IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)—Resource Manager

1609.1™

IEEE Vehicular Technology Society (VTS)

Sponsored by the
Intelligent Transportation Systems (ITS) Committee

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

13 October 2006

IEEE Std 1609.1™-2006

IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)— Resource Manager

Sponsor

Intelligent Transportation Systems (ITS) Committee
of the
IEEE Vehicular Technology Society (VTS)

Approved 15 September 2006

IEEE-SA Standards Board

Abstract: This standard specifies a wireless access in vehicular environments (WAVE) dedicated short-range communications (DSRC) application, known as the WAVE resource manager (RM), designed to allow applications at remote sites to communicate with devices known as onboard units (OBUs), which are mounted in vehicles, through devices known as roadside units (RSUs), which are mounted on the roadside. The WAVE RM, acting like an application layer, multiplexes the communications of multiple remote applications, each communicating with multiple OBUs. The purpose of the communication is to conduct information interchange, needed to implement the requirements of the remote WAVE DSRC applications.
Keywords: DSRC, WAVE, resource manager

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

Print: ISBN 0-7381-5241-2 SH95580
PDF: ISBN 0-7381-5242-0 SS95580

IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. The 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 the IEEE administers the process and establishes rules to promote fairness in the consensus development process, the IEEE does not independently evaluate, test, or verify the accuracy of any of the information contained in its standards.

Use of an IEEE Standard is wholly voluntary. The 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 this, or any other IEEE Standard document.

The IEEE does not warrant or represent the accuracy or content of the material contained herein, 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 herein 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 five years for revision or reaffirmation. When a document is more than five years old and has not been reaffirmed, 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 this document available, the IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is the IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing this, and any other IEEE Standards document, should rely upon the advice of a competent professional in determining the exercise of reasonable care in any given circumstances.

Interpretations: Occasionally questions may arise regarding the meaning of portions of standards as they relate to specific applications. When the need for interpretations is brought to the attention of IEEE, the Institute will initiate action to prepare appropriate responses. Since IEEE Standards represent a consensus of concerned interests, it is important to ensure that any interpretation has also received 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 interpretation requests except in those cases where the matter has previously received formal consideration. 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, explanation, or interpretation of the IEEE.

Comments for revision of IEEE Standards are welcome from any interested party, regardless of membership affiliation with IEEE. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Comments on standards and requests for interpretations should be addressed to:

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

Introduction

This introduction is not part of IEEE Std 1609.1-2006, IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)—Resource Manager.

This standard is based on IEEE Std 1455™-1999 [B4][a] but intended for the 5.9 GHz frequency band. Several changes have been incorporated to accommodate the new requirements of the communication stack described in the associated standards (IEEE Std 1609.2™,[b] IEEE P1609.3™, and IEEE P1609.4™ [B3]) and the User Datagram Protocol (UDP) described in IETF RFC 768. As well, some inconsistencies and errors in IEEE Std 1455-1999 have been addressed.

This standard specifies a wireless access in vehicular environments (WAVE) application, known as the *resource manager* (RM), designed to allow applications at remote sites to communicate with devices known as *onboard units* (OBUs), which are mounted in vehicles, through devices known as *roadside units* (RSUs), which are mounted on the roadside. The RM, acting like an application layer, multiplexes the communications of multiple remote applications, each communicating with multiple OBUs. The purpose of the communication is to conduct information interchange, needed to implement the requirements of the remote WAVE applications.

Notice to users

Errata

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

Interpretations

Current interpretations can be accessed at the following URL: http://standards.ieee.org/reading/ieee/interp/index.html.

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 with respect to the existence or validity of any patent rights in connection therewith. The IEEE shall not be responsible for identifying patents or patent applications for which a license may be required to implement an IEEE standard or for conducting inquiries into the legal validity or scope of those patents that are brought to its attention.

Publication of this trial-use standard for comment and criticism has been approved by the IEEE. Trial-use standards are effective for 24 months from the date of publication. Comments for revision will be accepted for 18 months after publication. Suggestions for revision should be directed to the Secretary, IEEE-SA Standards Board, 445 Hoes Lane, Piscataway, NJ 08854, and should be received no later than 13 April 2008. It is expected that following the 24-month period, this trial-use standard, revised and balloted as necessary, shall be submitted to the IEEE-SA Standards Board for approval as a full-use standard.

[a]The numbers in brackets correspond to the numbers of the bibliography in Annex E.

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

Participants

The following is a list of participants in the Dedicated Short-Range Communications (DSRC) Working Group.

Thomas M. Kurihara, Chair

TiVon Abrams
Scott Andrews
Lee Armstrong
James Arnold
Broady Cash
Ken Cook
Susan Dickey
Wayne Fisher
Gamez Gerges
Gloria Gwynne
Chris Hedges
Ron Hochnadel
Jason Hunzinger
Daniel Jiang
Carl Kain
Pankaj Karnik
Doug Kavner
David Kelley
Brian Kind
Peter Kofod
Japjeev Kohli
Ryan Lamm
Jerry Landt
Jules Madey
Alastair Malarky
Mira Marshall
Tim McGuickin
Justin McNew
Yasser Morgan
John Moring
Ross Morris
Richard Noens
Roger O'Connor
Peter Oomen
Sam Oyama
Frank Perry
Susan Proper
Mohan Pundari
Eric Rescora
Randy Roebuck
Richard Roy
Tom Schaffnit
Dick Schnacke
Jerry Serviss
John Sheppard
Robert Sorano
Stephen Spenler
Steve Tengler
Dan Terrier
Glenn Turnock
Bryan Wells
Filip Weytjens
William Whyte
Jijun Yin
Jeffery Zhu

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

Satish K. Aggarwal
Toru Aihara
Adewole C. Akpose
Thomas Alexander
Gerald W. Althauser
Butch Anton
Lee R. Armstrong
Charles L. Barest
John R. Barr
Saman Behtash
Sean S. Cai
James T. Carlo
Juan C. Carreon
Broady B. Cash
Jon S. Chambers
Danila Chernetsov
Elizabeth Chesnutt
Aik Chindapol
Keith Chow
J. Kenneth Cook
Tommy P. Cooper
Carlo Donati
Sourav K. Dutta
Wayne K. Fisher
Andre F. Fournier
Avraham Freedman
H. M. Glickenstein
Sergiu R. Goma
Randall C. Groves
Gloria G. Gwynne
Ronald D. Hochnadel
Werner Hoelzl
Dennis Horwitz
Russell D. Housley
David Hunter
Arshad Hussain
Peeya Iwagoshi
Raj Jain
Oh Jongtaek
Kevin J. Karcz
Pankaj R. Karnik
Piotr Karocki
Douglas M. Kavner
Thomas M. Kurihara
Jeremy A. Landt
Solomon Lee
Daniel G. Levesque
Jun Liu
William Lumpkins
G. L. Luri
Julius M. Madey
Faramarz Maghsoodlou
Justin P. Mcnew
George J. Miao
Gary L. Michel
William J. Mitchell
John T. Moring
Ross A. Morris
Michael S. Newman
Paul Nikolich
Richard H. Noens
Satoshi Obara
Peter Oomen
Satoshi Oyama
Subburajan Ponnuswamy
Vikram Punj
Philip T. Robinson
Robert A. Robinson
Randal D. Roebuck
Randall M. Safier
Osman Sakr
Stephen C. Schwarm
Rich Seifert
Gerald K. Serviss
John W. Sheppard
Robert T. Soranno
Stephen J. Spenler
Luca Spotorno
Thomas J. Timcho
Glenn Turnock
Mark-Rene Uchida
Scott A. Valcourt
Stephen C. Webb
Doulgas L. Welk
Filip Weytjens
William Whyte
Paul R. Work

When the IEEE-SA Standards Board approved this standard on 15 September 2006, it had the following membership:

Steve M. Mills, *Chair*
Richard H. Hulett, *Vice Chair*
Judith Gorman, *Secretary*

Mark D. Bowman
Dennis B. Brophy
Joseph Bruder
Richard Cox
Bob Davis
Julian Forster*
Joanna N. Guenin
Mark S. Halpin
Raymond Hapeman
William B. Hopf
Lowell G. Johnson
Herman Koch
Joseph L. Koepfinger*
David J. Law
Daleep C. Mohla
Paul Nikolich
T. W. Olsen
Glenn Parsons
Ronald C. Petersen
Gary S. Robinson
Frank Stone
Malcolm V. Thaden
Richard L. Townsend
Joe D. Watson
Howard L. Wolfman

*Member Emeritus

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

Satish K. Aggarwal, *NRC Representative*
Richard DeBlasio, *DOE Representative*
Alan H. Cookson, *NIST Representative*

Michelle D. Turner
IEEE Standards Program Manager, Document Development

Matthew Ceglia
IEEE Standards Program Manager, Technical Program Development

Contents

IEEE Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)— Resource Manager

1. Overview

There are two types of wireless access in vehicular environments (WAVE) devices. The first type, referred to as the *roadside unit* (RSU), is stationary while in operation and usually permanently mounted along the roadside. The second type, known as the *onboard unit* (OBU), may operate while mobile and is usually mounted onboard a vehicle (see Figure 1 in 4.1). Typically, the stationary devices host an application that provides a service, and the mobile devices host a peer application that uses this service. There may also be applications on devices remote from the RSU, whose purpose is to provide services to the OBU. This standard describes a WAVE application that resides on the RSU but is designed to multiplex requests from remote applications, providing them access to the OBU.

This standard specifies the WAVE application known as the *resource manager* (RM), which resides on, for example, the RSU, and its peer known as the *resource command processor* (RCP), which resides on the OBU. Remote from the RSU, the other applications, known as *resource manager applications* (RMAs), communicate with the RCP through the RM. This standard describes how the RM multiplexes requests from multiple RMAs, each of which is communicating with multiple OBUs hosting an RCP. The purpose of the communication is to provide the RMAs access to "resources," such as memory, user interfaces (UIs), and interfaces to other onboard equipment controlled by the RCP, in a consistent, interoperable, and timely manner to meet the requirements of RMAs.

The RM uses the concept of all of the communication being initiated from an entity known as a *provider*, which issues requests to an entity known as a *user*, which responds only to requests that it receives. Within this standard, the RM is the provider of a service (as a representative of the RMAs), and RCP the user of a service (representing the resources to be managed). Either the RSU or OBU can operate as the provider; in other words, either device type can host the RM. The device, either RSU or OBU, that is hosting the RM will be referred to as the *provider device*.

1.1 Scope

The scope of this standard is to specify the services and interfaces of the WAVE RM, including protective mechanisms for security and privacy, applicable and available to all users of DSRC and WAVE mode

operations in the 5.9 GHz band authorized by the Federal Communication Commission (FCC) for intelligent transportation systems (ITS).

NOTE—This version of the standard does not specify explicitly the details of the security interface. Security provisions are in IEEE Std 1609.2™.[1, 2]

1.2 Purpose

The purpose of this standard is to enable complete interoperability of applications using WAVE in a manner that simplifies the onboard vehicle systems, reducing cost and improving performance. Effective use of the memory pages by applications can also minimize configuration management issues over the life of a system.

This standard is intended to enable a wide range of applications to be supported by an OBU of the lowest possible cost. The low cost is enabled by removing the need for the OBU to interpret application messages. There is no OBU software representing applications using RM; thus the processing, memory, and configuration management requirements are removed from the OBU. Instead of putting such processing requirements on the OBU, they are placed on the RSU or an application processor remote from the RSU. The only processing requirement is that of interpreting the specific commands and message headers defined herein, which is application independent. The OBU merely serves as a mobile mailbox to carry application messages and data from one RSU to another or as a common interface point to transfer data to other onboard systems.

By allowing memory to be assigned to an application at any time during the life of the OBU, future applications can be developed and deployed without onboard hardware or software modification.

For applications using RM as a mobile mailbox, with no onboard use of the data stored in memory, there are significant security advantages. By having the OBU treat each application's messages as a bit-stream to be saved and later retrieved from memory, such data can be encrypted in a manner that is not known to the OBU. There is no need for the OBU to support the encryption schemes used by these applications, and such security schemes can be under the total and absolute control of each of these applications.

2. Normative references

The following referenced documents are indispensable for the application of this standard. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

IEEE P1609.3™, Draft Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)—Networking Services.[3, 4, 5]

IEEE Std 1609.2™, IEEE Trial-Use Standard for Wireless Access in Vehicular Environments—Security Services for Applications and Management Messages.

IETF RFC 768, User Datagram Protocol.[6]

---

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

[2]Notes in text, tables, and figures are given for information only and do not contain requirements needed to implement this standard.

[3]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[4]Numbers preceded by P are IEEE authorized standards projects that were not approved by the IEEE-SA Standards Board at the time this publication went to press. For information about obtaining drafts, contact the IEEE.

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

[6]Internet RFCs are available from the Internet Engineering Task Force (http://www.ietf.org/).

3. Definitions, abbreviations, and acronyms

For the purposes of this standard, the following terms and definitions apply. The glossary in Annex D and *The Authoritative Dictionary of IEEE Standards Terms* [B1][7] should be referenced for terms not defined in this clause.

3.1 Definitions

3.1.1 application class identifier (ACID): A code that identifies a class of applications.

NOTE—See IEEE P1609.3.

3.1.2 application context mark (ACM): A code that identifies a specific instance of an application within a class.

NOTE—See IEEE P1609.3.

3.1.3 application protocol data unit (APDU): Packets of information transmitted over the communication media between the RMA and RM by the application service layer.

NOTE—See protocol data unit (PDU) in Annex D.

3.1.4 onboard unit (OBU): A wireless access in vehicular environments (WAVE) device that can operate when in motion and supports information exchange with roadside units (RSUs) and other OBUs.

NOTE—See onboard equipment (OBE) in Annex D.

3.1.5 provider: The part of an application that provides services.

3.1.6 provider service table (PST): The collection of data describing the applications that are registered with and available though a wireless access in vehicular environments (WAVE) device, with supporting channel information.

3.1.7 response to provider service table (RPST): A data structure returned to the wireless access in vehicular environments (WAVE) application from an onboard unit (OBU), upon receipt of a provider service table (PST) identifying a resource hosted by the OBU.

3.1.8 roadside unit (RSU): A wireless access in vehicular environments (WAVE) device that operates only when stationary and supports information exchange with onboard units (OBUs).

NOTE—See roadside equipment (RSE) in Annex D.

3.1.9 wireless access in vehicular environments (WAVE) device: A device that contains a WAVE-conformant medium access control (MAC) and physical layer (PHY) interface to the wireless medium.

NOTE—See IEEE P802.11p™ [B2].

3.2 Abbreviations and acronyms

ACID application class identifier

ACM application context mark

APDU application protocol data unit

ASDU application service data unit

[7]The numbers in brackets correspond to the numbers of the bibliography in Annex E.

ASN.1	Abstract Syntax Notation One
CVO	commercial vehicle operations
HEX	hexadecimal
IP	Internet Protocol
ITS	intelligent transportation systems
MIB	management information base
MSB	most significant bit
OBU	onboard unit
PDU	protocol data unit
PST	provider service table
RCP	resource command processor
RF	radio frequency
RM	resource manager
RMA	resource manager application
RPST	response to provider service table
RSU	roadside unit
SAP	service access point
SDU	service data unit
UDP	User Datagram Protocol (IETF RFC 768)
UI	user interface
WAVE	wireless access in vehicular environments
WBSS	WAVE basic service set
WME	WAVE management entity (defined in IEEE P1609.3)

4. Architecture and communications flow summary

4.1 General

This standard describes one specific WAVE application consisting of the RM and its partner, the RCP, that provide services to a remote entity, an RMA. The RM (or provider) relays commands received from the RMA to the RCP (or user). In turn, the RCP executes the commands it receives and returns responses to the RMA via the RM. Whereas the RCP resides on an OBU, the RM may reside on an RSU or OBU.

