> <lsbyte>

<msbyte>      ::= <byte> {most significant byte}
<lsbyte>      ::= <byte> {least significant byte}

<byte>        ::= {See definition of Byte.}
```

5.2.2 Length fields encoding

ASN.1/BER field lengths are encoded using [ISO/IEC 8825-1](#). This encoding is defined as follows:

For values between 0 and 127, length fields are encoded using the short form. This form consists of a single byte representing the length of the subsequent message in octets.

For values greater than 127, length fields are encoded using the long form. This form consists of one byte with bit 7 set to one and bits 0 to 6 set to the number of additional octets, which follow. Additional octets represent the length encoded with most significant octet appearing first.

This is a restricted version of the BER. The BER length is restricted by Distinguished Encoding Rules (DER) encoding; specifically, it shall be encoded as the definite length on the minimum number of octets whether the encoding is in primitive or in constructed form.

Examples:

- Short form: 2C_H represents a length of 2C_H or 44
- Long form: 82_H 01_H 26_H represents a length 0126_H or 294
- Long form: 81_H 80_H represents a length of 80_H or 128

5.2.3 Universal identifiers encoding

This standard uses Universal Identifiers contained in the <aSO-context>, <mechanism-name>, the <calling-AP-title>, <called-AP-title>, Network and Relay Tables, EPSEM application layer services, and C12.22 Communication Module transport layer services. A Universal Identifier, <universal-id>, is an ordered list of sub-identifier values that are concatenated together and represented as follows:

value₁ .value₂. value_n. value_m

To guarantee the uniqueness of this identifier over all possible applications, the leading (left most) n values are obtained from an official registration body like the International Organization for Standardization (ISO). Any values following (branches of) this unique prefix (root) can be assigned locally by the owner of this prefix.

Universal identifier is encoded using the Basic Encoding Rules (BER) (ISO/IEC 8825-1) object identifier content encoding. This encoding is defined as follow:

- The first octet has value $40 \times \text{value}_1 + \text{value}_2$. (This is unambiguous because value_1 is limited to values 0, 1, and 2; value_2 is limited to the range 0 to 39.)
- The following octets, if any, encode $\text{value}_3, \dots, \text{value}_n$. Each value is encoded base 128, most significant digit first, with as few digits as possible, and the most significant bit of each octet except the last in the value's encoding set to one.

For efficiency, it is possible to encode a Universal Identifier relative to an ANSI C12 root. In that case, only the branch of the designated ANSI C12 root is included in the Relative Identifier.

The ASN.1 (ISO/IEC 8824-1, ISO/IEC 8824-2, ISO/IEC 8824-3, ISO/IEC 8824-4, ISO/IEC 8825-1) assigned tag for a Universal Identifier (OBJECT IDENTIFIER) is 06_H . The ASN.1 assigned tag for a Universal Relative Identifier (RELATIVE-UID) is $0D_H$. These tags (06_H and $0D_H$) may be used as described below only when placing object identifiers in C12.19 Tables, in EPSEM application layer services, and in C12.22 Communication Module transport layer services or used explicitly in ASN.1 syntax:

```
<universal-id-element> ::= 06H <universal-id-length> <universal-id>
                               {Absolute encoding of a universal
                               identifier, encoded as described in
                               ISO/IEC 8825-1 [BER].}

<universal-id-length> ::= <byte> {Length of <universal-id>. This value
                               shall range between 00H and 7FH.}

<universal-id> ::= <byte>*      {Absolute object identifier content
                               encoded as described in ISO/IEC 8825-1
                               [BER]. The size of this field shall not
                               exceed 127 bytes.}

<relative-uid-element> ::= 0DH <relative-uid-length> <relative-uid>
                               {Relative encoding of a universal
                               identifier as described in ISO/IEC 8825-1
                               [BER].}

<relative-uid-length> ::= <byte> {Length of <relative-uid>. This value
                               shall range between 00H and 7FH.}

<relative-uid> ::= <byte>*      {Relative object identifier content
                               encoded as described in ISO/IEC 8825-1
                               [BER].}
```

Note that the Connectionless-mode Association Control Service Element contains elements that encapsulate universal identifiers (such as <aSO-context-element>, <called-AP-title-element>, <calling-AP-title-element>, and <mechanism-name-element>). In this context, the ASN.1 tags of the universal identifier may be different from the tags defined in this subclause.

For example:

Let the ApTitle	= <application-context-oid>.0.156.5454 and
let the ANSI C12 root ApTitle	= <application-context-oid>.0

Then the Calling ApTitle element (<calling-AP-title-element>) shall be encoded (in hexadecimal notation) as follows:

Universal Calling ApTitle encoding= **A6 0E 06** 0C 60 7C 86 F7 54 01 16 00 81 1C AA 4E

Relative Calling ApTitle encoding = **A6 06 80** 04 81 1C AA 4E

Similarly for the Called ApTitle element (<calling-AP-title-element>)

Universal Called ApTitle encoding = **A2 0E 06** 0C 60 7C 86 F7 54 01 16 00 81 1C AA 4E

Relative Calling ApTitle encoding = **A2 06 80** 04 81 1C AA 4E

where **A2_H** and **A6_H** are the ACSE assigned tags of the Called and Calling ApTitle elements, respectively; and within each we have the encapsulate tags **06_H** and **80_H** that introduce the actual universal and relative identifiers, respectively. Please consult the <acse-pdu> element definitions for more details.

5.2.4 Universal identifiers canonical encoding

The C12.22 standard supports two forms of ApTitles, Universal (Absolute) and Relative ApTitles. The standard allows for C12.22 Relays to change the ApTitles form when a C12.22 Message is forwarded from one C12.22 Network Segment to another. In the context of C12.22 Message security, modification to ApTitles can be problematic. The called ApTitles and the calling ApTitles (when present) are part of the C12.22 Message authentication process and confidentiality nonce construction. If a message is modified during transmission, the authentication will fail. To resolve this, authentication of C12.22 Messages is always done against the Universal form of the <calling-AP-title> and the <called-AP-title>, regardless of how the message was transmitted. The implication of this is that the receiving C12.22 Node must be able to convert a relative ApTitle to a universal ApTitle form upon receipt before enacting the security algorithms.

C12.22 Relays, at their option, may transform relative ApTitles into absolute ApTitles. In order to ensure that relative ApTitles can be unambiguously converted, all C12.22 Message ApTitles that transverse any C12.22 Relay ApTitles shall always be universal (absolute) or they shall be relative to the <application-context-oid>.0.

When a C12.22 Node sends a C12.22 Message to another C12.22 Node that resides on the same C12.22 Segment (direct messaging), all relative ApTitles shall be relative to <application-context-oid>.0.

Note that when a C12.22 Message is sent using a proxy C12.22 Relay, the message sender shall indicate this fact by setting the PROXY_SERVICE_USED flag of the <epsem-control> of <epsem> to one (1). The recipient of any C12.22 Message that has this flag set to one (1) shall not include the <calling-AP-title-element> in the computation of the <mac> (authentication verification code).

Furthermore, note that proxy C12.22 Relays are trusted by the C12.22 Nodes that they serve. Therefore, proxy C12.22 Relays shall validate their content prior to passing the C12.22 Message to their client C12.22 Nodes on the return path (from the C12.22 Network to the local C12.22 Network Segment).

5.3 Layer 7—application layer

The Application Layer provides a minimal set of services and data structures required to support C12.22 Nodes for purposes of configuration, programming, and information retrieval in a networked environment.

This layer is composed of the following four nested components:

- ANSI C12.19 Table data structure
- EPSEM as defined in this section
- ACSE association control as defined by **ISO/IEC 15955** and presented in this subclause

5.3.1 Data structure—utility industry data tables

The data structures transported by this protocol are Tables as defined in ANSI C12.19.

5.3.2 EPSEM

This standard defines thirteen (13) EPSEM services. Each service consists of a request and a response. Each of these requests and responses is described in 5.3.2.1 through 5.3.2.7.

