INTERNATIONAL STANDARD

ISO 24102-2

Second edition
2018-11

Intelligent transport systems — ITS station management —

Part 2: Remote management of ITS-SCUs

Systèmes intelligents de transport — Gestion de la station ITS —

Partie 2: Gestion à distance des SCUs-ITS

Reference number
ISO 24102-2:2018(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Fax: +41 22 749 09 47
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 204, *Intelligent transport systems*.

This second edition cancels and replaces the first edition (ISO 24102-2:2015) which has been technically revised.

A list of all parts in the ISO 24102 series can be found on the ISO website.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

NOTE The former ISO 24102-5 has been converted into a separate standard ISO 22418, as it is not a station management standard.

Introduction

This document is part of a series of International Standards for communications in intelligent transport systems (ITS) based on the ITS station and communications architecture specified in ISO 21217 and illustrated in Figure 1.

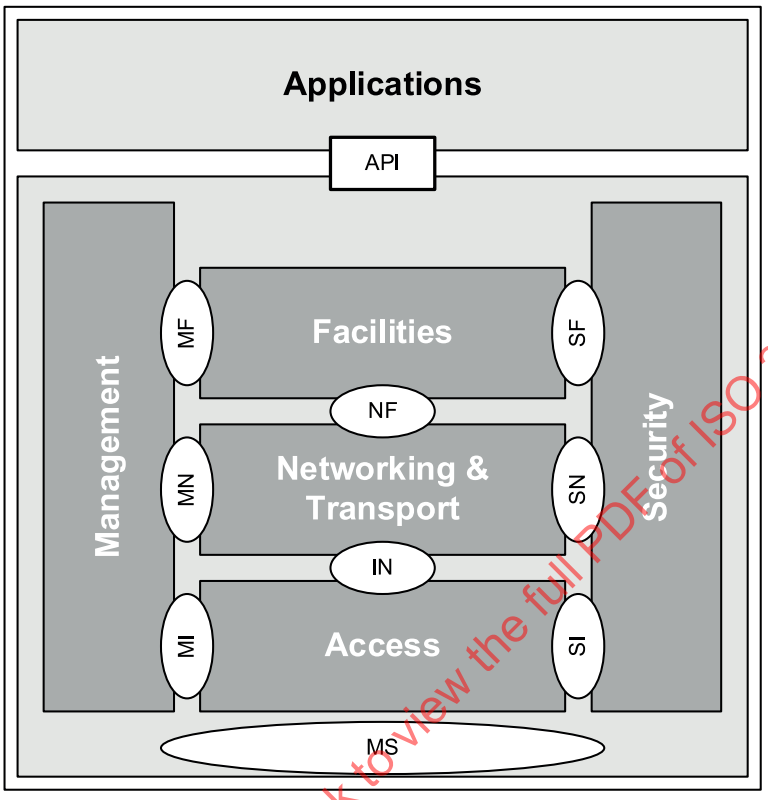


Figure 1 — ITS station reference architecture

This document is Part 2 of a multi-part document which determines remote management of an ITS station unit (ITS-SU) operated as a bounded secured managed entity (BSME).

Remote ITS station management has the purpose of

— setting, updating and deletion of configuration and operation information in an ITS station communication units (ITS-SCU) of an ITS station unit (ITS-SU) specified in ISO 21217, e.g. information on policies and regulations, security related information, accounting information, communication protocol layer parameters[5],

— installation, update and uninstallation of persistent information in an ITS-SCU, e.g. ITS-S application processes specified in ISO 21217, ITS-S communication protocols,

— notification and retrieval of management information, e.g. log files of events, alarms generated by the ITS-SCU(s) of an ITS-SU.

By this it covers the five management areas identified in ISO/IEC 7498-4[1].

Intelligent transport systems — ITS station management —

Part 2:
Remote management of ITS-SCUs

1 Scope

This document provides specifications for intelligent transport systems (ITS) station management to conform with the ITS station reference architecture.

Remote ITS station management is specified by means of protocol data units (PDUs) and procedures of the "Remote ITS Station Management Protocol" (RSMP) related to managed objects in an ITS station communication unit. Distinction is made between managed entities (management clients) and managing entities (management servers).

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 8825-2, *Information technology — ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)*

ISO TS 16460, *Intelligent transport systems — Communications access for land mobiles (CALM) — Communication protocol messages for global usage*

ISO 17419, *Intelligent transport systems — Cooperative systems — Globally unique identification*

ISO 17423, *Intelligent transport systems — Cooperative systems — Application requirements and objectives*

ISO 21217, *Intelligent transport systems — Communications access for land mobiles (CALM) — Architecture*

ISO 24102-1, *Intelligent transport systems — ITS station management — Part 1: Local management*

ISO 24102-3, *Intelligent transport systems — ITS station management — Part 3: Service access points*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 21217, ISO 24102-1, ISO 24102-3, ISO TS 16460, and ISO/IEC 7498-4[1] and the following apply.