Implemented as a WAVE application from the perspective of IEEE P1609.3, the RM provides services that allow RMA to access the memory and UIs within the OBU as well as interfaces to other onboard equipment controlled by the RCP. As such, the RM acts as if it were an application layer to an RMA.

In general, RMAs communicate with one or more RMs over a secure wired network, whereas an RM communicates with an RCP over the wireless link on which the security may be lacking. The RM multiplexes the communication sessions of multiple RMAs in a manner that allows each RMA to communicate end-to-end with RCPs as illustrated in Figure 1. For simplicity, this figure shows the RM hosted by an RSU as part of the roadside environment, but the RM may also be hosted by an OBU, a scenario not shown in this figure.

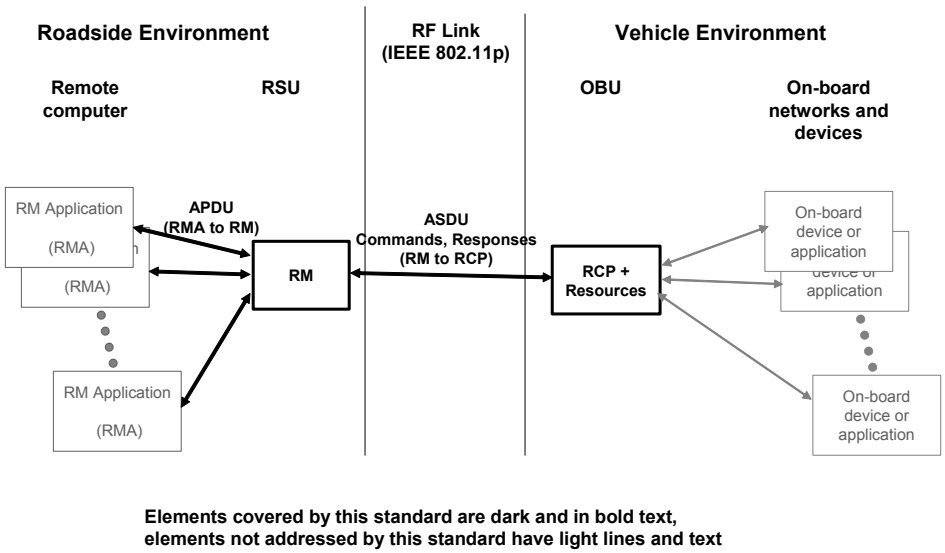

Figure 1—Components addressed in this standard

In general, each RMA uses a set of commands (executable by each RCP) to

- Store information in, and retrieve information from, OBUs
- Request the use of UI resources on OBUs (not to be confused with the vehicle's UIs)
- Access specific networks interfaced to OBUs
- Interrogate optional security devices hosted by OBUs

This standard refers to "managing resources" hosted by OBUs, where these resources consist of, but are not limited to,

- Read/write memory
- UIs that are included as part of the OBU
- Specialized interfaces to other onboard equipment
- Optional vehicle security devices interfaced to the OBU

This standard specifies

- The services provided by the RM to the RMA
- How the services provided by IEEE P1609.3 are used to advertise the presence of the RM and RMAs to OBUs
- How the RCP recognizes and responds to the presence of the RM and each of its associated RMAs to complete an application process or transaction
- The managed memory-based resources hosted by OBUs and how they are used to store and retrieve information and control OBU UIs and other equipment interfaces
- The command set available to RMAs to manage these resources, how these commands and their responses are interchanged between RMAs and the RM and RCPs over a secure WAVE radio frequency (RF) link
- The use of specialized read/write memory resources that allow the transfer of data to other equipment interfaced to the OBU and controlled by the RCP

4.2 Application component references

This standard governs the

— Managed resources hosted on the OBU (Clause 5)

— Commands and responses used by the RMAs to manage these resources (Clause 6)

— Management services and data transfer services provided to the RMAs by the RM (Clause 7)

— Interface to the lower layer services (Clause 8)

4.3 WAVE RMA data transfer and management services

As illustrated in Figure 1, the RM provides services to applications known as RMAs. These services, provided through a standardized interoperable interface, allow the implementation of low-latency applications.

An RMA, in its simplest form, retrieves data in blocks of memory that reside on the OBU. These blocks of memory are known as *pages* and are part of the OBU RCP resources.

All remote access to pages is via a secured RF link. Some pages are accessible to all RMAs; others are registered to a specific RMA and accessible only to that RMA. Such registration is in accordance with Annex B. Registered pages are allocated to a specific RMA across a set of OBUs and are considered resources accessible to the RMA to which they are registered. The management is done using the set of commands specified in Clause 6. Some of the pages are defined so that storing data on, and retrieving data from, them performs the management of UIs that are part of, or connected to, the OBU. These pages are said to be *memory-mapped* to the UI. Some pages are defined so that they act as a communication buffer between the OBU and onboard equipment or network interfaced to the OBU. These pages are known as *transfer pages*. Writing data to transfer pages initiates the transfer of these data to the associated equipment or network. As well, the equipment or network associated with the interface may write data onto this or another page allowing these data to be read by the associated RMA to complete the interchange.

This standard, by defining a command set that provides access to pages and by specifying that each resource be mapped to a page, allows any RMA, by the use of the command set, to manage in a standardized and interoperable manner all OBU RCP resources to which it has access.

Figure 1 illustrates that there is a two-stage process involved in the interchange of commands and responses. One stage is from RMA to RM via defined application protocol data unit (APDU); the other is from RM to RCP through a set of commands and responses. Commands, encapsulated and transmitted by the RMA in APDUs over a network, are received by the RM. Upon receipt, the RM extracts the commands and encapsulates them in User Datagram Protocol (UDP) packets that it transmits, via the RF link, to the specified OBU's RCP. Upon receipt, the RCP executes the commands and returns the appropriate responses to the RM, encapsulated in UDP packets. The RM, in turn, extracts and reencapsulates the responses in APDUs for transmission to the RMA. The APDUs are described in Clause 7. The commands and responses are described in Clause 6.

4.4 Data structure representation

Data interchanged between the RM and RCP over the RF link are encapsulated and transmitted between the RMAs and RM in packets of information known as *application protocol data units* (APDUs). Data interchanged between layers of the protocol stack within the RSU or within the OBU are transferred as *application service data units* (ASDUs).

4.4.1 Protocol data units (PDUs)

PDUs are data transmitted in both directions across the communication media and are said to be *visible*. The open, visible transmission requires that all implementations create PDUs that adhere to a specified structure. Abstract Syntax Notation One (ASN.1) is used to define the structure of each PDU and each component within the PDU. Packed encoding rules, aligned, (PER-Aligned) are used to convert ASN.1-defined structures to transmission format. The convention used in this standard for ASN.1 structure is as follows:

```
<ComponentStructureType>::= SEQUENCE
    {
    <descriptive-tag1>              <SubComponentStructureType1>,
    <descriptive-tag2>              <SubComponentStructureType2>,
    etc.
    }
```

4.4.2 Service data units (SDUs)

SDUs specify parameters to services provided by one protocol layer to another in the same device. Generally SDUs are implemented by a software mechanism such as a function call. SDUs are used to pass information across a software interface known as a *service access point* (SAP). As such, SDUs are said to be not visible, and the data fields may be transferred between internal layers in an implementation-specific manner. To distinguish SDUs from PDUs, this standard will adopt the following (non-ASN.1) format for the specification of an SDU:

```
<SDU-NAME<.request>|<.confirmation>|<.indication>|<.response>>
    ( <ASN.1 parameter1-type>       <parameter1-name>,
    <ASN.1 parameter2-type>         <parameter2-name>,
    etc.
    )
```

All services have four basic forms: request, indication, response, and confirmation. These services may be used in confirmed mode, nonconfirmed mode, and locally confirmed mode. Figure 2, Figure 3, and Figure 4 illustrate the concepts.

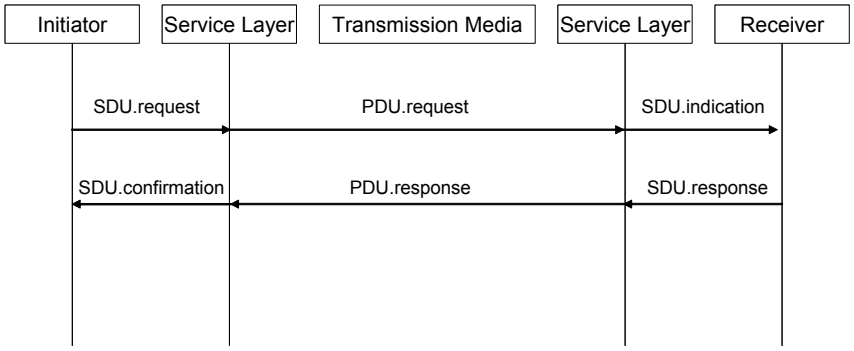

Figure 2—Services in confirmed mode

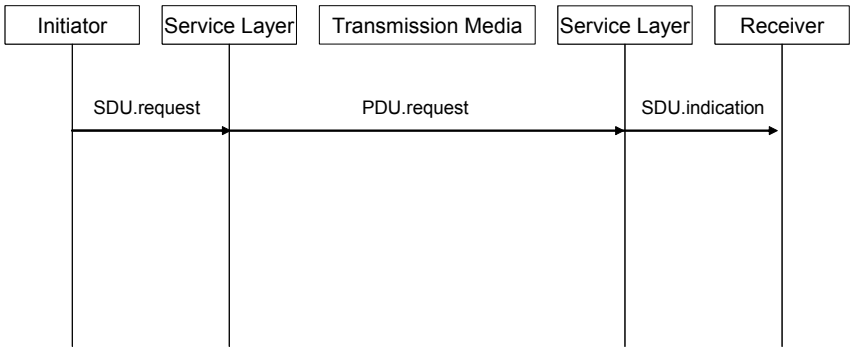

Figure 3—Services in nonconfirmed mode

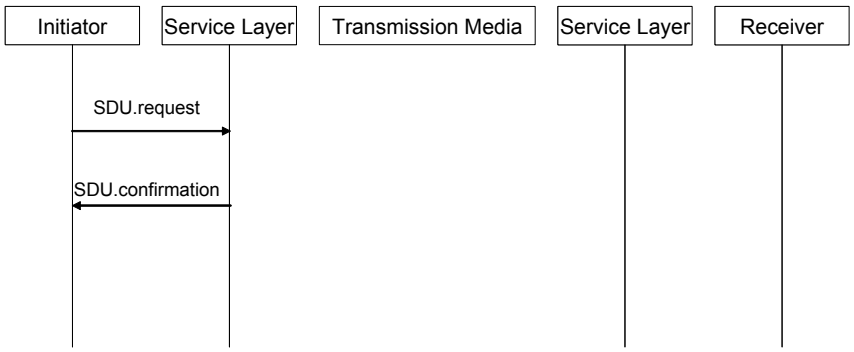

Figure 4—Services in locally confirmed mode

4.5 Summary of operation

The sequence diagrams of Figure 5 and Figure 6 use two approaches to showing the basic end-to-end flow of the application's data interchange. The details are described in subsequent clauses of this standard or in related standards. A simplified explanation of the operations shown in Figure 5 and Figure 6 are found at the end of this clause. Specific names of data units and structures will be described in subsequent clauses or in related standards.

A summary of the operation is as follows:

— Each RMA shall be registered with the RM with which it will be interacting. This registration includes a list of the OBU RCP resources to which access is desired.

— A list of the resources of interest to the RMA is incorporated into the application context mark (ACM). The RM creates the ACM in accordance with the ASN.1 definitions of Annex A (RM-ApplicationContextMark). There is an upper bound on the size of the ACM, described in IEEE P1609.3, which may not be exceeded.

— Each RMA, when ready to begin processing transactions, requests that it become "active" (see 7.4.1). In the activation, it specifies the OBU RCP resources in which it has interest (i.e., the pages to which it requires access). The RMA also specifies in the activation an optional auto-command sequence that the RM executes on its behalf upon receipt of a response to provider service table (RPST).

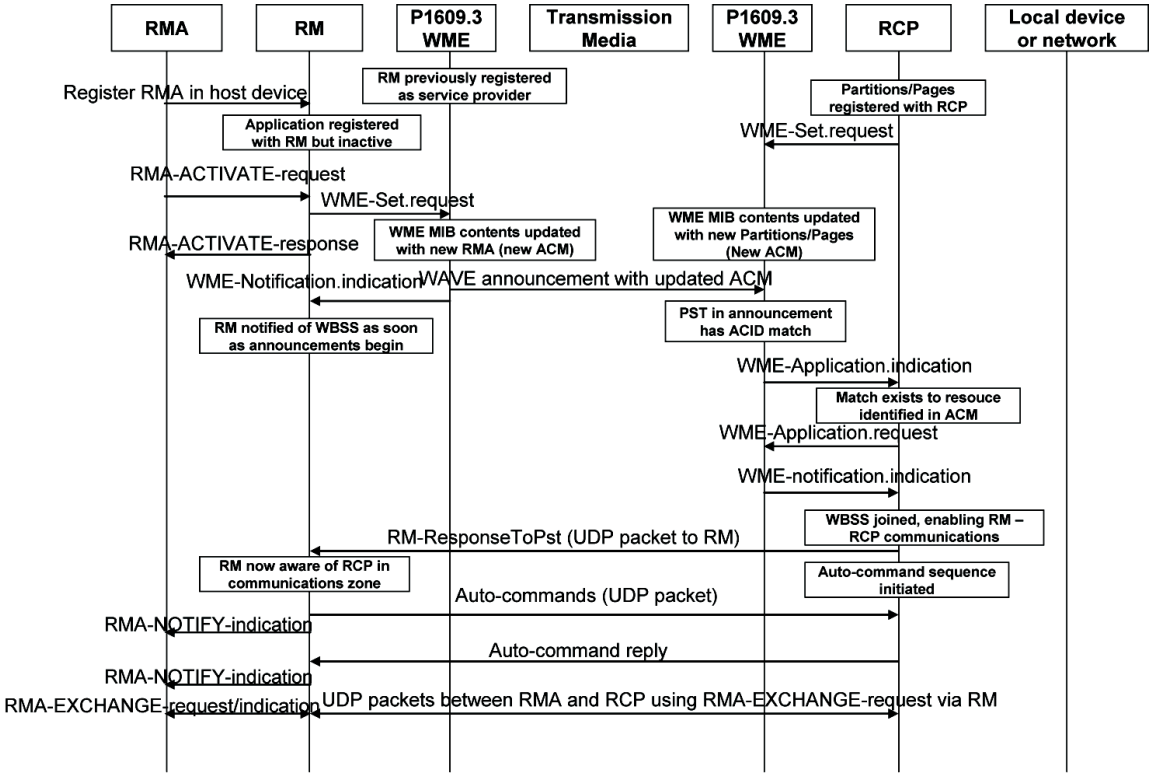

Figure 5—Data flow and management update operations in RMA/RM/RCP communication

— The RM and RCP (being WAVE applications) register with the WAVE management entity (WME) as specified in IEEE P1609.3. The RM registers so that it is incorporated into the application list sent in the provider service table (PST) that includes its ACM created above. The RCP on the OBU registers with the WME as a user application.