```

<request>      ::= <ident> |      { * Identification Service request }
                  <read> |        { Read Service request }
                  <write> |       { Write Service request }
                  <logon> |       { * Logon Service request }
                  <security> |    { Security Service request }
                  <logout> |     { * Logout Service request }
                  <terminate> |  { * Terminate Service request }
                  <disconnect> | { * Disconnect Service request }
                  <wait> |       { * Wait Service request }
                  <register> |   { ** Registration Service request }
                  <deregister> | { ** Deregistration Service }
                  <resolve> |   { ** Resolve Service request }
                  <trace> |     { ** Trace Service request }

<response>     ::= <ident-r> |   { * Identification Service response }
                  <read-r> |     { Read Service response }
                  <write-r> |    { Write Service response }
                  <logon-r> |    { * Logon Service response }
                  <security-r> | { Security Service response }
                  <logout-r> |   { * Logout Service response }
                  <terminate-r> | { * Terminate Service response }
                  <disconnect-r> | { * Disconnect Service response }
                  <wait-r> |     { * Wait Service response }
                  <register-r> | { ** Registration Service response }
                  <deregister-r> | { ** Deregistration Service response }
                  <resolve-r> |  { ** Resolve Service response }
                  <trace-r> |   { ** Trace Service response }

```

NOTE 1—* Definition or content revised from ANSI C12.18 and/or ANSI C12.21.¹⁴

NOTE 2—** New in ANSI C12.22.

NOTE 3—The network management services such as the <register>, <deregister>, <resolve>, and <trace> services may be transmitted authenticated but not encrypted.

5.3.2.1 Request codes

EPSEM requests always include a one-byte request code. Code numbers are assigned as follows:

00 _H –1F _H	Codes shall not be used to avoid confusion with response codes
20 _H –7F _H	Codes are available for use within ANSI C12 protocols
80 _H –FF _H	Codes shall be reserved for protocol extensions

¹⁴ Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

5.3.2.2 Response codes

EPSEM responses always include a one-byte response code. These codes are listed below in a suggested order of priority. They represent an extension to the response codes available in ANSI C12.18 and ANSI C12.21. When more than one response code is capable of indicating the error response condition of a C12.22 Node, the response code having the highest priority (from left to right) may be provided as follows:

```
<nok> ::= <sns>|<iss>|<iar>|<sme>|<isc>|<onp>|<bsy>|<dlk>|<dnr>|
          <rno>|<uat>|<netr>|<nett>|<rctl>|<rctl>|<sgnp>|<sgerr>|<err>
```

For example, if a C12.22 Device with a C12.22 Application contains ANSI C12.19 Tables, and Table 05 of this device is read-only, and it is encoded in nonvolatile memory, then a Write Service request to Table 05 would fail. The C12.22 Device may consider the following codes as suitable responses: <err> to indicate an error condition or <dlk> to indicate that the data is locked in memory and cannot be changed, <iar> to indicate that the action requested was not appropriate for this device design, or <isc> to indicate that the table access permission are “read-only” under the current security policy. The correct response would be <iar> as it is the highest priority among <iar>, <isc>, <dlk>, and <err>. However, if there is a mechanism for providing write access to Table 05, then <iar> should not be considered. Therefore, the response code becomes <isc>.

All service responses shall support communication-related error codes that may be generated by the intermediate C12.22 Relay (or internal interfaces):