ISO and IEC maintain terminological databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at http://www.electropedia.org/

3.1
remote management client
ITS station communication unit in which remote ITS station management is performed by a remote management server

3.2
remote management server
entity performing remote ITS station management in an ITS station communication unit

4 Symbols and abbreviated terms

For the purposes of this document, the abbreviated terms given in ISO 21217, ISO 24102-1, ISO 24102-3, ISO TS 16460, and ISO/IEC 7498-4[1] and the following apply.

BSME	Bounded Secured Managed Entity (from ISO 21217)
FSAP	Fast Service Advertisement Protocol (from ISO 22418[3])
ITS	Intelligent Transport Systems
ITS-SCU	ITS Station Communication Unit (from ISO 21217)
ITS-SCU-CMC	ITS-SCU Configuration Management Centre (from ISO 17419)
ITS-SU	ITS Station Unit (from ISO 21217)
RMC	Remote Management Client
RMCH	Remote Management Communication Handler
RMPE	Remote Management Protocol Execution
RMS	Remote Management Server
RSMP	Remote ITS-station Management Protocol

5 Requirements

The ITS station management entity provides the functionality specified in the various parts of this multi-part document:

— The functionality of local ITS station management specified in ISO 24102-1.

— **The functionality of remote ITS station management specified in this document (Part 2).**

— The functionality of management service access points specified in ISO 24102-3.

— The functionality of ITS station-internal management communications specified in [2] (ISO 24102-4).

— The functionality of path and flow management specified in ISO 24102-6.

Means to secure the access to management functionality need to be specified within the global context of ITS security. Details are outside the scope of this document.

Detailed mandatory requirements are specified in the following clauses of this document:

— Clause 6 presents the remote management architecture.

— Clause 7 specifies remote management protocol data units.

— Clause 8 specifies service primitive functions.

— Clause 9 specifies remote management procedures.

— Clause 10 specifies details needed for the Fast Service Advertisement Protocol (FSAP).

— Clause 11 identifies dynamic data.

— Clause 12 informs about conformance declaration.

— Clause 13 informs about conformance testing.

— The normative Annex A specifies contexts of the RMPE ITS application class.

— The normative Annex B specifies the ASN.1 module for remote management.

— The informative Annex C proposes settings of communication service parameters used for automatic selection of communication profiles specified in ISO 17423.

— The normative Annex D presents the implementation conformance statement (ICS) proforma.

6 Remote management architecture

6.1 Functionality

The "Remote ITS Station Management Protocol" (RSMP) specified in this document has the purpose of

— setting, updating and deletion of configuration and operation information in an ITS station communication unit (ITS-SCU) of an ITS station unit (ITS-SU) specified in ISO 21217, e.g. information on policies and regulations (ISO 17419), security related information, accounting information, access layer parameters[5], etc.

— installation, update and uninstallation of persistent information in an ITS-SCU, e.g. ITS-S application processes, ITS-S communication protocols,

— notification and retrieval of management information, e.g. log files of events, alarms generated by the ITS-SCU of an ITS-SU.

By this it covers the five management areas identified in ISO/IEC 7498-4[1].

Remote ITS station management covers a set of management processes where an ITS station unit (ITS-SU) acting as remote management server (RMS) manages an ITS station communication unit (ITS-SCU) of an ITS-SU acting as remote management client (RMC).

An RMS is associated with an ITS-SCU configuration management centre identified in ISO 17419. An RMS may be implemented e.g. in a roadside ITS sub-system, or in a central ITS sub-system. Several RMSs may be associated with the same ITS-SCU configuration management centre. A single RMSs may be associated with several ITS-SCU configuration management centre. A single ITS-SCU always is associated only with a single ITS-SCU configuration management centre.

Remote ITS station management is applied to managed objects[1] in remote management sessions. Such sessions may be initiated

— by the RMS (server initiated session), e.g. by means of the Fast Service Advertisement Protocol (FSAP)[3] or by direct IPv6 based access, or

— by the RMC (client initiated session), typically using IPv6 communications,

as illustrated in Figure 2 (server initiated session using FSAP), in Figure 3 (direct server initiated session), and in Figure 4 (client initiated session).

The mechanisms specified in this document enable future specifications of remote management features in separate standards or by means of registries.