— Upon receipt of a PST by the WME on the OBU, the RCP is notified that it is in the presence of an RM and receives, with the notification, access to the contents of the ACM.

— The RCP scans the ACM for a match on any resource that it hosts. If there is a match, it generates a RPST and, after requesting that the WME join the WAVE basic service set (WBSS, defined in IEEE P802.11p [B2] and IEEE P1609.3), transmits it directly to the provider RM.

— Upon receipt of this RPST, the RM notifies each associated RMA that it is in the presence of an OBU RCP hosting resources to which it has access, after the possible transmission of an auto-command sequence (see 7.4.1.3, 7.4.3.4, and 8.7) registered by the RMA.

— After this interchange is complete, the RMA, using the services of the RM, may request that commands be transmitted to the RCP using the exchange request service as specified in 7.5.1.

— As responses to these commands are received, the RM relays them in exchange responses to the correct RMA.

— The session is then formally terminated at the application level. The RMA terminates the session by transmitting a Sleep Transaction command, described in 6.4.5, to the RCP and a terminate session indication, described in 7.4.5, to the RM on the RSU.

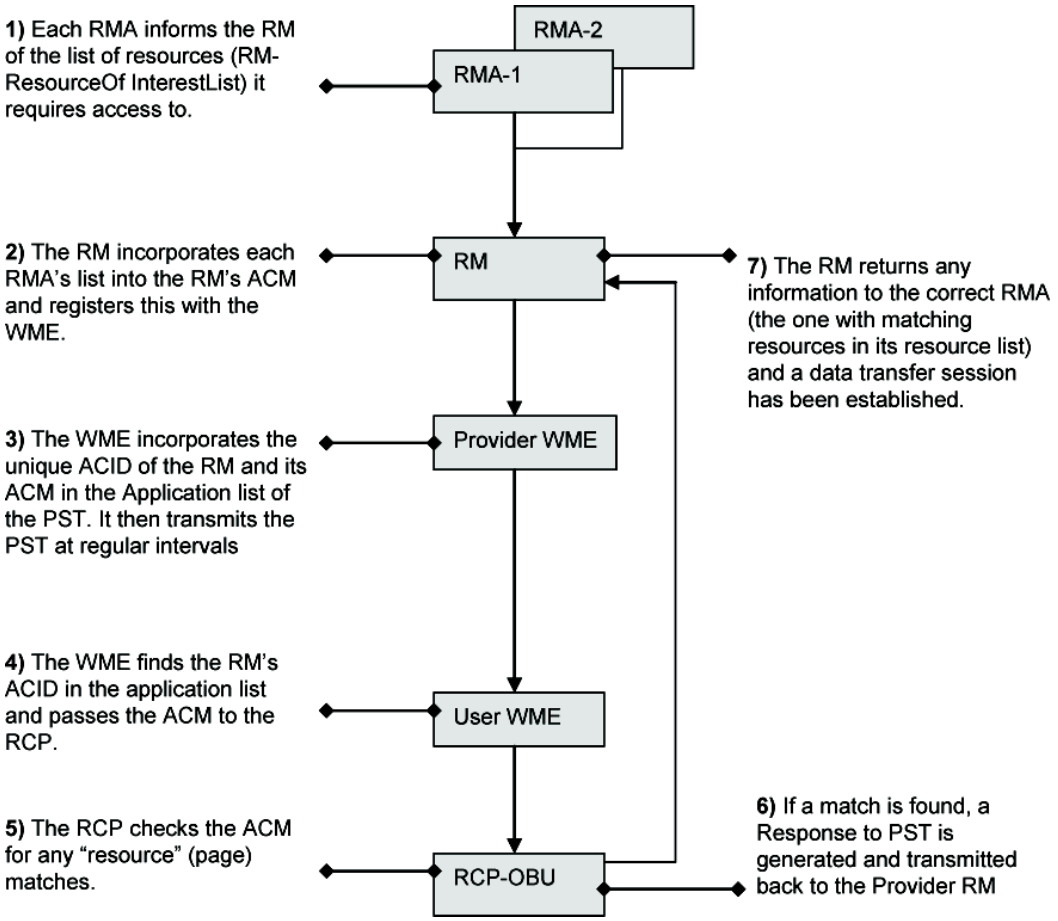

Figure 6—Steps in establishing a data transfer session between RMA and RCP

5. OBU resources

5.1 General

All OBUs will include memory in accordance with 5.2, but may also include additional resources such as UIs, equipment and network interfaces, and security devices. All resources provided shall be capable of being controlled by the RCP using the commands defined in Clause 6.

5.2 Memory

WAVE devices conforming to this standard provide memory, subdivided into addressable blocks known as *pages*. These pages are used by RMAs to store (write) and retrieve (read) data using the RM command set defined in Clause 6.

Pages in this standard are of the following types:

— Storage pages (general purpose read/write)

— Memory-mapped pages (e.g., UI)

— Transfer pages (to/from external interfaces)

When a page is reserved using the Reserve Memory command described in 6.4.6, the page type is specified by the enumerated value RM-PageType (defined in Annex A). Page types “storage” (type 0), “mapped” (type 1), and “transfer” (type 2) may use either an application-controlled format or the RM-Message format data structures. Page types “storageInsert” (type 128), “mappedInsert” (type 129), and “transferInsert” (type 130) shall use the RM-Message format data structure defined in 5.4.

Memory on the OBU that is accessible to RMAs shall have the following properties:

— It shall be capable of being subdivided into partitions (having the unique identifier “RM-partition” in Annex A) and addressable pages within each partition (having the unique identifier “RM-Page” in Annex A).

— The size in bytes of each partition shall be specified by a 16-bit integer (see 6.4.8), limiting them to 64k bytes.

— The size in bytes of each page shall be specified by a 16-bit integer (see 6.4.6), limiting them to 64k bytes. This size is further limited by the fact that several pages may be contained within a single partition.

— Page 0 in partition 0 shall be reserved for implementation-specific usage, and attempts to access its content shall be considered invalid.

— Partition 0 shall be mandatory, whereas the provision for defining other partitions is optional.

— An RMA may reserve, via an offline registration process, pages within the mandatory partition 0 or, if supported, a partition within which it may allocate pages used for its own specific purposes. This assures that each OBU that has been provisioned by any supplier acts as a virtual private device to multiple RMAs. The current allocation of pages in partition 0, and partitions, is found in Annex B.

— Memory-mapped pages, if provided, shall have the capabilities described in 5.2.2.

— Transfer pages, if provided, shall have the capabilities described in 5.2.3.

All partitions and all pages are identified by a 16-bit identifier. The currently defined identifiers for partitions and pages are provided in Annex B.

5.2.1 Storage pages

The most general type of page is the application-controlled storage page. A page of this type is used by an RMA to store and retrieve data. Usually, the implementor of an RMA requests that a population of OBUs be manufactured within which their specific page is reserved. This may be done by the manufacturer, using the Reserve Memory Page command, or at a facility operated by the implementor of the RMA. Subsequently, the RMA uses the services of the RM to store and retrieve data using its page. The data are stored in a format specific to the RMA and are not modified in any way by the RM or RCP.

5.2.2 Memory-mapped pages

The OBU may provide any combination of the UI devices described in this subclause. The availability of these devices accessible to an RMA shall be specified in a data field described in 8.5.2. If present, these UI devices shall be controlled using the Set User Interface command specified in 6.4.4. UIs are controlled by writing data to, or reading from, special memory pages referred to as *memory-mapped pages*. These memory-mapped pages act as buffers to the specific UI associated with each such page. The data written to one of these pages shall be used when a Set User Interface command (6.4.4) is given to transfer these data to the associated UI. In the case of data coming from the UI, such as a keypad, the data written to this page may be retrieved by a Read Memory Page command (6.4.1) to transfer these data to the RMA associated with this page. Memory-mapped pages are preassigned the identities in Table B.1.

5.2.3 Transfer pages

Pages designated as transfer pages may be provided as a means of interfacing with onboard equipment and networks. These pages shall be associated with a specific external interface controlled by the RCP if the interface is present. If an authorized RMA writes data to one of these pages, the data shall be transferred to the associated interface. The details of the transfer mechanism are implementation specific. The format of the data stored is defined when the specific application is designed. A means of storing data received from interfaced onboard equipment or networks onto these transfer pages may be provided to create a two-way communication path to the RMA.

5.3 User interfaces (UIs)

5.3.1 Visual displays

The OBU may provide visual displays such as colored light emitting diodes (LEDs). There may be up to three, with each being a different color. The acceptable colors are red, green, and yellow or their closest equivalent depending on the display technology.

5.3.2 Buzzers (audible alert)

The OBU may provide a buzzer used to alert the user with an audible signal.

5.3.3 Enunciators

The OBU may provide an enunciator. Enunciators are capable of simulating speech or music whereas a buzzer is limited to on/off fixed tones. The enunciator shall be used to present messages that have been written to the corresponding memory-mapped page (see Note 5 and Note 6 in 6.4.4).

5.3.4 Character readout

The OBU may provide a character readout. The character readout shall be used to display text messages that have been written to the corresponding memory-mapped page (see Note 5 and Note 6 in 6.4.4).

5.3.5 Keypad

The OBU may provide a keypad. The keystrokes that are entered using this keypad shall be inserted as a message into the corresponding memory-mapped page. This page may then be read by authorized applications allowing them access to the data that were entered.

5.3.6 Future UI resources

Future revisions of this standard may provide for additional UI resources. The extensibility of the OBU Configuration field (8.5.2) allows for the definition of bits that indicate the presence of new resources.

5.4 Data formats for memory resources

If interoperable applications are to share information with each other via the storage and retrieval of data, the format and content of the data shall be defined by a standardized message format (the RMA message format defined in this subclause). When interoperability with other applications is not required, this standard permits data to be in any format, including proprietary, and what is standardized is the means of storing and retrieving the data (see Clause 6).

Page types that support the Insert Message command defined in 6.4.3 shall make use of the RM-Message format. The RM-Message format enables the maintenance of lists of messages on these pages.

RM-Message format consists of a header and message contents following the ASN.1 description in Annex A. Each message priority shall be in the range of 0 to 3 where messages of priority 0 are of highest priority, or most urgent. (If the priority is specified as being from 4 to 255, it is treated as being priority 3). The assignment of message priorities is implemented through a registration mechanism outside the scope of this standard.

The Message Expiry field provides a means for the OBU RCP to remove messages that are no longer valid, by releasing the memory they occupy to be used to store newer messages. To extend the range of the expiry interval, the Message Expiry field is interpreted as shown in Table 1. This format provides the option to specify that messages expire in from 1 second to 192 days from the time of receipt of the message.

Table 1—Message Expiry field

Value modifier		**Interpretation of values (0 to 64) for each "value modifier"**					
bit 7	6	5	4	3	2	1	0
0x00		Equal to the time in seconds message is valid			1 to 64 s	0 = 1 s, 63 = 64 s	
0x01		Equal to the time in minutes message is valid			1 to 64 min	0 = 1 min, 63 = 64 min	
0x02		Equal to the time in hours message is valid			1 h to 64 h	0 = 1 h, 63 = 64 h	
0x03		Equal to the time in 3-day units message is valid			3 to 192 days	0 = 3 days, 63 = 192 days	

If there is not enough memory to store a new message on the requested page, the RCP shall delete the oldest message of lowest priority until there is sufficient memory to store the message. If this process fails, the error response Insufficient Memory will be returned to the RMA, and the messages marked for deletion are restored. The RCP shall never delete an unexpired message of the same or higher priority as the one that is to be inserted. The onus is on the RMA to allocate enough memory for this type of page in the Reserve Memory Page command to accommodate all of the messages that it estimates will be required to exist concurrently on the page.

6. RM commands and responses

6.1 Overview

The RCP shall provide the capability to process the mandatory commands specified in this clause. These commands reference memory partitions and pages that may be hosted by the OBU RCP.

All UIs, security devices, and onboard interfaces managed by the RM are mapped to a reserved page in partition 0. As such, the management of these resources is reduced to the proper use of these commands to store information on and retrieve information from these pages. Some of the commands described in this clause are mandatory, and some are optional. For each command, there is a defined response. The responses include values by which the RCP notifies the RMA that an optional command, although received, has not been implemented in this device.

Throughout this standard, all multi-byte fields with numerical significance are to be encoded most significant byte first and most significant bit (MSB) within each byte.

6.2 Command format

Each command consists of the four fields shown in Table 2. The individual commands are described in 6.4.1 through 6.4.9. All commands and command fields are transmitted most significant byte first and MSB within each byte.

Table 2—Command fields

bit 7 (MSB)	6	5	4	3	2	1	0
Reserved 1 bit	Command Identifier 7 bits						
No Response Indicator 1 bit	Command Transaction Identifier 7 bits						
Command Parameter Length 2 bytes (specifies the total size in bytes of the parameter fields that follow)							
Command Parameter — (may consist of multiple parameters) Size variable — equal to value in Command Parameter Length field							

With this structure as the basic unit, commands shall be chained together in a command sequence. A single command shall be transmitted as a command sequence containing one command. The command processor shall execute each of the commands in the order they appear in the sequence and generate a response sequence as described in 6.3. The format of a command sequence is shown in Table 3.

Table 3—Command sequence

Number of commands (1 byte)	*N* = 1–255
First command	Reserved bit
	Command 1 identifier
	No Response Indicator bit
	Command 1 transaction identifier
	Command 1 parameter length
	Command 1 parameter(s)
Second command	Reserved bit
	Command 2 identifier
	No Response Indicator bit
	Command 2 transaction identifier
	Command 2 parameter length
	Command 2 parameter(s)
Repeat for *N* commands	

Descriptions of the command fields are given in 6.2.1 through 6.2.5.

6.2.1 Command Identifier field

Field size

The size of the Command Identifier field is 7 bits.

Usage

The Command Identifier field identifies the command to be executed and shall be identified by the coded command identifiers shown in Table 4. A full description of each command is found in 6.4.

Table 4—Command identifiers

Command identifier	Command name	Implementation requirements and usage
hex(00)	Reserved	Reserved for future assignment.
hex(01.. 0F)	Available for new commands	Reserved for future assignment.
hex(10)	Read Memory Page	Mandatory Used to retrieve data previously stored on a page.
hex(11)	Write Memory Page	Mandatory only if read/write memory is present Used to store data on a page.
hex(12)	Insert Message	Mandatory only if read/write memory is present Used to insert messages, in priority sequence, into the list on a page that supports message inserts.
hex(13.. 1F)	Reserved	Reserved for future assignment.
hex(20)	Set User Interface	Mandatory only if any UI is present Used to control action of UI.
hex(21.. 2F)	Reserved	Reserved for future assignment.
hex(30)	Sleep Transaction	Mandatory Used to signal a pause or end of transaction to the RCP.
hex(31.. 3F)	Reserved	
hex(40)	Reserve Memory Page	Mandatory only if read/write memory is present Used to create a new page in a specified partition.
hex(41)	Release Memory Page	Mandatory only if Reserve Memory Page command is implemented Used to remove a page and release its memory.
hex(42)	Reserved	
hex(43)	Reserve Partition	Optional Used to create a new partition on the OBU. Partition 0 is mandatory and preallocated.