```
<nete> ::= <netr>|<nett>|<rctl>|<rctl>|<sgnp>|<sgerr>
```

Responses

<ok>	::= 00 _H	{Acknowledge No problems, request accepted.}
<err>	::= 01 _H	{Error This code is used to indicate rejection of the received service request. The reason for the rejection is not provided.}
<sns>	::= 02 _H	{Service Not Supported This Application-level error response will be sent to the device when the requested service is not supported. This error indicates that the message was valid but the request could not be honored. The <sns> error response has special implications in the context of a response to a Logoff, Terminate, or Disconnect service request. Specifically, a C12.22 Node in the Session State shall not issue this error, but it is permitted if the C12.22 Node only supports sessionless communication.}
<isc>	::= 03 _H	{Insufficient Security Clearance This Application-level error indicates that the current authorization level is insufficient to complete the request.}

IEEE Std 1703-2012
IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol
to Complement the Utility Industry End Device Data Tables

<onp>	::= 04 _H	{Operation Not Possible This Application-level error will be sent to the device that requested an action that is not possible. This error indicates that the message was valid but the message could not be processed and covers conditions such as invalid <length> or invalid <offset>. It can also be issued if the operation is not possible under the current C12.22 Node configuration.}
<iar>	::= 05 _H	{Inappropriate Action Requested This Application-level error indicates that the action requested was inappropriate. It covers conditions such as a Write Service request to a read-only Table or an invalid Table identifier.}
<bsy>	::= 06 _H	{Device Busy This Application-level error indicates that the request was not acted upon because the device was busy doing something else. The operation may be retried at a later time with success, as the data may then be ready for transportation during this active communication.}
<dnr>	::= 07 _H	{Data Not Ready This Application-level error indicates that the request was unsuccessful because the requested data is not ready to be accessed.}
<dlk>	::= 08 _H	{Data Locked This Application-level error indicates that the request was unsuccessful because the data cannot be accessed.}
<rno>	::= 09 _H	{Renegotiate Request This Application-level error indicates that the responding device wishes to return to the identification or Base State and to renegotiate communication parameters.}
<isss>	::= 0A _H	{Invalid Service Sequence State This Application-level error indicates that the request cannot be accepted at the current service sequence state. For more information on service sequence states, see 5.3.2.5 "Service sequence state control". This is an indication to the C12.22 Application not to reissue this request at this time because there

		is a service sequence state problem or an out-of-order operations problem.}
<sme>	::= 0B _H	{Security Mechanism Error This Application-level error may be returned when a security mechanism error is detected. This code covers errors such as a security mechanism not being supported and invalid encryption key.}
<uat>	::= 0C _H	{Unknown Application Title This Application-level error may be returned by a C12.22 Relay or the target node when an unknown or invalid <called-AP-title> is received.}
<nett>	::= 0D _H	{Network Time-out This Application-level error may be returned when a Network Time-out is detected.}
<netr>	::= 0E _H	{Network Not Reachable This Application-level error may be returned when a node is not reachable.}
<rctl>	::= 0F _H <max-request-size>	{Request Too Large This Application-level error may be returned when the request size is too large.}
<max-request-size>	::= <word32>	{Maximum request size in bytes allowed by the target device.}
<rctl>	::= 10 _H <max-response-size>	{Response Too Large This Application-level error may be returned when the response size of a response is too large.}
<max-response-size>	::= <word32>	{Maximum response size in bytes allowed by the target device.}
<sgnp>	::= 11 _H	{Segmentation not possible This Application-level error may be returned when a C12.22 Node received a segment and does not support the Application Segmentation Sub-layer.}
<sgerr>	::= 12 _H <segment-byte-offset> <apdu-size>	{Segmentation error This Application-level error may be returned when a C12.22 Node fails to segment or reassemble an <acse-pdu>.

<code><segment-byte-offset> ::= <byte> <word16> <word24></code>	<code>{Offset in bytes relative to the beginning of the fully assembled APDU in which the first error was detected.}</code>
<code><apdu-size> ::= <byte> <word16> <word24></code>	<code>{Size of the fully assembled APDU, <acse-pdu>, in bytes. The data encoding format of the <apdu-size> and <segment-byte-offset> shall be identical.}</code>
<code>13_H-1F_H</code>	<code>{Response codes in this range are not defined by this standard.}</code>
<code>20_H-7F_H</code>	<code>{These codes shall not be used to avoid confusion with request codes.}</code>
<code>80_H-FF_H</code>	<code>{These codes shall be reserved for protocol extensions.}</code>

5.3.2.3 Time-out

5.3.2.3.1 Session time-out

Each session established with a C12.22 Server shall be monitored by the C12.22 Server and shut down when the session becomes inactive. An inactive session is one that does not receive EPSEM messages from the C12.22 Client within an allowable time period (the Session Time-out). An EPSEM message may be a request or a response.

The Session Time-out value is set by the Logon Service request and can be temporarily modified for the next request through the use of the Wait Service.

The Session Time-out interval starts upon transmission by the C12.22 Server of an <ok> response to a Logon Service request that was issued by the C12.22 Client. The timer restarts upon transmission or reception of any byte of an <acse-pdu> on the C12.22 Server side during a session.

The Time-out timer stops when the session ends for any reason.

When multiple concurrent sessions are supported, the Session Time-out for each session is set independently by the Logon Service request that established the session.

The Session Time-out timer is not used in sessionless communication.

5.3.2.3.2 Application layer response time-out

The Application Layer Response Timeout is used by a C12.22 Node that issues service requests to another C12.22 Node and needs to know how long to wait for responses.

A nonrecoverable Application Layer Response Timeout shall terminate the associated session if one exists. A nonrecoverable Application Layer Response Timeout is the last one, for implementations that allow retries, or the first one in implementations that do not.

An example time-out algorithm is described in Annex F “(informative) APDU Response Timeout algorithm”

5.3.2.4 Services

5.3.2.4.1 Identification service

This service is used to obtain information about C12.19 Device functionality. The service returns a code identifying the reference standard, the version and revision of the reference standard implemented, and an optional feature list.

Request:

<ident> ::= 20_H

Response:

The responses <iss>, <bsy>, and <err> indicate a problem with the received service request. The response <ok> indicates the Identification Service request was accepted and the standard, version, revision, and optional feature list are included in the response.

<ident-r> ::= <iss> | <bsy> | <err> | <nete> |
<ok> <std> <ver> <rev> <feature>* <end-of-list>

<std> ::= <byte> {Code identifying reference standard. The codes are defined as follows:

00_H = ANSI C12.18
01_H = Reserved
02_H = ANSI C12.21
03_H = ANSI C12.22
04_H-FF_H = Reserved

This value shall be 03_H.)

<ver> ::= <byte> {Binary representation of the referenced standard version number to the left of the decimal point. This value shall be 01_H.)

<rev> ::= <byte> {Binary representation of the referenced standard version number to the right of the decimal point. This value shall be 00_H.)

<feature> ::= <security-mechanism> |
<session-ctrl> |
<device-class> |
<device-identity> {Features available.}

<end-of-list> ::= 00_H {End of list indicator.}

<security-mechanism> ::= 04_H <universal-id-element> |
04_H <relative-uid-element>
{Present in the feature list only if the C12.22 Node supports one or more security mechanisms. This feature element contains the universal id of the security mechanism supported. See the <mechanisms-name-element> in 5.3.4 "Association

control-Association Control Service
Element (ACSE)" for more information.}