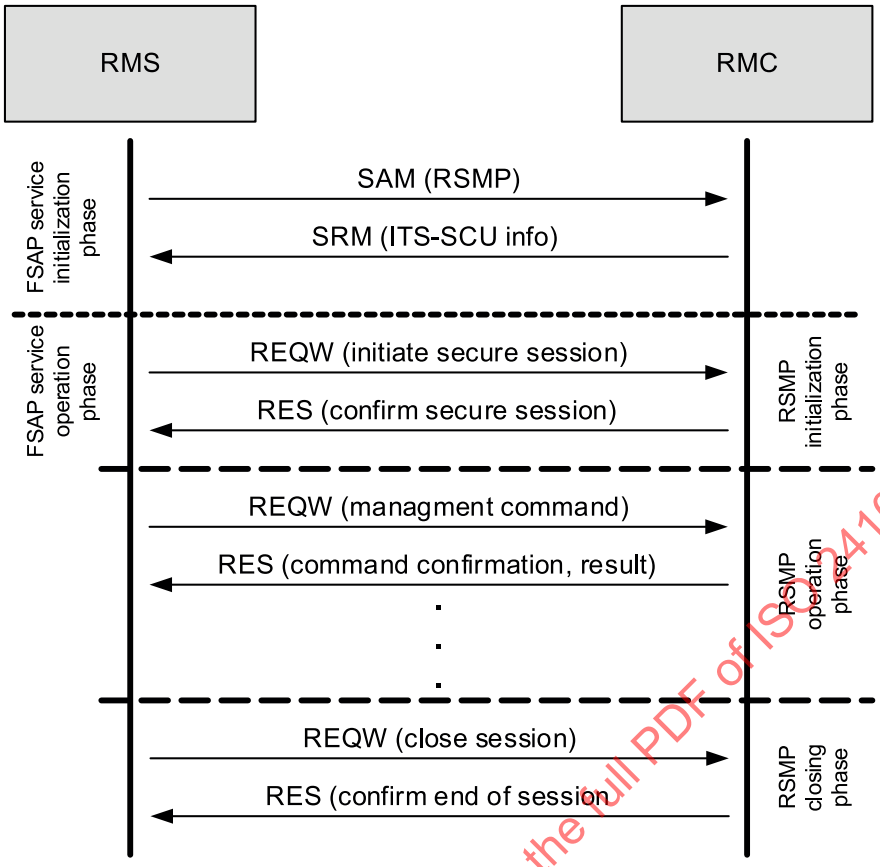


Figure 2 — Server initiated session (example with FSAP)

SAM and SRM specified in [3] with details specified in this document are used in the example of Figure 2 to prepare for the secured management session. During the FSAP service operation phase, first a secure session is requested from the RMS which is acknowledged by the RMC. After successful establishment of a session with mutual authentication of RMS and RMC with optional agreement on encryption of the management data to be exchanged in the session, the RMS may send out a sequence of management commands, each of which is acknowledged by the RMC providing also optional result data. Finally, the RMS closes the session, which also is acknowledged by the RMC. Subsequent to this, no more management data can be exchanged.

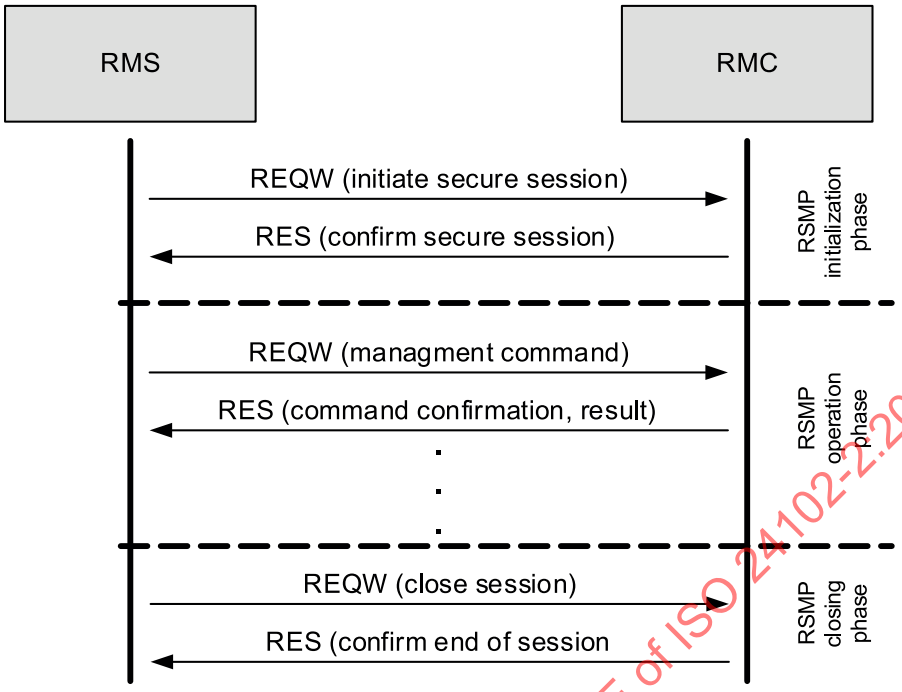


Figure 3 — Direct server initiated session

In the example of Figure 3, an RMS directly initiates a secure session with an RMC. After confirmation of the secure session by the RMC, the RMS runs and closes the secure session as illustrated above for the direct server initiated session.