Table 4—Command identifiers *(continued)*

Command identifier	Command name	Implementation requirements and usage
hex(44)	Release Partition	Mandatory only if Reserve Partition command is implemented Used to remove a partition and release its memory.
hex(45..6F)	Reserved	
hex(70..7F)	Available for manufacturer-specific testing	Optional Shall not be used in production units deployed in the field.

6.2.2 No Response Indicator bit

Field size

The size of the No Response Indicator bit is 1 bit.

Usage

— If set to zero, the No Response Indicator bit shall indicate to the RCP that the appropriate response to the transaction shall be returned.

— If set to one, the No Response Indicator bit shall indicate to the RCP that a response to this command shall not be returned.

NOTE—It is normal to request a response to assure that the command was received and the action taken. A means of suppressing the response is provided but should be used with care.

6.2.3 Command Transaction Identifier field

Field size

The size of the Command Transaction Identifier field is 7 bits.

Usage

The Command Transaction Identifier field shall be set by the RMA for each command transmitted to an RCP as a means of uniquely identifying this instance of a command. This identifier shall be returned unaltered in the response to that command. As such, the RMA can use this field in any manner that allows it to associate the response with the original command. For example, an incremental wraparound counter may be used.

6.2.4 Command Parameter Length field

Field size

The size of the Command Parameter Length field is 2 bytes.

Usage

The Command Parameter Length field shall be used by the RMA to specify the total length (in bytes) of the Command Parameter field that follows.

NOTE—The Command Parameter Length field allows the RCP to extract the parameters to the current command and then move to the next command in the sequence.

6.2.5 Command Parameter field

Field size

The length of the Command Parameter field is dependent on the command and shall be equal to the value specified in the Command Parameter Length field.

Usage

The Command Parameter field shall be used by the RMA to pass the information required by the RCP to execute the command.

6.3 Response format

The RCP processes commands within a sequence of commands up to the first command that is found to be invalid. For each valid command up to the first invalid command, plus this first invalid command, the RCP generates a response if the No Response Indicator bit is 0. All commands after the first invalid command are ignored.

The response to an individual command is formatted as shown in Table 5. If multiple commands are being responded to at the same time (as a sequence), this sequence of responses shall be formatted as in Table 3.

Table 5—Response fields

bit 7 (MSB)	6	5	4	3	2	1	0
Reserved 1 bit	Command Identifier 7 bits						
Unused 1 bit	Command Transaction Identifier 7 bits						
Response Status 1 byte							
Response Length 2 bytes (specified the total size in bytes of the response data that follow)							
Response Data Size variable							

NOTE—Just as commands are received in command sequences, responses to command sequences are returned in response sequences. Each of the commands in the sequence is responded to if requested, but the responses need not be returned in the same order as the commands were received. The RMA uses the Command Identifier and Command Transaction Identifier pair to associate responses with specific commands.

6.3.1 Command Identifier field

Field size

The size of the Command Identifier field is 7 bits.

Usage

The Command Identifier field shall contain the command identifier received in the original command sequence.

Default value

None

6.3.2 Command Transaction Identifier field

Field size

The length of the Command Transaction Identifier field shall be 7 bits.

Usage

The Command Transaction Identifier field shall contain the command transaction identifier received in the original command (see 6.2.3).

Default value

None

6.3.3 Response Status field

Field size

The length of the Response Status field shall be 1 byte.

Usage

The Response Status field shall contain a value indicating the status of command execution by the RCP (see Table 6). If a command failure is encountered, the process may stop returning responses only up to the command in error.

Default value

None

Response Status definitions

Table 6 lists the valid values for the Response Status field and the interpretation of each. A response to a command shall contain valid response data only when the Response Status field is set to Command Success. All other values shall indicate a failure condition.

Table 6—Response Status values

Response name	Value	Specification and description
Reserved	hex(0)	Reserved.
Command Success	hex(1)	Execution completed successfully.
Command Failed	hex(2)	The command failed due to some unspecified condition, such as a parameter error.
Command Not Recognized	hex(3)	The command identifier is invalid.
Command Not Supported	hex(4)	The command is not supported by the RCP.
Page Not Defined	hex(5)	The page identifier does not match a page hosted by the RCP.
Partition Not Defined	hex(6)	The partition identifier does not match a partition hosted by the RCP.
Device Error	hex(7)	A malfunction has occurred in the RCP hardware or software.
Memory Access Error	hex(8)	The requested memory could not be accessed due to an undefined internal error.
Page Length Mismatch	hex(9)	The length of the memory image is greater than the length of the referenced page.
Insufficient Memory	hex(A)	Available free memory is insufficient to perform the command.
Commands Not Executed	hex(B)	Due to internal memory limits, none of the command sequence was executed. Either the entire sequence is retransmitted or, due to encountering an invalid command in sequence, the processing was aborted.
Command Sequence Error	hex(C)	Unable to process command sequence due to sequence formatting error. (Typically the known parameter length is incorrect.)
Page Type Mismatch	hex(D)	Request to write data to a message page or request to insert a message on a page that does not support messages.
Page Already Exists	hex(E)	Trying to reserve a page that was previously reserved.
Partition Already Exists	hex(F)	Trying to reserve a partition that was previously reserved.
Unauthorized	hex (10)	Access was denied due to lack of proper authorization.
Write Error	hex(11)	Attempted write to a read-only page.
Nonexistent	hex (12)	Attempt to release a partition or page that does not exist.
Reserved	hex(13.. AF)	Reserved.
Vendor Area	hex(BF.. FF)	Available for vendor-specific failure conditions.

6.3.4 Response Length field

Field size

The size of the Response Length field is 2 bytes if the response is to a Read Memory Page command. For all other commands, this field is not present.

Usage

The Response Length field shall specify the total length (in bytes) of the data contained in the Response Data field.

Default value

None

6.3.5 Response Data field

Field size

For the current command set, the Response Data field shall appear only if it is a successful response to a Read Memory Page command. In this case, it is variable in length and defined by the value in the Response Length field.

Usage

If:

The command is	Read Memory Page
and the response status is	Command Success

The Response Data field contains the requested page image.

Default value

None

6.4 Command definitions

This subclause describes in detail the commands found in Table 4. For each command, the fields described in Table 2 and specific to that command appear under the **Command definition** headers. Each command is used by an RMA to perform an action to which the RCP (if requested) returns a response.

All fields that contain a size are encoded as an unsigned integer value with the most significant byte transmitted first.

6.4.1 Read Memory Page command

The Read Memory Page command is used to request that the memory image of the specified page be returned in the response. The No Response Indicator bit shall not be set for this command.

Command definition

The Read Memory Page command shall contain the fields shown in Table 7.

Table 7—Read Memory Page command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(10)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Value = 8 Total length in bytes of the command parameter fields that follow
Partition Identifier	2	Partition that contains the page
Page Identifier	2	Page within the specified partition
Page Offset	2	Byte at which to start
Number	2	The number of bytes to return

RCP normal behaviors

If the partition and page are hosted by the RCP, the response shall consist of the following:

— Response Status field set to Command Success
— Response Length field set to the length of the Response Data field
— Response Data field containing the part of the page's memory image that was requested

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.2 Write Memory Page command

The Write Memory Page command is used to request that the specified page be overwritten with the memory image that appears in the parameter list.

Command definition

The Write Memory Page command shall contain the fields shown in Table 8.

Table 8—Write Memory Page command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(11)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow
Partition Identifier	2	Partition that contains the page
Page Identifier	2	Page within specified partition
Page Offset	2	Byte at which to start
Number	2	The number of bytes to write
Memory Image[a]	Variable	The part of the image of the page to be written

[a]Note that the RF interface may not allow large data files, and such situations require fragmentation at the lower layers.

RCP normal behaviors

If the partition and page are hosted by the RCP, the section specified by the Page Offset field and Number field shall be completely overwritten with the memory image provided as parameter.

If this is a transfer page, then the local process for transferring these data to the local network/equipment associated with this page shall be notified. This is a locally unique process that is outside the scope of this standard.

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.3 Insert Message command

The Insert Message command is used to insert a message into a list of messages on a single page that has been reserved to support the insert message function. Such a page may thus contain multiple messages rather than a single one as in a basic read/write memory page. See Table 9 for Insert Message command fields. The Insert Message command shall not be used for pages other than RM-PageType “storageInsert” (type 128), “mappedInsert” (type 129), and “transferInsert” (type 130) pages.

Command definition

The Insert Message command shall contain the fields shown in Table 9.

Table 9—Insert Message command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(12)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow
Partition Identifier	2	Partition that contains the page
Page Identifier	2	Page within specified partition
Message Image	Variable	The message to append to the page

RCP normal behaviors

If the partition and page are hosted by the RCP, the RCP shall insert the message image command parameter into the list of messages on the page.

The RCP shall use the priority of the message and its expiry date to insert the message into the list of messages currently on the page, first ordered with respect to priority (highest priority first) and then, within each priority, ordered with respect to expiry time (soonest to expire first). (See 7.4 for details.)

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.4 Set User Interface command

The Set User Interface command is used to control the operation of UIs, the presence of which is specified in the OBU RCP Configuration field is defined in Table 19.

Command definition

The Set User Interface command shall contain the fields shown in Table 10.

Table 10—Set User Interface command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(20)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow
Number (of repeated UI control structures in this command)	1	N = 0–255 The number of times the structure from User Interface Element field to Attributes field is repeated in this single command
RMA Priority	1	This value is used to determine which UI settings take precedence if multiple Set User Interface commands are received. If multiple commands are received with the same RMA priority, the first one received takes precedence. Priority values are defined in 5.4.
User Interface Element	2	This structure affects each of the elements indicated by the bit mask defined in Table 19
Control Type	1	If - AbsoluteOff (0) If - AbsoluteOn (1) If - Timed (on) (2) If - Flashing Cycle (3) Reserved (4.. 255)
Attributes (depending on control type) If - Absolute If - Timed If - Flashing (2 fields) 1st 2nd	 0 2 4 (32 bits) 1	 (Not present for Absolute control type) Time period, in 125 ms increments, for which the elements are to be set to the “on” state Cycle state bit map, 125 ms per bit Repetition count (1.. 255)
(repeated for *N* instances of UI control structure)		
User Interface Element		
Control Type		
Attributes		

RCP normal behaviors

The RCP shall alter the specified UI elements, depending upon the value of the type and attribute parameters. The actions specified by the Control Type field and Attributes field are as follows:

- The AbsoluteOff control type shall turn the addressed UI element "off." The "off" state shall be maintained until changed by a subsequent command.
- The AbsoluteOn control type shall turn the addressed UI element "on." The "on" state shall be maintained until changed by a subsequent command.
- The Timed control type shall turn the addressed UI element "on" for a specified period of time and then "off" until changed by a subsequent command. The time period shall be interpreted as the number of 125 ms increments.
- The Flashing control type shall cycle the state of the addressed UI element based upon a 4-byte (32 bit) bit map. A one indicates "on," a zero "off."
  - Each bit in the map shall represent an interval of 125 ms (thus the total bit map represents 32 × [1/8] s = 4 s).
  - The repetition byte shall indicate the number of times (0 to 255) the bit map cycle pattern shall be performed.

NOTE—Following is an example of flashing parameters:

[11111111000000001111111100000000, 20] is interpreted as follows:
1 s on, 1 s off, 1 s on, 1 s off, repeated 20 times. (Total 80 s of flashing at 1 Hz.)

The bits in the 32-bit sequence are processed from the left to right of the representation above.

The bits are received in the following byte order in the command:
11111111 left-most bits of 32-bit sequence represented above: left-most bit processed first
00000000
11111111
00000000 right-most bits of 32-bit sequence represented above: left-most bit processed first

NOTE 1—An implementation that provides any colored display elements shall provide elements of colors as close to red, green, and yellow as possible for the given technology it uses.

NOTE 2—The RCP shall perform the action specified for each instance of the repeated control structure in the order they appear in the list.

NOTE 3—The completion of the specified UI control specifications shall not be affected by the reception of any other transmissions from any RM except for those with overriding RMA priority.

NOTE 4—If two display elements are to be set independently, (e.g., green solid, yellow flashing), a single UI command specifying two separate UI control structures may be used.

NOTE 5—The control of the character readout UI and of the enunciator UI is performed with two separate commands. The commands may be received in the same command sequence. The first one is a Write Page command, specifying the message to be displayed or enunciated, and the second one is a Set User Interface command, specifying that the character readout or enunciator be activated as specified in the UI control structure.

NOTE 6—A page has been allocated for use by the Set User Interface command. Its identifier is hex(FF03), and it is designated for use as [UI image storage]. This page is provided for convenience, and it is used to store the last image of the controls that are in effect for each UI.

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.5 Sleep Transaction command

The Sleep Transaction command is used to inform the receiving OBU that either

— The session is complete and no more commands will be sent in this communication session, or

— The session is to be paused and resumed at a specified interval in the future.

Command definition

The Sleep Transaction command shall contain the fields shown in Table 11.

Table 11—Sleep Transaction command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(30)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow (Value = hex(00 01) for this command)
Pause Interval	1	The time in 125 ms intervals that this application shall wait before resuming the session. Zero indicates that the session is complete.

RCP normal behaviors

If the Pause Interval field value is zero, it indicates that the transaction is complete. This command uses the WME-Application.request services of IEEE P1609.3 to inform the WME that it is inactive.

If the Pause Interval field value is between zero and hex(FF), it is an indication to the RCP that it shall temporarily stop transmitting data to the RM for the requested pause interval after it has returned the response to the command.

If the Pause Interval field value is hex(FF), the current pause shall be aborted.

The RCP is responsible for timing the interval and resuming the session.

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.6 Reserve Memory Page command

The Reserve Memory Page command is used to allocate a page of memory from the free memory within a specified partition. Such a reservation represents a permanent allocation until it has been released by a Release Memory Page command.

NOTE—This command shall be transmitted alone, that is, in a sequence of 1 command. This assures a quiescent state is reached before any other commands that reference this page are processed.

Command definition

The Reserve Memory Page command shall contain the fields shown in Table 12.

Table 12—Reserve Memory Page command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(40)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow
Partition Identifier	2	Partition in which the page is to be reserved
Page Identifier	2	Identifier for the new page
Page Size	2	Length (in bytes) of this page
Page Type	1	ASN.1 type RM-PageType (see Annex A). This field specifies if the page type is storage, transfer, or memory-mapped and whether it is to support Insert Message command.

RCP normal behaviors

If there is sufficient memory in the specified partition's pool of available memory and the page has not previously been reserved, the RCP shall create the page within the partition.

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.7 Release Memory Page command

The Release Memory Page command is used to release the memory occupied by a page that had previously been reserved to free memory within the specified partition.

Command definition

The Release Memory Page command shall contain the fields shown in Table 13.

Table 13—Release Memory Page command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(41)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow
Partition Identifier	2	Partition for this page
Page Identifier	2	Identifier for the page

RCP normal behaviors

If the partition and page exist, the RCP shall release the memory occupied by the page to the partition's available memory.

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.8 Reserve Partition command

The Reserve Partition command is used to allocate a partition from the OBU's available free memory. Such a reservation represents a permanent allocation until it has been released by a Release Partition command.

NOTE—This command shall be transmitted alone, that is, in a sequence of 1 command. This assures a quiescent state is reached before any other commands that reference this page are processed.

Command definition

The Reserve Partition command shall contain the fields shown in Table 14.

Table 14—Reserve Partition command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(43)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow
Partition Identifier	2	Partition identifier to be used
Partition Size	2	Length (in bytes) of the requested partition

RCP normal behaviors

If there is sufficient memory on the OBU's pool of available memory and the partition does not currently exist, the RCP shall allocate the partition.