<session-ctrl> ::= 05_H <byte> {Bit 0 to 6: NBR_SESSION_SUPPORTED
If greater than zero, the C12.22 Node supports session-based communication. A Session starts by the reception of the Logon Service and ends by the reception of the Logoff Service or the detection of a Session Time-out.

Bit 7: SESSIONLESS_SUPPORTED
If true, the C12.22 Node supports the use of the Read and Write Services outside a session.

By default, when <session-ctrl> field is not included in the identification response, one session and sessionless operations are supported.}

<device-class> ::= 06_H <absolute-device-class-element> |
06_H <relative-device-class-element>
{A Universal Identifier that uniquely identifies a C12.19 Device Class object for early detection per the value of the MANUFACTURER element as defined in Table 0 of ANSI C12.19-1997 or the DEVICE_CLASS element as defined in ANSI C12.19-2008. When <device-class> is provided it shall be placed before <feature>s with codes that are greater than 06_H. The C12.19 Device Class, e.g., <device-class-root-oid>.<device-class>, encoded as described in ISO/IEC 8825-1 [BER]. The last four bytes of this identifier shall be identical to the values delivered in the C12.19 Table Element MANUFACTURER as defined in Table 00 of ANSI C12.19-1997 or the DEVICE_CLASS as defined in ANSI C12.19-2008. For example, C12.19 Device Class "<device-class-root-oid>.26.0.0.0" can be encoded relatively as ".26.0.0.0".}

<absolute-device-class-element> ::= <universal-id-element>
{The absolute encoding of the object identifier encoded as described in ISO/IEC 8825-1 [BER].}

<relative-device-class-element> ::= <relative-uid-element>
{The relative encoding of the object identifier encoded as described in ISO/IEC 8825-1 [BER].}

<device-identity> ::= 07_H <identity-length><identity>
{An Identifier that uniquely identifies a C12.19 Device in the application space of the C12.19 Device. This provides for early detection of the device identification element as per IDENTIFICATION of the Device Identification Table (Table 05), or DEVICE_ID found in the Utility Information Table (Table 06) of ANSI C12.19. The C12.19 <device-identity> feature shall be supplied when the C12.19 Device's Table 05 or Table 06 are readable immediately following the <logon> service. When C12.19 Device Identification is provided, it shall not precede <feature>s with codes that are less than 07_H.}

<identity-length> ::= <byte> {Length of number of bytes that follow in <identity>. This value shall range between 01_H and 7F_H.}

<identity> ::= <char-format><identification>
{Provides for early (pre-logon) disclosure of the C12.19 Device identification.}

<char-format> ::= <byte> {The character encoding format of the bytes that make up <identification>. Its interpretation shall be according to the relevant ANSI C12.19 Standard data model referenced by the C12.19 registered Device Class feature <device-class>. When the <device-class> feature is not supplied in this <ident> response, the value of <char-format> shall be set to 01_H, and <identification> shall be encoded according to ISO 7-bit coded character set for information interchange, per **ISO/IEC 646**.}

<identification> ::= <byte>* {The C12.19 Device identification string encoded and transmitted according to <char-format>. If the C12.19 Device's ID_FORM in Table 00 is set to BCD, then the BCD digits shall be transmitted as their text equivalent also encoded as per <char-format>.

For example, if the C12.19 Device's GEN_CONFIG_TBL.ID_FORM is BCD and the device's GEN_CONFIG_TBL.CHAR_FORMAT is ISO 7-bit ASCII, as per **ISO/IEC 646**, then the BCD digits 00_H 01_H 02_H 03_H 0A_H 04_H 0D_H 05_H 06_H 07_H shall be transmitted as the

character sequence "123-4.567". The C12.19 application shall logically pad this string with trailing spaces, as needed, to fill the identification field width of the C12.19 Device.}

5.3.2.4.2 Read service

The Read Service is identical to that found in ANSI C12.18 and ANSI C12.21 with the inclusion of additional error response codes defined by this standard and the addition of the capability to receive Table data in excess of 65 535 bytes.

The Read Service is used to cause a transfer of Table data to the requesting device.

Request:

The read request allows both complete and partial Table transfers. The retrieval of a portion of a Table is possible through the use of both index/element-count and offset/octet-count methods. The complete rule set for using these methods is enumerated in 5.3.2.6 "Partial table access using index/element-count method" and 5.3.2.7 "Partial table access using offset/octet-count method", respectively.

Request codes 30_H–39_H, 3E_H, and 3F_H give access to all possible methods used for Table transfer. Request code 30_H specifies a complete Table transfer. Request codes 31_H through 39_H specify a partial Table transfer using 1 to 9 indices. Request code 3E_H specifies a default Table transfer. Request code 3F_H specifies a partial Table transfer using the offset/octet-count method.

<read> ::= <full-read> | <pread-index> | <pread-offset> | <pread-default>

<full-read> ::= 30_H <tableid>

<pread-index> ::= <read-index-type> <tableid> <index>+ <element-count>

<read-index-type>	::=	31 _H 32 _H 33 _H 34 _H 35 _H 36 _H 37 _H 38 _H 39 _H	{ 1 <index> included in request { 2 <index> included in request { 3 <index> included in request { 4 <index> included in request { 5 <index> included in request { 6 <index> included in request { 7 <index> included in request { 8 <index> included in request { 9 <index> included in request
-------------------	-----	---	---

<pread-default> ::= 3E_H {Transfer default Table}

<pread-offset> ::= 3F_H <tableid> <offset> <octet-count>

<tableid> ::= <word16> {Table identifier as defined in ANSI C12.19.}

<offset> ::= <word24> {Offset into data Table in bytes. Offset 0000_H is the offset to the first Table element of the Table selected by <tableid>..}

IEEE Std 1703-2012
IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol
to Complement the Utility Industry End Device Data Tables

<index>	::= <word16>	{Index value used to locate start of data. Index 0000 _H + is the index of the first Table element of the Table selected by <tableid>..}
<element-count>	::= <word16>	{Number of ANSI C12.19 Table Elements to read starting at the requested <index>..}
<octet-count>	::= <word16>	{Length of ANSI C12.19 Table data requested starting at <offset> in bytes}

Response:

Responses of type <nok> indicate a problem with the received service request.

The response <ok> indicates the Read Service was accepted and the <data> is part of the response.

<read-r>	::= <nok> <ok> <table-data>+	
<table-data>	::= <count> <data> <cksum>	
<count>	::= <word16>	{Length of <data> returned, in bytes, in this <table-data> element. When <count> equals to FFFF _H , it is an indication that the length of <data> returned in this <table-data> is 65 535 bytes, and that another <table-data> element shall follow. The last <table-data> element shall have a <count> that is less than FFFF _H . When the last <table-data> is exactly 65 535 bytes, it is followed by <table-data> with <count> equal to 0 _H . This <count> includes the optional pending header length as defined by ANSI C12.19.}
<data>	::= <byte>*	{The returned C12.19 Table data including the optional ANSI C12.19 pending header when requested. The optional pending header may be placed only in the beginning of the <data> of the first <table-data> element of any given read service response.}
<cksum>	::= <byte>	{2's complement checksum computed only on the <data> portion of <table-data>. The checksum is computed by summing the bytes (ignoring overflow) and negating the result.}

5.3.2.4.3 Write service

The Write Service is identical to that found in ANSI C12.18 and ANSI C12.21 with the inclusion of additional error response codes defined by this standard and the addition of the capability to transmit Table data in excess of 65 535 bytes.

The Write Service is issued to transfer Table data to the target device.

Request:

The Write Request allows both complete and partial Table transfers. The modification of a portion of a Table is possible through the use of both index/count and offset/count methods. The complete rule set for using these methods is enumerated in 5.3.2.6 and 5.3.2.7, respectively.

Request codes 40_H–49_H and 4F_H give access to all possible methods used for Table transfer. Request code 40_H specifies a complete table transfer. Request codes 41_H through 49_H specify a partial Table transfer using 1 to 9 indices. Request code 4F_H specifies a partial Table transfer using the offset/count method.