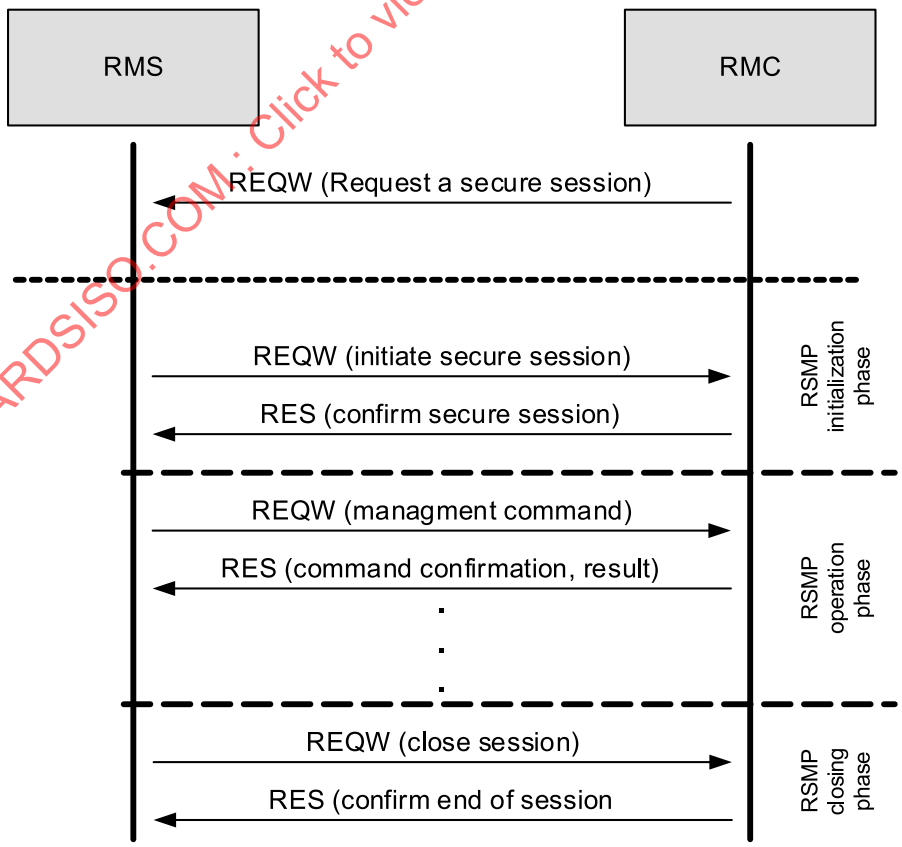


Figure 4 — Client initiated session

In the example of Figure 4, upon an event internal to an RMC, an RMC notifies the need for a secure session to the RMS. Then the RMS initiates, runs, and closes the secure session.

6.2 ITS station architecture

The "Remote ITS-station Management Protocol" (RSMP) consists of two functional blocks, i.e.

— the ITS-S application process "Remote Management Protocol Execution" (RMPE) with a registered "ITS Application Identifier" (ITS-AID) and "the ITS-S application process identifier" (ITS-SAPID) of values

  — 1 for the RSM client and

  — 2 for the RSM server;

— the ITS-S facility "Remote Management Communication Handler" (RMCH) using a well-known registered ITS port number PORT_RSMP and dynamically assigned ITS port numbers[4] for localized communications .The value of PORT_RSMP is 32763.

The allocation of these functional blocks in the ITS station architecture specified in ISO 21217 is presented in Figure 5. Globally unique identifiers are specified in ISO 17419.

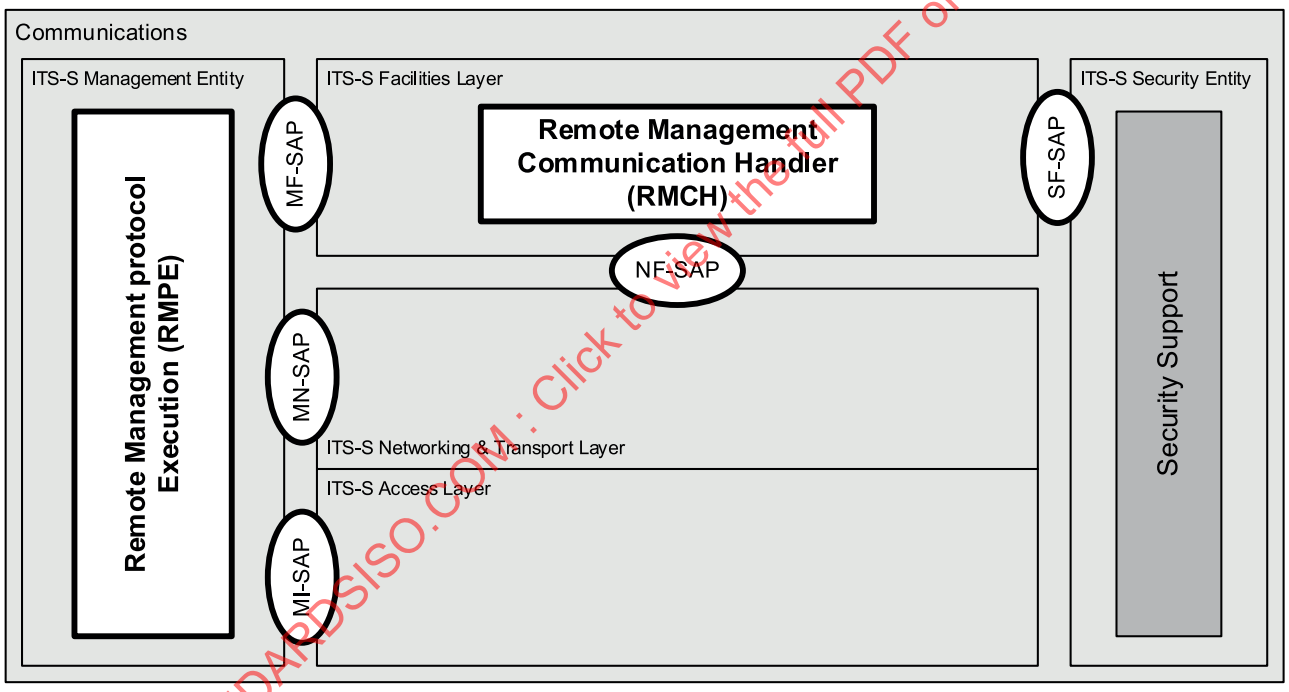


Figure 5 — Functional blocks of RSMP

The RMCH communication protocol is located in the ITS-S facilities layer.