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

6.4.9 Release Partition command

The Release Partition command is used to release the memory occupied by a partition that had previously been reserved to the OBU's available free memory.

Command definition

The Release Partition command shall contain the fields shown in Table 15.

Table 15—Release Partition command fields

Command field	Length (bytes)	Description
Command Identifier	1	hex(44)
Command Transaction Identifier	1	(RMA controlled)
Command Parameter Length	2	Total length of the command parameter fields that follow
Partition Identifier	2	Partition identifier to be released

RCP normal behaviors

If the partition exists, the RCP shall return the memory it occupied to the OBU's pool of available memory.

If requested, the RCP shall return a response with the Response Status field set to Command Success.

RCP abnormal responses

If a response is requested and the RCP is unable to process the command, it shall return in the Response Status field the appropriate value defined in Table 6.

7. Services provided by the RM to the RMA

7.1 General

As stated, the WAVE RM, although implemented like any WAVE application, acts as an application layer that provides services to WAVE RMAs to

— Activate and deactivate

— When active, make their presence known to OBUs that host RCP resources to which they have access

— Exchange commands and responses with the WAVE RCP on OBUs that respond

— Terminate an existing data exchange session

As reference:

— The “services” used by the RM to make the presence of an RMA known are found in Clause 7.

— The “resources” that an RMA manages are found in Clause 5.

— The “commands” that an RMA uses to manage the resources and the required “responses” are found in Clause 6.

The interface described in this clause is conducted by the interchange of APDUs described in this clause between RMAs and RM, typically over a secured Internet Protocol (IP) network. The flow of APDUs between RMAs and RM is depicted in Figure 5 (in 4.5).

NOTE 1—If conducted over an IP network, the activation request described in this clause would have to be addressed to a known UDP or Transmission Control Protocol (TCP) port. This port is currently not reserved.

NOTE 2—In some cases, an RMA may be co-resident with the RM in the provider device. In these cases, the protocol between the RMA and RM is not conducted over a network. Instead it is implemented across a SAP within the provider device. In these cases, it would make use of the ASDUs described for each service.

7.2 RMA-RM interface service elements

This subclause details the differences between APDU and ASDU.

Services are conducted by the exchange of requests and responses and by the exchange of indications and confirmations. Data units that are transmitted as part of the protocol are known as *protocol data units* (PDUs), whereas data units exchanged across a software interface are known as *service data units* (SDUs).

7.2.1 PDUs—RMA APDU

WAVE RMA APDUs are interchanged between a remote RMA and the RM on the provider device via a secure network connection. Being transmitted over a “visible” interface, their format is formally described by an ASN.1 type definition. The base ASN.1 type is RMA-APDU (defined in Annex A).

7.2.2 SDUs—RMA ASDU

SAPs are abstractions of an interface between implementation-specific software modules residing within a device. They are described in a notation that resembles a software interface. In this clause, for each formal definition of an APDU, there exists a matching ASDU to be used if the RMA resides on the same device as the RM.

7.2.3 PDU and SDU naming convention

Each service is given a service name. Subsequently, "<service>" is to be replaced by the associated service name. For example,

RMA-<SERVICE>-Request

represents an APDU described in ASN.1, transmitted over a "visible" interface to invoke a service, while

RMA-<service>.request

represents the associated ASDU, implemented internally, that is used to invoke the same service.

7.3 Protocol services

Protocol services are subdivided into

— Management services, used to perform administration of the interface (see 7.4)
— Data transfer services, used to exchange information across the communications link (see 7.5)

The ASN.1 type RMA-APDU specifies the APDUs that are provided. It consists of a choice of the services listed in Table 16.

Table 16—RMA-RM APDU

RMA-ACTIVATE-Request
RMA-ACTIVATE-Response
RMA-NOTIFY-Indication
RMA-NOTIFY-Confirmation
RMA-TERMINATESESSION-Indication
RMA-TERMINATESESSION-Confirmation
RMA-EXCHANGE-Request
RMA-EXCHANGE-Response
RMA-EXCHANGE-Confirmation
RMA-DEACTIVATE-Request
RMA-DEACTIVATE-Response

NOTE—In 7.4 and 7.5, the APDU types are found in Annex A.

7.4 Protocol management services

7.4.1 Activate request service

When an RMA whose resource(s) of interest have been incorporated into the ACM of the RM[8] is ready to start communicating with RCPs, it shall inform the RM via an activation request formatted as follows:

APDU:

RMA-ACTIVATE-Request

ASDU:

```
Status = RMA-activate.request
    (RM-ID                          rm-ID
    , RM-ResourceOfInterestList     rm-ResourceOfInterestList
    , RM-CommandSequence            rm-AutoCommandSequence
    )
```

Descriptions of the fields are given in 7.4.1.1 through 7.4.1.3.

7.4.1.1 rm-ID field

The rm-ID field is a structure containing a 16-bit value that uniquely defines the RMA and an 8-bit value specifying its priority. The ASN.1 type RM-ID defines its structure.

7.4.1.2 rm-ResourceOfInterestList field

The rm-ResourceOfInterestList field contains the identification of resources, hosted by an OBU RCP, that the RMA has the privilege to access. Since all resources are mapped to specific pages, each unique partition-page identifier defines a resource, and the rm-ResourceOfInterestList consists of partition-page pairs and the page access capability requested.

An RMA may require access to, and an RCP may host, multiple resources. The ASN.1 type RM-ResourceOfInterestList is used to identify the resources that are of interest to the RMA.

The PST that is transmitted contains a certificate (see IEEE Std 1609.2), signed by a certificate authority, and "installed"[9] on the provider device. Accessible to the RM on the provider device[10] is the complete list of resources accessible to each RMA. At the time of activation, an RMA shall specify the list of resources (rm-ResourceOfInterestList) to which it requires access. (This may be a subset of the resources it has the privilege to access.) The RM shall validate that the list is consistent with what has been certified as being accessible to the RMA. If there is a mismatch, the entire request shall be ignored.

7.4.1.3 rm-AutoCommandSequence field

Each RMA may include in its registration a properly formatted command sequence as described in 6.2. The RM shall verify the commands in the rm-AutoCommandSequence field to assure that it contains no commands that are incorrectly formatted or that attempt to access resources that the RMA does not have the privilege to access. If the auto-command sequence received is empty (i.e., has number of commands in sequence equal to zero), it shall be an indication that the RMA wishes to be immediately notified when an RCP hosting one of its resources of interest responds to the PST.

[8]Refer to Figure 5.

[9]Refer to IEEE Std 1609.2 for the details of this procedure.

[10]This is possible via the IEEE 1609.2 services.

NOTE—If an RCP responds to the PST indicating that it hosts resources of interest to multiple RMAs, although feasible, no attempt will be made to amalgamate the auto-command sequences from the multiple RMAs into a unified auto-command sequence. Instead, the RM will issue each auto-command sequence separately in the order of RMA priority or, if the RMAs are of the same priority, in an implementation-specific order.

7.4.2 Activate response service

To complete the activation process, a response to the activation shall be returned by the RM to the originator of the request via an activation response formatted as follows:

APDU:

RMA-ACTIVATE-response

ASDU:

```
Status = RMA-activate.response
    (RM-ConnectionID            rm-ConnectionID
    , RM-ActivationStatus       rm-ActivationStatus
    )
```

Descriptions of the fields are given in 7.4.2.1 and 7.4.2.2.

7.4.2.1 rm-ConnectionID field

The rm-ConnectionID field value shall be assigned by the RM. Its usage is internal to the RM.

7.4.2.2 rm-ActivationStatus field

The rm-ActivationStatus field shall be used to return the status of the request. The only information returned in the response will be an indication of success. If there is an error in the APDU, the request shall be ignored. If a valid response is lost in transmission, the RMA shall repeat the request and eventually receive the response with successful status.

7.4.3 Notify indication service

After an RMA has activated, it waits for notifications from the RM of potential communication partners. The notification service shall be provided as a means of indicating to an RMA that an OBU RCP hosting at least one resource to which it has access has entered the provider device's communication zone. The format of this service is as follows:

APDU:

RMA-NOTIFY-Indication

ASDU:

```
RMA-notify.indication
    (RM-ConnectionID            rm-ConnectionID
    , RM-LinkIdentifier         rm-LinkID
    , RM-ResourceElementList    rm-Resources
    , RM-ResponseSequence       rm-AutoResponseSequence
    )
```

Descriptions of the fields are given in 7.4.3.1 through 7.4.3.4.

7.4.3.1 rm-ConnectionID field

The rm-ConnectionID field shall contain a copy of the value received in the RMA-ACTIVATE-Response.

7.4.3.2 rm-LinkID field

The rm-LinkID field, passed to the RMA, shall contain an internal value that is subsequently used by the RM on the provider device to identify the session just established. The RM shall use this value to identify each of the concurrent WAVE communication sessions, in all subsequent services.

7.4.3.3 rm-Resources field

The rm-Resources field shall contain a list of the resources that are hosted by the receiving OBU and of interest to the particular RMA.

NOTE—The RPST contains the complete list of resources hosted by the RCP that are of interest to all RMAs being serviced by the RM. The RM shall reformat the list into sub-lists, each properly formatted into an RM-ResourceElementList, that pertain only to a particular RMA. These sub-lists are returned to their associated RMA in the notify indication.

7.4.3.4 rm-AutoResponseSequence field

The rm-AutoResponseSequence field shall contain the sequence of responses to the auto-command sequence received by the RM from the RCP.

7.4.4 Notify confirmation service

Upon receipt of an RMA-NOTIFY-Indication, the RMA shall confirm its receipt by returning an RMA-NOTIFY-Confirmation. The format of this service is as follows:

APDU:

RMA-NOTIFY-Confirmation

ASDU:

```
RMA-notify.confirmation
    (RM-ConnectionID            rm-ConnectionID
    , RM-LinkIdentifier         rm-LinkID
    )
```

Descriptions of the fields are given in 7.4.4.1 and 7.4.4.2.

7.4.4.1 rm-ConnectionID field

The rm-ConnectionID field shall contain a copy of the value received in the RMA-ACTIVATE-Response.

7.4.4.2 rm-LinkID field

The rm-LinkID field shall contain an internal value that is subsequently used by the RM on the provider device to identify the session just established. The RM shall use this value to identify each of the concurrent WAVE communication sessions in all subsequent services.

7.4.5 Terminate session indication service

The RMA shall inform the RM when a transaction identified by a link identifier has been completed via a terminate indication. The format of this service is as follows:

APDU:

RMA-TERMINATESESSION-Indication

ASDU:

```
RMA-terminateSession.indication
    (RM-ConnectionIdentifier        rm-ConnectionID
    , RM-LinkIdentifier             rm-LinkID
    , RM-ID                         rm-ID
    )
```

Descriptions of the fields are given in 7.4.5.1 through 7.4.5.3.

7.4.5.1 rm-ConnectionID field

The rm-ConnectionID field shall contain the value returned to the RMA in the RMA-ACTIVATE-Response.

7.4.5.2 rm-LinkID field

The rm-LinkID field shall contain the value returned to the RMA in the RMA-NOTIFY-Indication.

7.4.5.3 rm-ID field

The rm-ID field is a structure containing a 16-bit number that uniquely defines the WAVE RMA and an 8-bit number specifying its priority. The ASN.1 type RM-ID defines it.

7.4.6 Terminate session confirmation service

The RM confirms reception of a terminate session indication by returning a terminate session confirmation. The format of this service is as follows:

APDU:

RMA-TERMINATESESSION-Confirmation

ASDU:

```
RMA-terminateSession.confirmation
    (RM-ConnectionIdentifier        rm-ConnectionID
    , RM-LinkIdentifier             rm-LinkID
    , RM-ID                         rm-ID
    )
```

Descriptions of the fields are given in 7.4.6.1 through 7.4.6.3.

7.4.6.1 rm-ConnectionID field

The rm-ConnectionID field shall contain the value returned to the RMA in the RMA-ACTIVATE-Response.

7.4.6.2 rm-LinkID field

The rm-LinkID field shall contain the value returned to the RMA in the RMA-NOTIFY-Indication.

7.4.6.3 rm-ID field

The rm-ID field is a structure containing a 16-bit number that uniquely defines the WAVE RMA and an 8-bit number specifying its priority. The ASN.1 type RM-ID defines it.

7.4.7 Deactivate request service

When the RMA wishes to stop operation for any reason, it shall inform the RM via a deactivate request. The format of this service is as follows:

APDU:

RMA-DEACTIVATE-Request

ASDU:

```
Status = RMA-deactivate.request
    (RM-ConnectionIdentifier        rm-ConnectionID
    , RM-ID                         rm-ID
    )
```

NOTE—Deactivation is equivalent to a suspension of operation. To resume service, a new activation request is made.

Descriptions of the fields are given in 7.4.7.1 and 7.4.7.2.

7.4.7.1 rm-ConnectionID field

The rm-ConnectionID field shall contain the value returned to the RMA in the RMA-ACTIVATE-Response.

7.4.7.2 rm-ID field

The rm-ID field is a structure containing a 16-bit number that uniquely defines the WAVE RMA and an 8-bit number specifying its priority. The ASN.1 type RM-ID defines it.

7.4.8 Deactivate response service

Upon receipt of a deactivate request from an RMA that is currently active, the RM shall return as confirmation a deactivate response. The format of this service is as follows:

APDU:

RMA-DEACTIVATE-Response

ASDU:

```
Status = RMA-deactivate.request
    (RM-ActivationStatus            rm-DeactivateStatus
    )
```

A description of the field is given in 7.4.8.1.

7.4.8.1 rm-DeactivateStatus field

The rm-DeactivateStatus field shall be used to return the status of the request only if the RMA specified by the rm-ID field is currently active. If there is in an error in the APDU, the request shall be ignored. If a valid response is lost in transmission, the RMA shall repeat the request and eventually receive the response with successful status.

7.5 Protocol data transfer services

After an application communication session has been established, the RMA may send RM command sequences to the RCP. This shall be done using the exchange services.

7.5.1 Exchange request service

An RMA may send a command sequence to the RCP using the exchange request service. The format of this service is as follows:

APDU:

RMA-EXCHANGE-Request

ASDU:

```
RMA-exchange.request
    (RM-ConnectionIdentifier        rm-ConnectionID
    , RM-LinkIdentifier             rm-LinkID
    , RM-CommandSequence            rm-CommandSequenceToSend
    )
```

Descriptions of the fields are given in 7.5.1.1 through 7.5.1.3.

7.5.1.1 rm-ConnectionID field

The rm-ConnectionID field shall contain the value returned to the RMA in the RMA-ACTIVATE-Response.

7.5.1.2 rm-LinkID field

The rm-LinkID field shall contain the value returned to the RMA in the RMA-NOTIFY-Indication.

7.5.1.3 rm-CommandSequenceToSend field

The rm-CommandSequenceToSend field shall contain a properly formatted command sequence to be sent to the RCP on the OBU. Before transmission of this sequence to the RCP, the RM shall analyze this sequence to assure that the RMA has the privilege to access the resources it addresses in this sequence and that the syntax of each command is valid. If an error is encountered, the request is ignored.