```

<write>                ::= <full-write> | <pwrite-index> | <pwrite-offset>

<full-write>           ::= 40H <tableid> <table-data>+

<pwrite-index>         ::= <write-index-type> <tableid> <index>+ <table-data>

<write-index-type>     ::=
    41H | {1 <index> included in request}
    42H | {2 <index> included in request}
    43H | {3 <index> included in request}
    44H | {4 <index> included in request}
    45H | {5 <index> included in request}
    46H | {6 <index> included in request}
    47H | {7 <index> included in request}
    48H | {8 <index> included in request}
    49H | {9 <index> included in request}

<pwrite-offset>        ::= 4FH <tableid> <offset> <table-data>

<tableid>              ::= <word16>          {Table identifier as defined in
                                              ANSI C12.19.}

<offset>               ::= <word24>          {Offset into data Table in bytes. Offset
                                              0000H is the offset to the first element
                                              of the Table selected by <tableid>..}

<index>                ::= <word16>          {Index value used to locate start of
                                              data. Index 0000H+ is the index of the
                                              first element of the Table selected by
                                              <tableid>..}

<table-data>          ::= <count> <data> <cksum>

<count>               ::= <word16>          {Length of <data> to be written, in
                                              bytes, from this <table-data> element.
                                              When <count> equals to FFFFH, it is an
                                              indication that the length of <data>
                                              contained in this <table-data> is 65 535
                                              bytes, and that another <table-data>
                                              element shall follow. The last <table-
                                              data> element shall have a <count> that
                                              is less than FFFFH. When the last <table-
                                              data> is exactly 65 535 bytes, it is
                                              followed by <table-data> with <count>
                                              equal to 0H. This <count> includes the

```

		optional pending header length as defined by ANSI C12.19.}
<data>	::= <byte>*	{The Table data elements including the optional pending header as per ANSI C12.19 when requested. The optional pending header may be placed only in the beginning of the <data> of the first <table-data> element of any given write service request.}
<cksum>	::= <byte>	{2's compliment checksum computed only on the <data> portion of <table-data>. The checksum is computed by summing the bytes (ignoring overflow) and negating the result.}

Response:

Responses of type <nok> indicate a problem with the received service request.

The response <ok> indicates the Write Service was successfully completed and the data was successfully transmitted to the device.

<write-r>	::= <nok> <ok>
-----------	---------------------

5.3.2.4.4 Logon service

Logon Service establishes a session without establishing access permissions. It provides for immediate transfer to the session state from the idle state. A peer-to-peer association shall be established.

Request:

The <user-id> may be inserted in the Event and History Logs as defined in ANSI C12.19, when supported by the C12.22 Node. The <user> field provides the name of the operator requesting the access to the C12.22 Node and may be recorded in the Utility Information Table (Table 06).

The Logon Service request has the following format:

<logon>	::= 50 _H <user-id> <user> <req-session-idle-timeout>
<user-id>	::= <word16> {User identification code}
<user>	::= <byte> ⁺¹⁰ {10 bytes containing user identification}
<req-session-idle-timeout>	::= <word16> {The desired number of seconds a session may be idle on the C12.22 Server side before the C12.22 Server may terminate the session. A value of zero means a request to keep the session open forever.}

Response:

All responses other than <ok> indicate a problem with the received service request and the session and the association shall not be established. The C12.22 Node shall remain in the idle state.

All error responses (including those that are not listed below or extensions on those covered by <nok>) shall be generically interpreted as an indication to the application not to issue another Logon request without first addressing the cause of the indicated error condition.

The response <ok> indicates that the Session was successfully established.