The RMPE ITS-S application process is located in the ITS-S management entity.

RMCH and RMPE are connected via the MF-SAP services MF-COMMAND and MF-REQUEST specified in ISO 24102-3 with service primitive functions specified in Clause 8.

6.3 Distributed implementation of an ITS-S

The "Remote ITS-station Management Protocol" (RSMP) supports distributed implementations of ITS-S roles identified in ISO 21217, i.e. several ITS-SCUs per ITS-SU. The RMCH thus may communicate via the ITS station-internal network with an ITS-SCU providing the link to the peer ITS station unit.

Details depend on the ITS-S networking & transport layer protocol used and are outside the scope of this document.

6.4 RMPE

"Remote Management Protocol Execution" (RMPE) is an ITS-S application process located in the ITS-S management entity. There are two distinct instantiations of the RMPE, i.e. the server instantiation and the client instantiation. There is exactly one instantiation of RMPE in each ITS-SCU of an ITS-SU. The RMPE cannot manage ITS-SCUs in which it is not instantiated.

NOTE For more information on ITS-S application processes see ISO 21217.

Management activities include:

— updating firmware in the ITS-SCU;

— maintenance of ITS-S application processes:

  — new installations;

  — updates of existing installations;

  — deletion of existing installations;

— maintenance of communication, management and security protocols:

  — new installations;

  — updates of existing installations;

  — deletion of existing installations;

— maintenance of management parameters:

  — setting of parameter values and other information;

  — retrieval of parameter values and other information, e.g. logfiles;

— maintenance of security related managed objects.

6.5 RMCH

The "Remote Management Communication Handler" (RMCH) is a communication facility located in the ITS-S facilities layer. The RMCH

— receives service data units which contain "RMCH Protocol Data Units" (RMCH-PDUs) illustrated in Figure 6 from peer ITS-SUs,

— exchanges RSMP-PDUs illustrated in Figure 7 with the RMPE via the MF-SAP,

— transmits RMCH-PDUs to peer ITS-SUs, and

— uses services from the ITS-S security entity via SF-SAP service primitives to authenticate peer ITS station units, and to optionally encrypt and decrypt RMPE-PDUs.

The well-known ITS port PORT_RSMP[4] for localized communications is used by

a) an RMS for transmission of a message

  — as a source port number;

  — as a destination port number in case of direct session initiation, and only in the REQW (initiate secure session) message shown in Figure 3. With the REQW (initiate secure session),

an RMS requests an RMC to use a specific dynamically assigned port number for subsequent communications.

b) an RMC for transmission of a message

— as a destination port number in case of direct session initiation, and only in the "Request a secure session" message shown in Figure 4.

Client instantiations use dynamically assigned port numbers as source port number.

7 Remote management protocol data units

Remote station management uses the protocol data units, data elements and security elements illustrated in Figure 6, Figure 7, Figure 8 and Figure 9. The ASN.1 presentation of these PDUs is specified in Annex B.

RMCH-Request/Response:

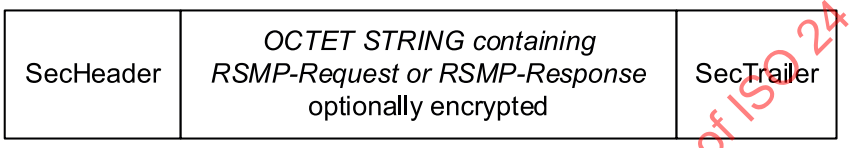
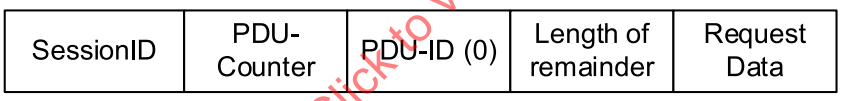


Figure 6 — RMCH protocol data units

RMCH-Request and RMCH-Response messages encapsulate RSMP-Request and RSMP-Response messages, respectively, between a security header and a security trailer. RSMP-Request and RSMP-Response messages may be encrypted as indicated in the SecHeader fields.

RSMP-Request:



RSMP-Response:

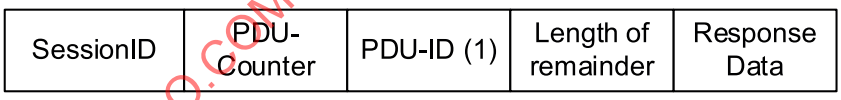


Figure 7 — RSMP protocol data units RSMP-Request and RSMP-Response

RSMP-Request messages of ASN.1 type `RSMPmessage` are sent by RMSs, and by RMCs to request a remote management session as illustrated in Figure 4. RSMP-Response messages of ASN.1 type `RSMPmessage` are sent only by RMCs.