7.5.2 Exchange response service

After a command sequence has been transmitted to the RCP and the response sequence has been received, the RM shall forward it to the RMA via an exchange response. The format of this service is as follows:

APDU:

RMA-EXCHANGE-Response

ASDU:

```
RMA-exchange.response
    (RM-ConnectionIdentifier        rm-ConnectionID
    , RM-LinkIdentifier             rm-LinkID
    , RM-ResponseSequence           rm-ResponseSequenceReceived
    )
```

Descriptions of the fields are given in 7.5.2.1 through 7.5.2.3.

7.5.2.1 rm-ConnectionID field

The rm-ConnectionID field shall contain the value returned to the RMA in the RMA-ACTIVATE-Response.

7.5.2.2 rm-LinkID field

The rm-LinkID field shall contain the value returned to the RMA in the RMA-NOTIFY-Indication.

7.5.2.3 rm-ResponseSequenceReceived field

The rm-ResponseSequenceReceived field shall contain the sequence of responses to the commands returned from the RCP.

7.5.3 Exchange confirmation service

After the RMA has received an exchange response from the RM, it shall confirm its reception by returning to the RM an exchange confirmation. The format of this service is as follows:

APDU:

```
RMA-EXCHANGE-Confirmation
```

ASDU:

```
RMA-exchange.confirmation
    (RM-ConnectionIdentifier        rm-ConnectionID
    , RM-LinkIdentifier             rm-LinkID
    )
```

Descriptions of the fields are given in 7.5.3.1 and 7.5.3.2.

7.5.3.1 rm-ConnectionID field

The rm-ConnectionID field shall contain the value returned to the RMA in the RMA-ACTIVATE-Response.

7.5.3.2 rm-LinkID field

The rm-LinkID field shall contain the value returned to the RMA in the RMA-NOTIFY-Indication.

8. Interface to services used by RM

8.1 General

RM is implemented as a WAVE application that uses services provided by the

— WME, as described in IEEE P1609.3
— Security services, as described in IEEE Std 1609.2
— UDP, as described in IETF RFC 768

This clause reviews the services of which an implementor of this standard makes use and provides a general description of the interface to these services. The reader requires knowledge of the referenced standards.

NOTE—Throughout this clause, the phrase *page of interest* is considered synonymous with the term *resource*, and the two are used interchangeably.

8.2 OBU information not accessible using standard command set

Information not stored on registered pages is not accessible via the RM command set. Each OBU may contain manufacturing information that is accessible only by a proprietary and secure mechanism. Storage formats and change/retrieval mechanisms of this type of information are not part of this standard.

8.3 Registration of RM and RCP

IEEE P1609.3 provides the WME-ApplicationRegistration.request service primitive: This service allows an application to register and thereby have its presence advertised to OBUs.

On the provider device, the RM registers as application class identifier (ACID) = "resource-manager" and provides its ACM. The ACM contains a list of all the resources that are of interest to all of the currently “active” RMAs that have registered with the RM and, as a result, contains the access privileges of each RMA.

The structure of the ACM is defined in Annex A as RM-ApplicationContextMark.

On the OBU, the RCP registers as ACID = "resource-manager". In the registration, it requests that the WME pass the ACM to it for processing and wait for a request to join the WBSS to access the provider device.

8.4 WME notification to RCP

As described in IEEE P1609.3, when a match of the ACID is made by the WME on the OBU, the associated application, in this case the RCP, is notified of the presence of the provider device and is passed the associated ACM extracted from the PST. The RCP, from the contents of the ACM, then determines whether it hosts any of the resources that appear in the RM-ResourceOfInterestList field (ASN.1 definition in Annex A).

If there is any resource match, the RCP prepares an RPST as defined in 8.5.

8.5 OBU information returned in the response to PST by RCP

The following data are not accessible via the RM command set, but they shall be returned to the RM in the RPST (ASN.1 definition in Annex A). In turn, they shall be passed to the RMA by the RM.

The RPST structure contains the following:

— The OBU RCP configuration information (Table 17)

— A list of images of the requested pages of interest (a sequence of RM-ResourceElement defined in Annex A) that are hosted by the OBU RCP and are to be returned in the RPST

— A list of pages of interest (RM-ResourceList defined in Annex A) that are hosted by the OBU RCP but not returned in the RPST

To create the RPST, the RCP, starting from the first entry in the resources of interest list, takes the following actions:

— First, selects all of the page images that it hosts whose rm-PageAccess field specifies that they are to be returned in the RPST and that fit into the maximum sized response packet, and creates the RM-ResourceElementList (ASN.1 definition in Annex A).

— Then, creates the rm-UnsentElements field, consisting of a list of pages that are not to be returned in the RPST plus all of the pages that were to be returned but could not be fit into the maximum length response.

After creating the RPST, the RCP requests that the WME establish an infrastructure link with the provider device. IEEE P1609.3 provides the WME-Application.request service primitive for this purpose. When the link is established, signaled by the reception of a WME-Notification.indication (defined in IEEE P1609.3), the RM-ResponseToPst structure is returned directly to the RM on the provider device using the UDP services that are now available.

Table 17—OBU information fields

Field name	Specification and description
Memory Configuration	Defines the availability of various memory regions.
OBU Configuration	Defines the availability of various OBU peripherals, such as UI and enunciators. For future expansion, the OBU Configuration field is defined as an extensible field allowing for the definition of new resources.
Maximum Application Data Block	Some devices may have a limitation on the maximum block of data they can handle received in an APDU. The RM is made aware of this limit via the value of this field.

8.5.1 Memory Configuration field

The Memory Configuration field shall be used to indicate which read/write memory regions are present. Each bit shall indicate the presence or absence of a supported feature. Five bits are set aside for possible future feature support. Values shall be interpreted as defined in Table 18.

Table 18—Interpretation of Memory Configuration field

Location (bits)	Interpretation (1 present, 0 not present)
7 (MSB)	Multiple partition support
6	Commercial vehicle operations (CVO) file transfer page
5	Vehicle Information page
4	Reserved for future use
3	Reserved for future use
2	Reserved for future use
1	Reserved for future use
0 (LSB)	Reserved for future use

8.5.2 OBU Configuration field

The OBU Configuration field shall be used to indicate the configuration of installed UIs. Each bit indicates the presence or absence of a UI. The field is extensible by the use of bit 7 in each byte being set to 1 if another byte follows. The bits reserved for future use will be defined as new resources are standardized.

Table 19—OBU Configuration field

Byte 1 - bitfields	Interpretation
7 (MSB)	Extensibility indicator = 0 if only 1 byte; 1 if byte 2 present
6	Red display element
5	Green display element
4	Yellow (or similar) display element
3	Character readout
2	Keypad
1	Buzzer
0 (LSB)	Enunciator
Byte 2 - bitfields	(if byte 2 present)
7 (MSB)	extensibility indicator = 0 if only 2 bytes; 1 if byte 3 is present
6	Security Device (future)
5	Reserved for future use
4	Reserved for future use
3	Reserved for future use
2	Reserved for future use
1	Reserved for future use
7 (MSB)	Reserved for future use

8.5.3 Maximum Application Data Block field

The Maximum Application Data Block field shall be used to indicate to the RM and the RMA the maximum size of the data portion of commands sent to the RCP. This information is necessary when large pages are defined on an OBU and the OBU has limited internal buffering space. A mechanism is provided in the command set to set up an end-to-end fragmentation scheme. (This mechanism is addressed in 6.4.1 and 6.4.2 via the Page Offset field and Number field.)

A 16-bit integer is used to specify the value, allowing values from 0 to 65 535 bytes. The provider sets the value to the minimum of what it and the user can accommodate.

8.6 Reception of RPST by RM

At the provider device, the reception of the RM-ResponseToPst (ASN.1 definition in Annex A) by the RM is the RM's notification that an OBU has entered the communication zone and that it hosts at least one page of interest (resource) that at least one of the RMAs has registered as being of interest. The RM shall first determine which of the RMAs that it will notify.

The received RM-ResourceElementList (ASN.1 definition in Annex A) contains pages of interest for all of the RMAs. The RM shall associate each page image (or indication of the presence of a page) it receives with its corresponding RMA. Then, for each RMA, it shall create a sub-list formatted as an RM-ResourceElementList and return it to the correct RMA via the RMA-NOTIFY-Indication service defined in 7.4.3.

8.7 Auto-command sequence processing

Before sending the RMA-NOTIFY-Indication to the RMA, the RM transmits to the RCP, if present, the optional auto-command sequence that was registered by each RMA. As the UDP link has been established, the auto-command sequences (one for each RMA receiving a resource match) are sent to the UDP port associated with the RCP. This is done in the order of highest priority to lowest priority RMA. If the priorities are the same, the order is implementation specific. (See 6.2 and 6.3 for the structure of command and response sequences.)

The RCP, upon receipt of the auto-command sequence, executes the commands received and returns to the RM the auto-command response sequence.

The RM then forwards the response sequence and the RM-ResourceElementList to the associated RMA using the RMA-NOTIFY-Indication service. The receipt of this notification indicates to the RMA that an OBU to which it has requested access is in the WAVE communication zone.

8.8 Conducting an application session

If required, an application session may then be conducted between the RMA and RCP with the RM acting as a command validator and forwarding agent. The RMA constructs command sequences that it sends to the RM using the RMA-EXCHANGE-Request service defined in 7.5.1.

The RM, upon receipt of the RMA-EXCHANGE-Request, extracts the rm-SendCommandSequence and transmits it to the RCP on the associated OBU. This is done after the command sequence is analyzed by the RM to assure that the syntax of each command is valid and that no command requests access to resources that the RMA does not have privilege to access. If a violation is found, the request is ignored.

Upon receipt of the command sequence, the RCP on the OBU executes the commands and returns to the RM the associated response sequence.

Upon receipt of the packet from the RCP, the RM extracts the response sequence and forms the rm-ReturnResponseSequence, which it transmits to the associated RMA using the RMA-EXCHANGE-Response service defined in 7.5.2.

8.9 Termination of application session

When the RMA decides that the session is complete, it transmits a Sleep Transaction command to the RCP. Upon receipt of the response to this command, or if no response is received in the timeout period, the RMA informs the RM that the session is complete using the RMA-TERMINATESESSION-Indication service.

The RM confirms receipt of the indication with a corresponding confirmation and internally clears its internal resources.

Annex A

(normative)

ASN.1 encoding

IEEE-1609-1 DEFINITIONS::=

BEGIN

```
-- ============================================================
   RMA-APDU::= CHOICE
   { rma-activate-request                [0]     RMA-ACTIVATE-Request
   , rma-activate-response               [1]     RMA-ACTIVATE-Response
   , rma-notify-indication               [2]     RMA-NOTIFY-Indication
   , rma-notify-confirmation             [3]     RMA-NOTIFY-Confirmation
   , rma-terminateSession-indication     [4]     RMA-TERMINATESESSION-Indication
   , rma-terminateSession-confirmation   [5]     RMA-TERMINATESESSION- Confirmation
   , rma-exchange-request                [6]     RMA-EXCHANGE-Request
   , rma-exchange-response               [7]     RMA-EXCHANGE-Response
   , rma-deactivate-request              [8]     RMA-DEACTIVATE-Request
   , rma-deactivate-response             [9]     RMA-DEACTIVATE-Response
   , rma-force-alignment                 [255]   NULL
   }


   RMA-ACTIVATE-Request::= SEQUENCE
   {rm-ID                                RM-ID
   , rm-ResourceList                     RM-ResourceList
   , rm-AutoCommandSequence              RM-CommandSequence
   }


   RMA-ACTIVATE-Response::= SEQUENCE
   {rm-ConnectionID                      RM-ConnectionIdentifier
   , rm-ActivationStatus                 RM-ActivationStatus
   }


   RMA-DEACTIVATE-Request::= SEQUENCE
   {rm-ConnectionID                      RM-ConnectionIdentifier
   , rm-ID                               RM-ID
   }


   RMA-DEACTIVATE-Response::= RM-ActivationStatus


   RMA-EXCHANGE-Request::= SEQUENCE
   {rm-ConnectionID                      RM-ConnectionIdentifier
   , rm-LinkID                           RM-LinkIdentifier
   , rm-SendCommandSequence              RM-CommandSequence
   }
```

```
RMA-EXCHANGE-Response::= SEQUENCE
{rm-ConnectionID                RM-ConnectionIdentifier
, rm-LinkID                     RM-LinkIdentifier
, rm-ReturnResponseSequence     RM-ResponseSequence
}

RMA-EXCHANGE-Confirmation::= SEQUENCE
{rm-ConnectionID                RM-ConnectionIdentifier
, rm-LinkID                     RM-LinkIdentifier
}

RMA-NOTIFY-Confirmation::= SEQUENCE
{rm-ConnectionID                RM-ConnectionIdentifier
, rm-LinkID                     RM-LinkIdentifier
}

RMA-NOTIFY-Indication::= SEQUENCE
{rm-ConnectionID                RM-ConnectionIdentifier
, rm-LinkID                     RM-LinkIdentifier
, rm-Resources                  RM-ResourceElementList
, rm-AutoResponseSequence       RM-ResponseSequence
}

RMA-TERMINATESESSION-Confirmation::= SEQUENCE
{rm-ConnectionID                RM-ConnectionIdentifier
, rm-LinkID                     RM-LinkIdentifier
, rm-ID                         RM-ID
}

RMA-TERMINATESESSION-Indication::= SEQUENCE
{rm-ConnectionID                RM-ConnectionIdentifier
, rm-LinkID                     RM-LinkIdentifier
, rm-ID                         RM-ID
}

RM-ActivationStatus::= INTEGER
{rm-Success                     (0)
, rm-Fill                       (255)
}(0..255)

RM-ApplicationContextMark::= RM-ResourceOfInterestList

RM-CommandSequence::= OCTET STRING

RM-ConnectionIdentifier::= INTEGER(0..65535)--Range:[0-2^16-1]

RM-ID::= SEQUENCE
{rm-AppID                       INTEGER(0..65535)
, rm-AppPriority                INTEGER(0..255)
}
```



```
-- Detailed contents defined in IEEE Std 1455-1999 [B4]
-- encoded as an octet string for storage as RM message


   RM-IEEE1455::= OCTET STRING (SIZE (0..127,...))

   RM-LinkIdentifier::= INTEGER(0..127,...)

   RM-MaximumApplicationDataBlock ::= INTEGER(0..65535)

   RM-Message     ::= SEQUENCE
   {rm-MsgPriority                 RM-MessagePriority
   , rm-MsgExpiry                  RM-MessageExpiry
   , rm-MsgBody                    RM-MessageInformation
   }


   RM-MessageExpiry   ::= INTEGER(0..255)


   RM-MessageInformation::= CHOICE --add choices as required
   {text-ieee1609                  [0]     UTF8String (SIZE(1..127,...))
   , sae                           [1]     OCTET STRING(SIZE(1..127,...))
   , ieee1455-1999                 [2]     RM-IEEE1455
   , rm-MsgProprietary             [3]     RM-MessageProprietary
   , force-Alignment               [255]   NULL
   }



-- additional values may be defined if/as required
-- values 4 to 255 are currently treated as 3.


   RM-MessagePriority  ::= INTEGER
   {rm-priorityUrgent              (0)
   , rm-priorityHigh               (1)
   , rm-priorityMedium             (2)
   , rm-priorityLow                (3)
   , rm-filler                     (255)
   }(0..255)
   RM-MessageProprietary::= OCTET STRING (SIZE(1..127,...))


   RM-OBUConfig::= INTEGER(0..127,...) -- bit-fields


   RM-OBUInformation  ::= SEQUENCE
   {rm-MemoryConfig                RM-OBUMemoryConfig
   , rm-OBUConfig                  RM-OBUConfig
   , rm-MaxAppDataBlock            RM-MaximumApplicationDataBlock
   }


   RM-OBUMemoryConfig::= OCTET STRING (SIZE(1))
```

```
RM-Page::= INTEGER(0..65535)


RM-PageAccess ::= INTEGER
{readWrite                      (0)
, readOnly                      (1)
,rwReturnInRPST                 (2) -- a readWrite page
,roReturnInRPST                 (3) -- a readOnly page
, fill                          (255)
}(0..255)


RM-partition::= INTEGER(0..65535)


RM-PageType     ::= INTEGER
{storage                        (0)
, storageInsert                 (128)
, mapped                        (1)
, mappedInsert                  (129)
, transfer                      (2)
, transferInsert                (130)
, fill                          (255)
}(0..255)


RM-ResourceElement ::= SEQUENCE
{rm-ResourceID                  RM-ResourceID
, rm-ResourceImage              OCTET STRING (SIZE(0..65535))
}
RM-ResourceID    ::= SEQUENCE
{rm-partition                   RM-partition
, rm-Page                       RM-Page
}
RM-ResourceOfInterest::= SEQUENCE
{rm-ResourceID                  RM-ResourceID
, rm-PageAccess                 RM-PageAccess
}


RM-ResourceOfInterestList::= SEQUENCE(SIZE(0..127,...))
                                OF RM-ResourceOfInterest


RM-ResourceElementList::=SEQUENCE
{rm-OBUInformation              RM-OBUInformation
, rm-Elements                   SEQUENCE(SIZE(0..7,...))
                                OF RM-ResourceElement
, rm-UnsentElements             RM-ResourceList
}


RM-ResourceList::= SEQUENCE(SIZE(0..31,...)) OF RM-ResourceID


RM-ResponseSequence::= OCTET STRING
```



```
    RM-ResponseToPst  ::= RM-ResourceElementList


-- ============================================================

END -- of IEEE-1609-1 DEFINITIONS
```

Annex B

(normative)

Registered pages and partitions

The partitions and pages in Table B.1 and Table B.2 have been registered.