```
<logon-r>          ::= <nok> |
                                <ok> <resp-session-idle-timeout>

<resp-session-idle-timeout> ::= <word16>
                                {The number of seconds a session may be
                                 idle on the C12.22 Server before the
                                 C12.22 Server may terminate the Session.
                                 This value shall be less than or equal to
                                 <req-session-idle-timeout>.}
```

5.3.2.4.5 Security service

The Security Service is provided for setting access permissions.

It should be noted that sending passwords as Cleartext (unencrypted) over the network is a security concern. Available privacy and authentication encryption services are described in 5.3.4.13 “C12.22 Security Mechanism”.

It should also be noted that the EPSEM Security Service is unrelated to the C12.22 Security Mechanism as described in 5.3.4.13 “C12.22 Security Mechanism”. The Security Service works solely within the context of the EPSEM services. Its purpose is to provide a mechanism to change the C12.19 Tables access privilege or authorization when issuing commands to a C12.22 Node. It does not “secure” the C12.22 Message in any way.

The Security Mechanism uses cryptographic techniques to provide authentication and privacy for C12.22 Messages, regardless of the contents of the C12.22 Message. The Security Mechanism works at the C12.22 Network Application Layer level, and it is described in detail in 5.3.4.13 “C12.22 Security Mechanism”.

Request:

A password is used as a means to select access permissions. The <password> field will be compared with those in the Security Table (Standard Table 42) defined in ANSI C12.19, if the password Tables are supported by the C12.22 Node.

```
<security>          ::= 51H <password> [ <user-id> ]

<password>          ::= <byte>+20      {20 byte field containing password}

<user-id>           ::= <word16>      {User identification code. This field
                                       shall be present when this service is
                                       included in a sessionless message. In the
                                       context of a session, this information is
                                       provided by the Logon service and shall
                                       not be included in this service.}
```

Response:

The responses <sns>, <isss>, <bsy>, and <err> indicate a problem with the received service request.

The response <ok> indicates the security service was successfully completed and the access permissions associated with the password were granted.

<security-r> ::= <sns> | <isss> | <bsy> | <err> | **<nete>** |
<ok>

5.3.2.4.6 Logoff service

The Logoff Service provides for an orderly termination of the session that was established by the Logon Service. It provides for immediate transfer to the idle state from the session state. The peer-to-peer association shall terminate, and all previously negotiated settings shall reset to their default idle-state values.

The Physical Layer signaling parameters of the C12.22 Node shall not be affected by this service.

Request:

<logoff> ::= 52_H

Response:

All responses other than <ok> indicate a problem with the received service request, and the session and the association shall retain the settings negotiated prior to the issuance of the Logoff Service request unless otherwise specifically indicated by the error code.

All error responses shall be generically interpreted as an indication to the application not to issue another Logoff Service request without first addressing the cause of the indicated error condition.

When a response other than <ok> is understood to be an indication other than the severance of the communication path or association, the application may issue any valid session state service request or choose to terminate the association or it may let the Session Time-out period expire to force the Session to be aborted.

<logoff-r> ::= <nok> |
<ok> {Response code or error codes as per
5.3.2.6 "Response codes" **.**}

The response <ok> indicates the successful termination of the Session. This is an indication to the C12.22 Application that the C12.22 Node completed all session-related processing before terminating the session and that it reset its communication parameters to their default settings and entered the idle state. The Association between the peer C12.22 Nodes was terminated.

5.3.2.4.7 Terminate service

The Terminate Service provides for an orderly abortion of the Session that was established by the Logon Service. It provides for transfer to the idle state from the session state. The peer-to-peer Association shall terminate, and all previously negotiated settings shall reset to their default idle-state values.

This application-level service may be used to terminate a Session for reasons such as excessive errors, security issues, internal error conditions, a need to abort session, or other reasons.

The Physical Layer signaling parameters of the C12.22 Node shall not be affected by this service.

$$\langle \text{terminate} \rangle ::= 21_H$$

All responses other than <ok> indicate a problem with the received service request, and the session and the association shall retain the settings negotiated prior to the issuance of the Terminate Service request.

When a not <nok> response is understood to be an indication other than the severance of the communication path or association, the application may issue any valid session state service request or choose to terminate the Association or it may let the Session timeout to force the session to be aborted.

$$\langle \text{terminate-r} \rangle ::= \langle \text{nok} \rangle \mid \langle \text{ok} \rangle$$

5.3.2.4.8 Disconnect service

The Disconnect Service, when issued within a Session, provides for an orderly abortion of the Session that was established by the Logon Service. It is functionally equivalent to the Terminate Service request that is followed by a transition to the off-line state.

All peer-to-peer Associations across the interface of the C12.22 Node on the C12.22 Network segment that processed this request shall terminate. The C12.22 Node's settings shall reset to their default off-line state values for that C12.22 Network Segment.

For this service request to be successful, the initiator shall have write access permission to Procedure 25 NETWORK INTERFACE CONTROL PROC.

$$\langle \text{disconnect} \rangle ::= 22_{\text{H}}$$

All responses other than <ok> indicate a problem with the received service request and the Physical Layer signaling parameters, session, and the association shall retain the settings negotiated prior to the issuance of the Disconnect Service request.

31

When a <nok> response is understood to be an indication other than the severance of the communication path or association, the C12.22 Application may issue any valid service request or choose to terminate the association or it may let the association time out to force a session to be aborted.

```
<disconnect-r> ::= <sns> | <iar> | <sme> | <isc> | <onp> | <bsy> | <dlk> | <dnr> |  
                  <rno> | <uat> | <nete> |  
                  <err> |  
                  <ok> {Response code or error codes as defined  
                        in 5.3.2.2 "Response codes".}
```

The response <ok> indicates that the Disconnect Service was accepted. If the C12.22 Node was in the session state, then this is an indication of the successful abortion of the Session.

This is also an indication that the C12.22 Node aborted all Session-related processing, removed itself from the C12.22 Network Segment (by de-asserting the appropriate Physical Layer signals), entered the off-line state, and reset its communication parameters to their default settings. The Association between the peer C12.22 Nodes was terminated, and all other Associations that share the same Interface on the C12.22 Node network segment that processed this response were also terminated.

5.3.2.4.9 Wait service

The Wait Service is used to maintain an established Session during idle periods, thus preventing automatic termination. This service temporarily extends the Session Time-out to the <time> specified in the request upon acknowledgment of the Wait Service request. The Session Time-out will be reset to the default value once the next valid service is received by this target.

Request:

```
<wait>           ::= 70H <time>  
<time>          ::= <byte> {Requested wait period in seconds. The  
                           value zero does not affect the Channel  
                           settings.}
```

Response:

The responses <sns>, <iss>, <bsy>, and <err> indicate a problem with the received service request and Session Time-out is not extended.

The response <ok> indicates the service request was accepted and the Session Time-out is extended to the value requested.