The SessionID identifies uniquely a session for a specific ITS-SCU and a specific management centre. The value zero is used by a RMC in the RSMP-Request message requesting initiation of a secure session by a RMS.

The PDU-Counter distinguishes PDUs uniquely in a session. The value presented in a RSMP-Request is used in the corresponding RSMP-Response.

The PDU-ID distinguishes RSMP-Request and RSMP-Response messages.

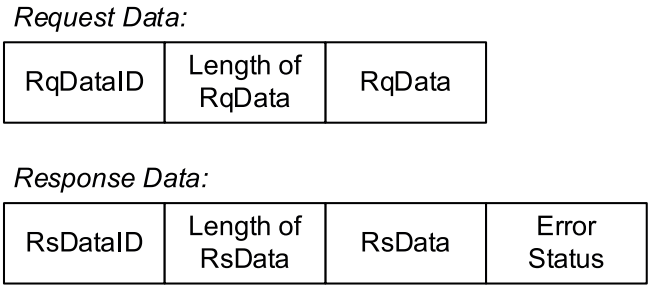


Figure 8 — Request and response data elements `RequestData` and `ResponseData`

RqDataIDs are given in the ASN.1 type `RefRSMPREQ`, and RsDataIDs are given in the ASN.1 type `RefRSMPRES`. These IDs uniquely identify RqData and RsData, respectively.

The ErrorStatus is given by the ASN.1 type `RSMPErrStatus`.

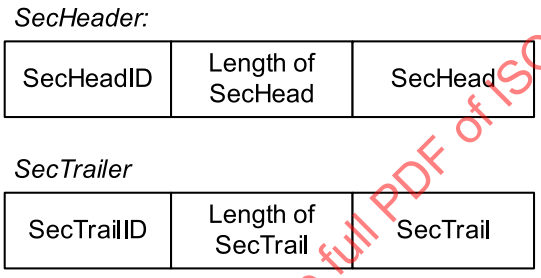


Figure 9 — Security header and trailer `SecHeader` and `SecTrailer`

SecHeadIDs and SecTrailIDs are given in the ASN.1 type `RefSECRSMP`. Not applying any security is given by the ASN.1 value `c-noSecurity`; the corresponding SecHead and SecTrail are of ASN.1 type `NullType`.

8 Service primitive functions

8.1 Generic service primitives

The service primitives MF-COMMAND.request. MF-COMMAND.confirm, MF-REQUEST.request, MF-REQUEST.confirm, SF-REQUEST.request and SF-REQUEST.confirm are specified in ISO 24102-3. This document specifies functions for these generic service primitives applicable for RSMP.

8.2 MF-SAP service primitive functions

8.2.1 Transmission request of RSMP-Request and RSMP-Response

The service primitive MF-COMMAND.request is used. ASN.1 types and values for the applicable function shall be as specified in Table 1, with ASN.1 details specified in Annex B.

Table 1 — Transmission request of RSMP-Request and RSMP-Response

MF-Command-request		
&mxref	**&MXParam**	**Description**
c-rsmpMessageTX	RSMPmessageTX	Request to transmit RSMP-Request and RSMP-Response to a specific peer station
MF-Command-confirm		
&mxref	**&MXParam**	**Description**
c-rsmpMessageTX	RSMPmessageTXconf	The MF-Command-request RSMPmessageTX shall be confirmed with RSMPmessageTXconf and with ErrStatus specified in ISO 24102-3.

8.2.2 Notification of reception of RSMP-Request and RSMP-Response

The service primitive MF-REQUEST.request is used. ASN.1 types and values for the applicable functions shall be as specified in Table 2, with ASN.1 details specified in Annex B.

Table 2 — Notification of reception of RSMP-Request and RSMP-Response

MF-Request-request		
&mxref	**&MXParam**	**Description**
c-rsmpMessageRX	RSMPmessageRX	Notification of reception of RSMP-Request and RSMP-Response from specific peer station
MF-Request-confirm		
&mxref	**&MXParam**	**Description**
c-rsmpMessageRX	RSMPmessageRXconf	The request RSMPmessageRX shall be confirmed with RSMPmessageRXconf and with ErrStatus specified in ISO 24102-3.

8.3 SF-SAP service primitive functions

8.3.1 Security procedure applied to RSMP-Request and RSMP-Response

The service primitive SF-REQUEST.request is used. ASN.1 types and values for the applicable functions shall be as specified in Table 3, with ASN.1 details specified in Annex B.

Table 3 — Security procedure applied to RSMP-Request and RSMP-Response

SF-Request-request		
&mxref	**&MXParam**	**Description**
c-secRSMPmessageTX	SecRSMPmessageTX	Request to secure RSMP-Request/RSMP-Response prior to transmission to a specific peer station
SF-Request-confirm		
&mxref	**&MXParam**	**Description**
c-secRSMPmessageTX	RMCHmessage	The SF-Request-request SecRSMPmessageTX shall be confirmed with the secured copy of RSMP-Request/RSMP-Response, i.e. the RMCH-Request/RMCH-Response and with ErrStatus specified in ISO 24102-3.