Table B.1—Registered page identifiers for partition 0

Page number	Page designation
hex(0.. 00FF)	Reserved
hex(0100)	California Department of Transportation
hex(0101)	Onboard vehicle network - SAE J1708 [B5] formerly reserved by IEEE Std 1455™-1999 [B4]
hex(0102)	Onboard vehicle network - SAE J1939 [B6] formerly reserved by IEEE Std 1455-1999 [B4]
hex(0103.. EFFF)	Available for registration
hex(F000.. FEFF)	Available for device testing before registration
Interface pages	
hex(FF00)	Keypad
hex(FF01)	Enunciator
hex(FF02)	Character readout
hex(FF03)	UI image storage
hex(FF04..FFF0)	Reserved by IEEE for future use
hex(FFF1)	Onboard vehicle network - SAE J1708 [B5]
hex(FFF2)	Onboard vehicle network - SAE J1939 [B6]
hex(FFF3.. FFFD)	Reserved by IEEE for future use
hex(FFFE)	Vehicle Information page
hex(FFFF)	File Transfer page (CVO)

Table B.2—Registered partition identifiers

Partition number	Partition designation
hex(0000)	See Table 8
hex(0001)	Lockheed Martin
hex(0002)	MFS Transportation Systems
hex(0003)	Transcore
hex(0004)	Intermec Technologies, Amtech Systems Division
hex(0005)	Mark IV
hex(0006)	Raytheon
hex(0007)	Sirit
hex(0008.. FE00)	Available for registration
hex(FFFF)	Reserved

Annex C

(normative)

Protocol Implementation Conformance Statement (PICS) proforma[11]

C.1 General

The supplier of a protocol implementation that is claimed to conform to IEEE Std 1609.1 shall complete the following protocol implementation conformance statement (PICS) proforma.

A completed PICS proforma is the PICS for the implementation in question. The PICS is a statement of which capabilities and options of the protocol have been implemented. The PICS can have a number of uses, including use

a) By the protocol implementor, as a checklist to reduce the risk of failure to conform to the standard through oversight;

b) By the supplier and acquirer, or potential acquirer, of the implementation, as a detailed indication of the capabilities of the implementation, stated relative to the common basis for understanding provided by the standard PICS proforma;

c) By the user, or potential user, of the implementation, as a basis for initially checking the possibility of interworking with another implementation (note that, while interworking can never be guaranteed, failure to interwork can often be predicted from incompatible PICS proformas);

d) By a protocol tester, as the basis for selecting appropriate tests against which to assess the claim for conformance of the implementation.

C.2 Abbreviations and special symbols

C.2.1 Symbols for Status column

M	mandatory
O	optional
O.<n>	optional, but support of at least one of the group of options labeled by the same <n> is required
pred:	conditional symbol, including predicate identification

C.3 Instructions for completing the PICS proforma

C.3.1 General structure of the PICS proforma

The first parts of the PICS proforma, Implementation identification and Protocol summary, are to be completed as indicated with the information necessary to identify fully both the supplier and the implementation.

[11]Copyright release for PICS proformas: Users of this standard may freely reproduce the PICS proforma in this annex so that it can be used for its intended purpose and may further publish the completed PICS.

The main part of the PICS proforma is a fixed questionnaire, divided into parts, each containing a number of individual items. Answers to the questionnaire items are to be provided in the rightmost column, either by simply marking an answer to indicate a restricted choice (usually Yes or No) or by entering a value or a set or a range of values.[12]

Each item is identified by an item reference in the first column. The second column contains the question to be answered. The third column contains the reference or references to the material that specifies the item in the main body of this standard. The remaining columns record the status of each item, i.e., whether support is mandatory, optional, or conditional, and provide the space for the answers (see also C.3.4). Marking an item as supported is to be interpreted as a statement that all relevant requirements of the subclauses and normative annexes, cited in the References column for the item, are met by the implementation.

A supplier may also provide, or be required to provide, further information, categorized as either additional information or exception information. When present, each kind of further information is to be provided in a further subclause of items labeled A<I> or X<I>, respectively, for cross-referencing purposes, where <I> is any unambiguous identification for the item (e.g., simply a numeral). There are no other restrictions on its format or presentation.

The PICS proforma for an implementation consists of C.4.1 through C.4.4 corresponding to the network services implemented.

A completed PICS proforma, including any additional information and exception information, is the PICS for the implementation in question.[13]

C.3.2 Additional information

Items of additional information allow a supplier to provide further information intended to assist in the interpretation of the PICS. It is not intended or expected that a large quantity of information will be supplied, and a PICS can be considered complete without any such information. Examples of such additional information might be an outline of the ways in which an (single) implementation can be set up to operate in a variety of environments and configurations or information about aspects of the implementation that are outside the scope of this standard but have a bearing upon the answers to some items.

References to items of additional information may be entered next to any answer in the questionnaire and may be included in items of exception information.

C.3.3 Exception information

It may happen occasionally that a supplier will wish to answer an item with mandatory status (after any conditions have been applied) in a way that conflicts with the indicated requirement. No preprinted answer will be found in the Support column for this. Instead, the supplier shall write the missing answer into the Support column, together with an X<I> reference to an item of exception information, and shall provide the appropriate rationale in the exception information item itself.

An implementation for which an exception information item is required in this way does not conform to this standard.[14]

[12]Note that there are some items where two or more choices from a set of possible answers may apply. All relevant choices are to be marked in these cases.

[13]Where an implementation is capable of being configured in more than one way, a single PICS may be able to describe all such configurations. However, the supplier has the choice of providing more than one PICS, each covering some subset of the implementation's capabilities, if this makes for easier and clearer presentation of the information.

[14]A possible reason for the situation described above is that a defect in this standard has been reported, a correction for which is expected to change the requirement not met by the implementation.

C.3.4 Conditional status

The PICS proforma contains a number of conditional items. These are items for which both the applicability of the item itself, and its status if it does apply (mandatory or optional), are dependent upon whether certain other items are supported.

Where a group of items is subject to the same condition for applicability, a separate preliminary question about the condition appears at the head of the group, with an instruction to skip to a later point in the questionnaire if the N/A answer is selected. Otherwise, individual conditional items are indicated by a conditional symbol in the Status column.

A conditional symbol is of the form <pred>:<N>, where <pred> is a predicate as described below, and <N> is one of the status symbols M or O.

If the value of the predicate is true, the conditional item is applicable, and its status is given by S, the support column is to be completed in the usual way. Otherwise, the conditional item is not relevant, and the N/A answer is to be marked.

A predicate is one of the following:

a) An item-reference for an item in the PICS proforma: The value of the predicate is true if the item is marked as supported, and the value is false otherwise.

b) A boolean expression constructed by combining item-references using the boolean operator OR: The value of the predicate is true if one or more of the items is marked as supported, and the value is false otherwise.

Each item referenced in a predicate, or in a preliminary question for grouped conditional items, is indicated by an asterisk in the Item column.

C.4 PICS proforma—IEEE Std 1609.1

C.4.1 Implementation identification

Supplier	
Contact point for queries about the PICS	
Implementation name(s) and version(s)	
Other information necessary for full identification, e.g., name(s) and version(s) of the machines and/or operating systems(s), system names	

NOTE 1—Only the first three items are required for all implementations. Other information may be completed as appropriate in meeting the requirement for full identification.

NOTE 2—The terms *name* and *version* should be interpreted appropriately to correspond with a supplier's terminology (e.g., type, series, model).

C.4.2 Protocol summary

Identification of protocol standard	IEEE Std 1609.1 Clause 6
Identification of amendments and corrigenda to this PICS proforma that have been completed as part of this PICS	Amd. : Corr. : Amd. : Corr. :
Have any exception items been required? (See C.3.3; the answer Yes means that the implementation does not conform to IEEE Std 1609.1)	☐ Yes ☐ No ☐ Yes ☐ No
Date of statement (dd/mm/yy)	

C.4.3 Top-level architecture

Item	OBU resources memory/UIs	Reference	Status	Conformance
S1	Device functionality	4.1	M	☐ RSU ☐ OBU
S2	RMA accessible Read/Write memory provided	4.1, 5.2	O	☐ Yes ☐ No
S3	UIs included	4.1, 5.2	O	☐ Yes ☐ No
S4	Specialized interfaces via transfer pages exist	4.1, 5.2	O	☐ Yes ☐ No
S5	Vehicle Security devices provided	4.1	O	☐ Yes ☐ No
S6	Size of read/write memory provided for pool of partitions and pages	4.1	S2:M	Kbytes:_____
S7	Partition 0 supported	5.2	M	☐ Yes ☐ No
S8	Partition 0 page 0 reserved and access attempts treated as invalid	5.2	M	☐ Yes ☐ No
S9	Memory partitions (other than 0) supported	5.2	O	☐ Yes ☐ No
S10	Storage pages supported	Clause 5, 5.2, 5.2.1	S2:M	☐ Yes ☐ No
S11	Memory-mapped pages supported	Clause 5, 5.2, 5.2.2	S3:M	☐ Yes ☐ No
S12	Transfer pages supported	Clause 5, 5.2, 5.2.3	S4:M	☐ Yes ☐ No

C.4.4 Normative section:

Item	OBU resources memory/UIs	Reference	Status	Conformance
S13	Visual displays	5.3.1	S11:O	☐ R ☐ G ☐ Y
S14	Buzzer	5.3.2	S11:O	☐ Yes ☐ No
S15	Enunciator	5.3.3	S11:O	☐ Yes ☐ No
S16	Character readout	5.3.4	S11:O	☐ Yes ☐ No ☐ Yes ☐ No
S17	Keypad	5.3.5	S11:O	☐ Yes ☐ No
S18	Other future resource	5.3.6	S11:O	Specify: _____

Item	Data formats for message resources	Reference	Status	Support
S19	RM-Message supported	5.4 O	S28:M	☐ Yes ☐ No
S20	Resource manager (RM) messages comply to ASN.1 definitions	5.4	S19:M	☐ Yes ☐ No
S21	Resource command processor (RCP) removes expired messages and releases memory	5.4	S28:M	☐ Yes ☐ No
S22	RCP deletes messages based on priority & age to make room for new messages	5.4	S28:M	☐ Yes ☐ No
S23	All multi-byte fields most significant byte first and most significant bit (MSB) first in each byte	6.1	M	☐ Yes ☐ No
S24	Command format complied with	6.2	M	☐ Yes ☐ No
S25	Response format complied with	6.3	M	☐ Yes ☐ No

Item	RM command	Reference	Status	Support
S26	Read Memory Page	6.4.1	M	☐ Yes ☐ No
S27	Write Memory Page	6.4.2	S2:M	☐ Yes ☐ No
S28	Insert Message	6.4.3	S2:O	☐ Yes ☐ No
S29	Set User Interface	6.4.4	S3:M	☐ Yes ☐ No
S30	Sleep Transaction	6.4.5	M	☐ Yes ☐ No
S31	Reserve Memory Page	6.4.6	S2:M	☐ Yes ☐ No
S32	Release Memory Page	6.4.7	S31:M	☐ Yes ☐ No
S33	Reserve Partition	6.4.8	S2 & S9:M	☐ Yes ☐ No
S34	Release Partition	6.4.9	S34:M	☐ Yes ☐ No
S35	Response Message	6.4.1–6.4.9	S26 OR: S27 OR: S28 OR: S29 OR: S30 OR: S31 OR: S32 OR: S33 OR: S34:M	☐ Yes ☐ No

Item	RMA service provided	Reference	Status	Conformance
S36	RMA-ACTIVATE-Request	7.4.1	M	☐ Yes ☐ No
S37	RMA-ACTIVATE-Response	7.4.2	S35:M	☐ Yes ☐ No
S38	RMA-NOTIFY-Indication	7.4.3	M	☐ Yes ☐ No
S39	RMA-NOTIFY-Confirmation	7.4.4	S37:M	☐ Yes ☐ No
S40	RMA-TERMINATESESSION-Indication	7.4.5, 8.9	S30:M	☐ Yes ☐ No
S41	RMA-TERMINATESESSION-Confirmation	7.4.6	S39:M	☐ Yes ☐ No
S42	RMA-DEACTIVATE-Request	7.4.7	M	☐ Yes ☐ No
S43	RMA-DEACTIVATE-Response	7.4.8	S41:M	☐ Yes ☐ No
S44	RMA-EXCHANGE-Request	7.5.1	M	☐ Yes ☐ No
S45	RMA-EXCHANGE-Response	7.5.2	S43:M	☐ Yes ☐ No
S46	RMA-EXCHANGE-Confirmation	7.5.3	S43:M	☐ Yes ☐ No

Item	Interface to services	Reference	Status	Support
S47	Compliance to application context mark (ACM) format	8.3	M	☐ Yes ☐ No
S48	Compliant usage of IEEE P1609.3 services to request ACM pass through and indicate that the WME is waiting for response from RCP to join a WBSS	8.3	M	☐ Yes ☐ No
S49	Processing of ACM received in notification from IEEE P1609.3 checks for all resources of interest	8.4	M	☐ Yes ☐ No
S50	Generates response to provider service table (RPST)	8.4	S47:M	☐ Yes ☐ No
S50.1	RPST includes all onboard unit (OBU) information fields	8.5, Table 17, Table 18, Table 19	M	☐ Yes ☐ No
S50.2	RPST includes all pages of interest that can be returned in a maximum sized packet	8.5	M	☐ Yes ☐ No
S50.3	RPST contains a list of pages of interest that exist but not requested to be returned in the RPST	8.5	M	☐ Yes ☐ No
S50.4	The pages of interest requested to be returned but that would not fit in the maximum sized packet are appended to the list of pages of interest not returned	8.5	M	☐ Yes ☐ No
S51	RM-ResponseToPst received and parsed as described	8.6	S48:M	☐ Yes ☐ No
S51.1	Auto-command sequence, if one exists, issued upon receipt of RPST	8.7	M	☐ Yes ☐ No