```
<wait-r>         ::= <sns> | <iss> | <bsy> | <err> | <nete> |  
                  <ok>
```

5.3.2.4.10 Registration service

The Registration Service is used to add and keep routing-table entries of C12.22 Relays active. To be part of a C12.22 Network, a C12.22 Node shall send a Registration Service request to one of the C12.22 Master Relays. This service is carried in an ACSE Application Data Unit <acse-pdu> with the Calling ApTitle set to the ApTitle of the registering C12.22 Node and the Called ApTitle set to the ApTitle of the C12.22 Master Relay. This service carries the node type, the node class, serial number, registration lifetime parameters, and an optional native address of the registering node. The native address is used only by the neighbor C12.22 Relay to send messages back to this node, and it is ignored by all others nodes.

The C12.22 Master Relay shall send a copy of each first registration received (“first” here meaning an actual registration request as contrasted with the reuse of the register service as keep-alive) to all C12.22 Notification Hosts and all C12.22 Authentication Hosts that need to be notified. In this case, the Registration Service is sent with the Calling ApTitle set to the ApTitle of the C12.22 Master Relay and the Called ApTitle set, in turn, to the ApTitle of each C12.22 Host notified and all other parameters set in accordance with the registration response that was issued by the C12.22 Master Relay to the registered C12.22 Node. The C12.22 Master Relay shall send updated copies of each first registration to subscribed C12.22 Notification Hosts when the C12.22 Notification Hosts subscribe to this service, including registration records that came into effect before the subscription for notifications.

Request:

```
<register>      ::= 27H <node-type> <connection-type> <device-class>  
                  <ap-title> <electronic-serial-number>  
                  <address-length> <native-address> <registration-period>  
                  [<my-domain-pattern>]
```