8.3.2 Security procedure applied to RMCH-Request and RMCH-Response

The service primitive SF-Request.request is used. ASN.1 types and values for the applicable functions shall be as specified in Table 4, with ASN.1 details specified in Annex B.

Table 4 — Security procedure applied to RMCH-Request and RMCH response

SF-Request-request		
&mxref	**&MXParam**	**Description**
c-secRMCHrX	SecRMCHrX	Request to process security of a received RMCH-Request/RMCH-Response from a specific peer station prior to forwarding to the RMPE
SF-Request-confirm		
&mxref	**&MXParam**	**Description**
c-secRMCHrX	RSMPmessage	The SF-Request-request SecRMCHrX shall be confirmed with the unsecured copy of RMCH-Request/RMCH-Response, i.e. the RSMP-Request/RSMP-Response and with ErrStatus specified in ISO 24102-3.

9 Remote management procedures

9.1 Remote management session initiation

9.1.1 Initiation by server

An RMS may initiate a remote management session upon purpose. In order to start a remote management session, the RMS shall send the RSMP-Request(RMSinitSession) to the RMC in the desired ITS-SCU.

Upon reception of a valid RSMP-Request(RMSinitSession), the RMC shall acknowledge the request with the RSMP-Response(RMSinitSessionRs).

9.1.2 Initiation by client

An RMC may initiate a remote management session upon purpose. In order to request initiation of a remote management session, the RMC shall send the RSMP-Request(RMCreqSession) to the desired management center. This request may be repeated until the RMC receives an RSMP-Request(RMSinitSession) from the selected RMS starting the requested session.

NOTE An ITS-SCU knows how to reach its associated ITS-SCU management centre and the necessary communication protocols and addresses. Details are out of scope of this document. Typically a cellular network and Internet can be used.

Upon reception of the RSMP-Request(RMSinitSession), the RMC shall acknowledge the request with the RSMP-Response(RMSinitSessionRs).

9.1.3 RSMP session identifier

Every remote management session is identified by an RSMP session identifer "SessionID" of ASN.1 type RsmpSessionID assigned by the RMC. The values of SessionID shall be unique for a given RMC. The value zero shall be used by a RMC in the RSMP-Request message requesting initiation of a secure session by a RMS as specified in Clause 7.

9.1.4 RSMP session security

Details of security schemes to be applied for a remote management session may be negotiated between RMS and RMC. This negotiation is performed using the RSMP-Request and RSMP-Response PDUs specified in Clause 7. Details of these PDUs will be specified in a future version of this document.

9.2 Remote management session closure

9.2.1 Active closure

As soon as no further remote management actions are needed in a remote management session, the RMS shall close the session. Closure of a remote management session shall be indicated by the RMS by sending the RSMP-Request(`RMScloseSession`) to the RMC.

Upon reception of the RSMP-Request(`RMScloseSession`), the RMC shall acknowledge the request with the RSMP-Response(`RMScloseSessionRs`), and shall treat the remote management session as closed.

9.2.2 Timeout

In case an RMC does not receive RSMP-Requests from a RMS during an open remote management session for a time period larger than the time indicated in the management parameter RSMPtimeout of ASN.1 type `RSMPtimeout`, the RMC shall close the remote management session on its own.

In order to keep a session alive, an RMS may periodically send the Ping-command RSMP-Request(`RSMPping`) with arbitrary data which shall be acknowledged by the RMC with the Ping-command RSMP-Response(`RSMPping`) returning the received arbitrary data.

NOTE This document does not specify a value for RSMPtimeout.

9.2.3 No active session

As long as an RMC is not in an active remote management session with an RMS, it shall ignore all RSMP-Request messages except the one to start a remote management session and an optionally repeated RSMP-Request(`RMScloseSession`).

9.3 Firmware update

A partial or complete update of the firmware of an ITS-SCU may be requested by the RMS by sending the RSMP-Request(`firmwareUpdate`) message. Upon reception of the RSMP-Request(`firmwareUpdate`) message the RMC shall first acknowledge reception of the RSMP-Request(`firmwareUpdate`) message by sending the RSMP-Response(`firmwareUpdateRs`) message, and shall then perform the requested firmware update in line with requirements on secure operation of the platform in which the ITS-SCU is installed. Upon automatic restart of the ITS-SCU after the firmware update, the ITS-SCU shall request a remote management session at the RMS to confirm success of the operation, and to close correctly the session.

As long as a firmware update session is active, other management sessions are prohibited.

9.4 Maintenance of ITS-S protocols

ITS-S protocols are protocols different to ITS-S application processes which reside in the ITS-S access layer, the ITS-S networking & transport layer, the ITS-S facilities layer, the ITS-S management entity and the ITS-S security entity.

Maintenance of an ITS-S protocols may be performed with RSMP-Request(`ProtMaintenance`) distinguishing three types of maintenance operation:

a) Installation of a new ITS-S protocol requested with ASN.1 type `ProtMgmtInstallRq`.

b) Update of an existing ITS-S protocol requested with ASN.1 type `ProtMgmtUpdateRq`. The existing ITS-S protocol is identified with ASN.1 type `ProtocolID` provided by the RSC in the acknowledgement of an initial installation.

c) Deletion of an existing ITS-S protocol requested with ASN.1 type `ProtMgmtDeleteRq`. The existing ITS-S protocoll is identified with ASN.1 type `ProtocolID` provided by the RSC in the acknowledgement of an initial installation.