Item	Interface to services *(continued)*	Reference	Status	Support
S51.2	Notification sent to resource manager application (RMA) — Immediately if no auto-command sequence present or — After auto-command sequence response if it exists	8.7	M	☐ Yes ☐ No
S52	Exchange requests received from RMAs are passed to OBU as required	8.8	M	☐ Yes ☐ No
S53	Termination of session received at the roadside unit (RSU) clears all internal tables to assure continuous operation of the RSU	8.9	S39: M	☐ Yes ☐ No

Item	Registered pages	Reference	Status	Support
S54	hex(0100) - California Dept of Transportation	Annex B Table B.1	O	☐ Yes ☐ No
S55	hex(0101) - Onboard vehicle network - SAE J1708 [B5]	Annex B Table B.1	O	☐ Yes ☐ No
S56	hex(0102) - Onboard vehicle network - SAE J1939 [B6]	Annex B Table B.1	O	☐ Yes ☐ No
S57	hex(0103..EFFF	Annex B Table B.1	O	☐ Yes ☐ No
S58	hex(FF00) - Keyboard	Annex B Table B.1	S17:M	☐ Yes ☐ No
S59	hex(FF01) - Enunciator	Annex B Table B.1	S15:M	☐ Yes ☐ No
S60	hex(FF02) - Character Readout	Annex B Table B.1	S16:M	☐ Yes ☐ No
S61	hex(FF03) - UI Image	Annex B Table B.1	O	☐ Yes ☐ No
S62	hex(FF04..FFF0) - pages for future registration and usage	Annex B Table B.1	O	☐ Yes ☐ No
S63	hex(FFF1) - SAE J1708 [B5] interface	Annex B Table B.1	O	☐ Yes ☐ No
S64	hex(FFF2) - SAE J1939 [B6] interface	Annex B Table B.1	O	☐ Yes ☐ No
S65	hex(FFF3..FFFD) - pages for future registration and usage	Annex B Table B.1	O	☐ Yes ☐ No
S66	hex(FFFE) - Vehicle Information page	Annex B Table B.1	O	☐ Yes ☐ No
S67	hex(FFFF) - File Transfer page (CVO)	Annex B Table B.1	O	☐ Yes ☐ No

Item	Registered partitions	Reference	Status	Support
S68	hex(0001) - Lockheed Martin	Annex B Table B.2	O	☐ Yes ☐ No
S69	hex(0002) - MFS	Annex B Table B.2	O	☐ Yes ☐ No
S70	hex(0003) - Transcore	Annex B Table B.2	O	☐ Yes ☐ No
S71	hex(0004) - Intermec	Annex B Table B.2	O	☐ Yes ☐ No
S72	hex(0005) - Mark IV	Annex B Table B.2	O	☐ Yes ☐ No
S73	hex(0006) - Raytheon	Annex B Table B.2	O	☐ Yes ☐ No
S74	hex(0007) - Sirit	Annex B Table B.2	O	☐ Yes ☐ No
S75	hex(0008..FE00) - Other	Annex B Table B.2	O	Specify: _____

Annex D

(informative)

Definitions

The following definitions are from *The Authoritative Dictionary of IEEE Standards Terms* [B1], provided here as an aid to the reader. Additional explanation is provided to tailor the definition slightly to this standard. Such additions are provided for explanatory purposes, not to alter the core definition from *The Authoritative Dictionary* (i.e., IEEE 100).

D.1 Command

From IEEE 100:

(1) **(logical link control)** In data communications, an instruction represented in the control field of a protocol data unit (PDU) and transmitted by a logical link control (LLC). It causes the addressed LLC(s) to execute a specific data link control function. (LM/PE/C/TR/CC) 799-1987w, 8802-2-1998

(2) **(A) (electronic computation)** One of a set of several signals (or groups of signals) that occurs as a result of interpreting an instruction; the commands initiate the individual steps that form the process of executing the instruction's operation. **(B) (electronic computation)** Loosely: an instruction in machine language. **(C) (electronic computation)** Loosely: a mathematical or logic operator. **(D) (electronic computation)** Loosely: an operation. (MIL/C/Std100) [2], [20], [85], 162-1963

(3) An input variable established by means external to, and independent of, the feedback (automatic) control system. It sets, is equivalent to, and is expressed in the same units as the ideal value of the ultimately controlled variable. *See also:* feedback control system; set point. (IA/ICTL/IAC) [60]

(4) **(software)** An expression that can be input to a computer system to initiate an action or affect the execution of a computer program; for example, the "log on" command to initiate a computer session. (C) 610.12-1990

(11) A package of information transmitted from the roadside to the vehicle that requests that the transponder on the vehicle perform a specific action. (SCC32) 1455-1999

As applied to this standard:

A packet of information that is transmitted from the RMA hosted by the "provider" and that requests that the "user" perform a specific action and possibly return a response. Commands are the means by which the RMA manages resources on the "user."

D.2 Onboard equipment (OBE)

From IEEE 100:

onboard equipment (OBE) Equipment located within a vehicle that supports the information exchange with roadside equipment (RSE). (SCC32) 1455-1999

As applied to this standard:

This former terminology is what this standard now refers to as *onboard unit* (OBU), which is essentially the radio with the IEEE 802.11 and IEEE 1609 protocol stack. The change is intended to reflect that there is an evolving transition from totally self-contained OBUs (which, when they were a totally self-contained system, could properly be referred to as the onboard

equipment) to an onboard system of which the OBU is but a part. There will be onboard networks, computers, and other devices that need to be identified as an entity of the total communications system, and this overall set is now being referred to as the *onboard equipment* (OBE) to distinguish it from the radio itself, which is now the OBU.

D.3 Page

From IEEE 100:

(1) **(A)** A fixed-length segment of data or of a computer program treated as a unit in storage allocation. **(B)** In a virtual storage system, a fixed-length segment of data or of a computer program that has a virtual address and is transferred as a unit between main and auxiliary storage. (C/Std100) 610.12-1990, 610.5-1990

As applied to this standard:

A segment of memory within a partition having a unique identifier.

D.4 Partition

From IEEE 100:

(1) **(A)** A portion of a computer's main storage that is set aside to hold a single program. **(B)** A portion of a storage medium that is set aside for some special purpose; for example, the boot partition of a magnetic disk contains operating system files from which the computer can be booted. **(C)** A portion of a storage medium that is treated as if it were an individual medium; as in a partition of a hard disk. (C) 610.10-1994

As applied to this standard:

A block of OBU memory that is assigned a unique identifier by which it may be referenced. Each page of memory must be contained within a partition. Within each partition, the page identifiers must be unique. Different partitions may contain pages with the same page identifiers. A partition's size is limited to a maximum of 64K bytes.

D.5 Protocol data unit (PDU)

From IEEE 100:

(1) Information that is delivered as a unit between peer entities a *local area network (LAN)* or a *metropolitan area network (MAN)* and that contains control information, address information, and may contain user data. (LM/C) 8802-6-1994

(2) A block of data that is exchanged between two devices using a protocol. (C) 610.7-1995

(3) Information delivered as a unit between peer entities that may contain control information, address information, and data. (C/MM) 1394-1995

(4) A unit of data specified in a protocol and consisting of protocol information and, possibly, user data. (C/LM) 802.10g-1995

(5) A Distributed Interactive Simulation (DIS) data message that is passed on a network between simulation applications according to a defined protocol. (DIS/C) 1278.1-1995, 1278.2-1995, 1278.4-1997

(6) Information delivered as a unit between peer entities that contains control information and, optionally, data. (C/LM) 8802-5-1998

(7) The sequence of contiguous octets delivered as a unit to the MAC sub-layer or received as a unit from the MAC sublayer. A valid LLC PDU is at least 3 octets in length, and contains two

address fields and a control field. A PDU mayor may not include an information field in addition. (C/LM/CC) 8802-2-1998

(8) Information delivered as a unit between peer entities that contains control information and, optionally, data. (EMB/MIB) 1073.3.2-2000

As applied to this standard:

Data transmitted over the communication media by a service layer containing information required to establish and conduct a communication session between applications.

D.6 Roadside equipment (RSE)

From IEEE 100:

Equipment located at a fixed position along the road transport network, providing communication and data exchange with the onboard equipment (OBE).(SCC32) 1455-1999

As applied to this standard:

This former terminology is what this standard now refers to as *roadside unit* (RSU), which is essentially the radio with the IEEE 802.11 and IEEE 1609 protocol stack. The change is intended to reflect that there has been a transition from totally self-contained roadside equipment into distributed equipment and processor(s) where the RSU is but a part. There will be application processors and other devices that need to be identified as an entity of the total equipment hosted at the roadside. This overall set is now being referred to as the *roadside equipment* (RSE) to distinguish it from the radio itself, which is now the RSU.

D.7 Service data unit (SDU)

From IEEE 100:

(1) Information that is delivered as a unit between peer service access points (SAPs). *See also:* service access point. (LM/C/EMB/MIB) 8802-6-1994, 1073.3.2-2000

(2) The 48-byte data payload of an asynchronous transfer mode (ATM) Cell. (C/BA) 1393-1999

(3) The data associated with a service primitive. (SCC32) 1455-1999

(4) Information delivered as a unit between adjacent entities that may also contain a PDU of the upper layer. (C/LM) 8802-5-1998

As applied to this standard:

The data associated with a service primitive that communicates information between service layers within a protocol stack.

D.8 Session

From IEEE 100:

(1) The period of time during which a user of a terminal can communicate with an interactive system, usually equal to elapsed time between logon and logoff. (C) 610.10-1994w

(2) An execution of a software administration command from initiation to completion on all applicable roles. (C/PA) 1387.2-1995

(3) A collection of process groups established for job control purposes. Each process group is a member of a session. A process is considered to be a member of the session of which its process group is a member. A newly created process joins the session of its creator. A process can alter its session membership. Implementations that support the *setpgid*() function can have multiple process groups in the same session. (PA/C) 9945-1-1996, 9945-2-1993

(4) A portion of an exercise that is contiguous in wall clock (sidereal) time and is initialized per a session database. (DIS/C) 1278.3-1996

(5) A sequence of directory operations requested by a particular user of a particular DUA using the same session OM object. (PA/C) 1328.2-1993w, 1326.2-1993w, 1327.2-1993w, 1224.2-1993w

(7) A portion of an exercise that is contiguous in wall clock (sidereal) time and is initialized by a session database that includes network, entity, and environment initialization and control data. (C/DIS) 1278.4-1997

(8) A series of transactions exchanged between the roadside and the vehicle while the vehicle is within a beacon's communications zone. (SCC32) 1455-1999

As applied to this standard:

A transaction consisting of an exchange of data packets between communicating devices.

D.9 Transaction

From IEEE 100:

(1) An event that requires data contained in a master file to be processed. *See also:* change transaction; null transaction; update transaction; delete transaction; add transaction. (C) 610.2-1987

(2) A data element, control element, signal, event, or change of state that causes, triggers, or initiates an action or sequence of actions. (SE/C) 610.12-1990

(3) **(supervisory control, data acquisition, and automatic control)** That sequence of messages between master and remote stations required to perform a specific function (for example, acquire specific data or control a selected device). (SUB/PE) C37.1-1994

(6) A sequence of messages between cooperating terminals to perform a specific function. Usually a minimum of one message in each direction that is comprised of a command followed by a response. (SUB/PE) 999-1992w

(7) A unit of work consisting of an arbitrary number of individual operations, all of which will either complete successfully or abort with no effect on the intended resources. A transaction has well-defined boundaries. A transaction starts with a request from the application program and either completes successfully (commits) or has no effect (abort). Both the commit and abort signify completion of a transaction. (C/PA) 14252-1996

(8) An information exchange between two nodes. A transaction consists of a request subaction and a response subaction. The request subaction transfers commands (and possibly data) between a requester and a responder. The response subaction returns status (and possibly data) from the responder to the requester. (C/MM) 1596.5-1993, 1596-1992

(9) A transfer between requester and responder consisting of a request and response subaction. The request subaction transfers a command (and sometimes data) between a requester and responder. The response subaction returns status (and sometimes data) from the responder to the requester. A transaction may be either unified or split. (C/MM) 1212-1991s

(10) A single use of a service. (ATLAS) 1232-1995

(11) A transaction is a sequence of packets sent between two or more terminal nodes to perform some function. *See also:* transaction layer. (C/BA) 1355-1995

(12) A request and the corresponding response. The response may be null for transactions with broadcast destination addresses. This is the PDU for the transaction layer. (C/MM) 1394-1995

(13) An event initiated with a connection phase and terminated with a disconnection phase. Data may or may not be transferred during a transaction. Often used instead of the more precise phrase "bus transaction" for the sake of brevity. *See also:* bus transaction; system transaction. (C/BA) 10857-1994

(14) A functionally continuous and complete exchange of information between the roadside equipment (RSE) and the vehicle transponder. (SCC32) 1455-1999

As applied to this standard:

A functionally continuous and complete exchange of information between the “provider” and the “user.”

D.10 User

From IEEE 100:

(1) **(radix-independent floating-point arithmetic) (binary floating-point arithmetic)** Any person, hardware, or program not itself specified by IEEE Std 754™-1985 or IEEE Std 854™-1987 or both, having access to and controlling those operations of the programming environment specified in these standards. (MM/C) 854-1987r, 754-1985r

(2) One who uses the services of a computer system. (C) 610.2-1987, 610.10-1994w

(3) **(broadband local area networks)** An individual whose principal concern is the transfer of information through the system, and to whom the system is transparent. The user is assumed to be in possession of a device that is capable of one- or two-way communication through the system. (LM/C) 802.7-1989r

(4) **(software user documentation)** Person who uses software to perform a task. (C/SE) 1063-1987r

(8) The ultimate human interface or top-most application program. For example, text typed into a terminal interface can be referred to as “user data”. This document does not use the ISO Open Systems “layer-user” concept, in which each module in a vertical stack is the “user” of the adjacent, lower module. (MM/C) 1212.1-1993

As applied to this standard:

The part of an application that uses services of a “provider.”

Annex E

(informative)

Bibliography

[B1] IEEE 100, *The Authoritative Dictionary of IEEE Standards Terms*, Seventh Edition.[15]

[B2] IEEE P802.11p, Draft Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Wireless Access in Vehicular Environments (WAVE).

[B3] IEEE P1609.4™, Draft Trial-Use Standard for Wireless Access in Vehicular Environments (WAVE)—Multi-Channel Operation.

[B4] IEEE Std 1455-1999, IEEE Standard for Message Sets for Vehicle/Roadside Communications.

[B5] SAE J1708, Serial Data Communications Between Microcomputer Systems in Heavy-Duty Vehicle Applications.[16]

[B6] SAE J1939, Recommended Practice for a Serial Control and Communications Vehicle Network.

[15]IEEE publications are available from the Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ 08854, USA (http://standards.ieee.org/).

[16]SAE publications are available from the Society of Automotive Engineers, 400 Commonwealth Drive, Warrendale, PA 15096, USA (http://www.sae.org/).