```
<node-type>     ::= <byte> {An identification of the C12.22 Node's  
                           Attributes. These values may be set to  
                           advertise the capabilities of this C12.22  
                           Node and to assist C12.22 Master Relay in  
                           the decision making for the successful  
                           completion of all communication with  
                           other C12.22 Nodes that participate in  
                           the registration process.}
```

Bit 0 to 5: NODE_TYPE as per
INTERFACE_STATUS_TBL.NODE_TYPE_BFLD.

Bit 0: RELAY

When set to 1, it is an indication that this C12.22 Node is a C12.22 Relay. All C12.22 Relays shall set this bit to 1.

Bit 1: MASTER_RELAY

When set to 1, it is an indication that this C12.22 Node is a C12.22 Master Relay. All C12.22 Master Relays shall set this bit to 1.

Bit 2: HOST

When set to 1, it is an indication that this C12.22 Node is a C12.22 Host.

Bit 3: NOTIFICATION_HOST

When set to 1, it is an indication that this C12.22 Node is a C12.22 Notification Host. All C12.22 Notification Hosts shall set this bit to 1, if they wish the C12.22 Master Relay to add them to their list of Notification Hosts or issue notifications to this C12.22 Node when other C12.22 Nodes register with the servicing C12.22 Master Relay (See <called-AP-title>). Note that this does not preclude static registration;

however, notifications may not be received until the C12.22 Master Relay registers this C12.22 Node as a C12.22 Notification Host.

Bit 4: AUTHENTICATION_HOST

When set to 1, it is an indication that this C12.22 Node is a C12.22 Authentication Host. All C12.22 Authentication Hosts shall set this bit to 1, if they wish the C12.22 Master Relay to add them to their list of Authentication Hosts or issue registration requests to this C12.22 Node when other C12.22 Nodes register with the servicing C12.22 Master Relay (see <called-AP-title>). Note that this does not preclude static registration; however, notifications may not be received until the C12.22 Master Relay registers this C12.22 Node as a C12.22 Authentication Host.

Bit 5: END_DEVICE

When set to 1, it is an indication that this C12.22 Node is a C12.19 Device. When set to 0, it is an indication the C12.22 Node is not a C12.19 Device.

Bit 6: Reserved and shall be set to 0.

Bit 7: MY_DOMAIN_PATTERN

When set to 0, the <my-domain-pattern> parameter is not present. When set to 1, the <my-domain-pattern> parameter is present.

<connection-type> ::= <byte>

{An indication of the type of connection requested and the core capability related to this C12.22 Node in regard to its connection to the C12.22 Network Segment. These bits are identical to those defined by INTERFACE_STATUS_TBL. CONNECTION_TYPE_BFLD.

Bit 0: BROADCAST_AND_MULTICAST_SUPPORTED

Set to one (1) when the C12.22 Node has the capability to accept broadcast and multicast messages. Set to zero (0) otherwise.

Bit 1:

MESSAGE_ACCEPTANCE_WINDOW_SUPPORTED

Set to one (1) when the C12.22 Node is capable of implementing time-based C12.22 Message acceptance windows. When set to

0, it is an indication that this C12.22 Node is not capable of implementing time-based C12.22 Message acceptance windows.

Bit 2: PLAYBACK_REJECTION_SUPPORTED

This bit is set to one (1) when the C12.22 Node is capable of performing playback rejection algorithms on duplicate incoming C12.22 Messages. This bit is set to zero (0) when the C12.22 Node is not capable of performing playback rejection algorithms on duplicate incoming C12.22 Messages.

Bit 3: Reserved and shall be set to zero (0).

Bit 4: CONNECTIONLESS_MODE_SUPPORTED

This bit is set to one (1) if the local network segment and the node support a connectionless oriented protocol. This bit is set to zero (0) if the local network segment or the node supports do not support a connectionless protocol. At least one of CONNECTIONLESS_MODE_SUPPORTED or CONNECTION_MODE_SUPPORTED shall be set to one (1).

Bit 5: ACCEPT_CONNECTIONLESS

This bit is set to one (1) when the local network segment and the node support a connectionless oriented protocol (Bit 4 set to 1) and the registering node accepts unsolicited incoming connectionless messages.

Bit 6: CONNECTION_MODE_SUPPORTED

This bit is set to one (1) if the local network segment and the node support a connection-oriented protocol. This bit is set to zero (0) if the local network segment or the node does not support a connectionless protocol. At least one of CONNECTIONLESS_MODE_SUPPORTED or CONNECTION_MODE_SUPPORTED shall be set to one (1).

Bit 7: ACCEPT_CONNECTIONS

This bit is set to one (1) when the local network segment and the node support a connection-oriented protocol (Bit 6 set to 1) and the registering node accepts connections.

IEEE Std 1703-2012
IEEE Standard for Local Area Network/Wide Area Network (LAN/WAN) Node Communication Protocol
to Complement the Utility Industry End Device Data Tables

`<device-class> ::= <byte>+4` {A list of encoding of sub-identifiers (integers) expressed as the four bytes containing the MANUFACTURER_ID as defined in Table 00 of ANSI C12.19-1997 or the DEVICE_CLASS as defined in ANSI C12.19-2008 registered class.

This is a subset of the Relative Object Identifier defined in ISO/IEC 8825-1 and expressed by the `<relative-uid-element>` as follows:

The leading Relative Object Identifier introducer, 0D_H, and length fields are not present in the Table 00 element. The length is assumed to be 4. Then each sub-identifier is represented as a series of (one or more) bytes. Bit 7 of each byte shall be cleared to zero when it is the last member in the series. Bit 7 of each preceding octets in the series is set one. Bits 6-0 of the byte in the series collectively encode the sub-identifier concatenated to form an unsigned binary number whose most significant bit is bit 6 of the first octet and whose least significant bit is bit 0 of the last octet. Each sub-identifier value shall be encoded in the fewest possible bytes such that the leading octet of the sub-identifier will not have bit 7 set to one.

When the sub-identifiers that make up the device-class are encoded in this way, they will always result in four bytes of data.

For example, if the value reported by MANUFACTURER_ID is "TEMP", representing the relative object identifier 84.69.77.80, it is encoded as 54_H 45_H 4D_H 50_H. It shall be interpreted as a Relative UID encoding of 0D_H 04_H 54_H 45_H 4D_H 50_H.

Similarly when the value reported by DEVICE_CLASS is 8571.3 or encoded as C2_H 7B_H 03_H 00_H, it is interpreted as a Relative UID encoding of 0D_H 04_H C2_H 7B_H 03_H 00_H. }

`<ap-title> ::= <universal-id-element> | <relative-uid-element>`
{ApTitle of the C12.22 Node to be registered. The field is optional, and when not provided, its content length is set to zero, implying that the ApTitle

shall be automatically obtained from an authoritative proxy Relay. When the <ap-title> is expressed as <relative-uid-element>, then it shall be encoded relative to <application-context-oid>.0. The relative portion (which follows the <application-context-oid>.0) of a registered ap-title shall not include assigned or reserved subbranches.}

<electronic-serial-number> ::= <universal-id-element> | <relative-uid-element>
{Unique ISO object identifier assigned to this C12.22 Device by its owner and provided for registration authentication. The field is optional, and when not provided, its length is set to zero. This is the same number as appears as ELECTRONIC_SERIAL_NUMBER in Table 122 Interface Control Table.}

<address-length> ::= <byte> {Number of bytes in <native-address>.}

<native-address> ::= <byte>*
{Native address to use to forward messages to this node. This field is optional if lower layers of the protocol already provide it.

When defined for a specific protocol stack, this field can be subdivided in sub-fields to provide address type and address data to facilitate address resolution.}

<registration-period> ::= <word24> {Maximum period in seconds desired by the C12.22 Node to elapse between successive re-registration requests (keep-registration-alive). The value 0 implies that a re-registration life timer is not supplied by the registering node. The maximum value may not exceed 3 932 100 seconds (65 535 minutes), to comply with the range of REGISTRATION_LIST_TBL. REGISTRATION_LIST_ENTRY_RCD. REGISTRATION_TIME_OUT (UINT16 minutes). Values greater than 3 932 100 seconds shall be limited to 3 932 100 seconds or less.}

<my-domain-pattern> ::= <notification-pattern-length> <notification-pattern>
{The optional <my-domain-pattern> may be present only when <node-type> indicates that the registering node is a NOTIFICATION_HOST or an AUTHENTICATION_HOST.}

<notification-pattern-length> ::= <byte> {Number of bytes in <notification-pattern>. This value shall not be greater than 30 or 120. See ACT_RELAY_TBL.PATTERN_LEN_MULT.}

<notification-pattern> ::= <byte>* {An ApTitle pattern that is associated with the C12.22 Node-population that is of interest to the authentication or notification host. An ApTitle pattern is represented as a dot delimited numeric or as '*' strings as described in C.4 "Universal ID pattern description of ApTitles"). The encoding of the pattern shall be UTF-8. A registering C12.22 Notification Host or C12.22 Authentication Host may supply this pattern as a means to restrict the number of notifications that arrive from a C12.22 Master Relay to a subset that are of interest to this registering notification or authentication host. See ACT_RELAY_TBL.PATTERN_LEN_MULT.}

Response:

The response <nok> indicates that the registration was rejected for some reason by the Master Relay. The Calling ApTitle shall be set to the ApTitle of the Master Relay that responded. The response <ok> indicates that this ApTitle's <universal-id-element> has been registered and routing tables have been updated.

<register-r> ::= <nok> | <ok> <reg-ap-title> <reg-delay> <reg-period> <reg-info>

<reg-ap-title> ::= <universal-id-element> | <relative-uid-element> {Registered ApTitle to be used by the C12.22 Node for future communication on this route. When the <reg-ap-title> is expressed as <relative-uid-element>, it shall be encoded relative to the <application-context-oid>.0. This ensures that even when relative encoding is used, the C12.22 Node can determine its absolute position in the C12.22 Network hierarchy. The relative portion (which follows the <application-context-oid>.0) of a registered ap-title shall not include assigned or reserved subbranches.}

<reg-delay> ::= <word16> {Maximum delay in seconds that the device should wait before registering after a power-up. A value of 3600 seconds shall be used as a default if this value cannot be retained. The actual delay used to re-register shall be a random value between 0 and this value.}