A maintenance request shall be acknowledged by an RMC with RSMP-Response (`protMaintenanceRs`) distinguishing three types of maintenance operation:

a) Acknowledgement of an installation of a new (instantiation of an) ITS-S application process with ASN.1 type `ProtMgmtInstallRs` providing the unique application Identifier of ASN.1 type `ProtocolID` of this instantiation of an ITS-S application process. In case of failure `ProtocolID.instance` shall contain the value zero, otherwise the value contained in `ProtMgmtInstallRq`.

b) Acknowledgement of an update of an existing ITS-S protocoll with ASN.1 type `ProtMgmtUpdateRs`. In case of failure `ProtocolID.instance` contained in `ProtMgmtUpdateRs` shall contain the value zero, otherwise the value contained in `ProtMgmtUpdateRq`.

c) Acknowledgement of deletion of an existing ITS-S protocoll with ASN.1 type `ProtMgmtDeleteRs`. In case of success `ProtocolID.instance` contained in `ProtMgmtDeleteRs` shall contain the value zero, otherwise the value contained in `ProtMgmtDeleteRq`.

The return status of ASN.1 type `RSMPErrStatus` contained in the acknowledgement shall present a value as specified in Table 5.

Table 5 — Protocol maintenance result status

Status value in `RSMPErrStatus`	Installation	Update	Deletion
`rsmpErrSuccess`	Successful installation	Successful update	Successful deletion
`rsmpErrRejected`	Request rejected by RSC for unknown reasons		
`rsmpErrProtUnknown`	n.a.	Request rejected as referenced ITS-S protocol given in ASN.1 type `ProtocolID` is not known at the RSC.	
`rsmpErrUnspecFailure`	Request failed for unknown reasons		

9.5 Maintenance of ITS-S application processes

Maintenance of an ITS-S application process may be performed with RSMP-Request(`appMaintenance`) distinguishing three types of maintenance operation:

a) Installation of a new (instantiation of an) ITS-S application process requested with ASN.1 type `AppMgmtInstallRq`.

b) Update of an existing ITS-S application process requested with ASN.1 type `AppMgmtUpdateRq`. The existing ITS-S application process is identified with ASN.1 type `ITSsapiid` provided by the RSC in the acknowledgement of an initial installation.

c) Deletion of an existing ITS-S application process requested with ASN.1 type `AppMgmtDeleteRq`. The existing ITS-S application process is identified with ASN.1 type `ITSsapiid` provided by the RSC in the acknowledgement of an initial installation.

A maintenance request shall be acknowledged by an RMC with RSMP-Response(`appMaintenanceRs`) distinguishing three types of maintenance operation:

a) Acknowledgement of an installation of a new (instantiation of an) ITS-S application process with ASN.1 type `AppMgmtInstallRs` providing the unique application Identifier of ASN.1 type `ITSsapiid` of this instantiation of an ITS-S application process. In case of failure `ITSsapiid.AppInstance` shall contain the value zero, otherwise the value contained in `AppMgmtInstallRq`.

b) Acknowledgement of an update of an existing ITS-S application process with ASN.1 type `AppMgmtUpdateRs`. In case of failure `ITSsapiid.AppInstance` contained in `AppMgmtUpdateRs` shall contain the value zero, otherwise the value contained in `AppMgmtUpdateRq`.

c) Acknowledgement of deletion of an existing ITS-S application process with ASN.1 type `AppMgmtDeleteRs`. In case of success `ITSsapiid.AppInstance` contained in `AppMgmtDeleteRs` shall contain the value zero, otherwise the value contained in `AppMgmtDeleteRq`.

The return status of ASN.1 type `RSMPErrStatus` contained in the acknowledgement shall present a value as specified in Table 6.

Table 6 — Application maintenance result status

Status value in `RSMPErrStatus`	Installation	Update	Deletion
`rsmpErrSuccess`	Successful installation	Successful update	Successful deletion
`rsmpErrRejected`	Request rejected by RSC for unknown reasons		
`rsmpErrAppUnknown`	n.a.	Request rejected as referenced instantiation of an ITS-S application process given in ASN.1 type `ITSsapiid` is not known at the RSC.	
`rsmpErrUnspecFailure`	Request failed for unknown reasons		

9.6 Maintenance of configuration information

In order to read the value of one or several management parameters (M-Params specified in ISO 24102-1), an RMS shall send the RSMP-Request(`getMparams`) message. Upon reception of the RSMP-Request(`getMparams`) message, the RMC shall return the requested information in the RSMP-Response(`getMparamsRs`) message. In case a parameter value cannot be provided, it shall be omitted in RSMP-Response(`getMparamsRs`).

In order to write the value of one or several management parameters (M-Params specified in ISO 24102-1), an RMS shall send the RSMP-Request(`setMparams`) message. Upon reception of the RSMP-Request(`setMparams`), the RMC shall perform the requested parameter settings in case no access violation occurred, and shall report success or failure in the RSMP-Response(`setMparamsRs`). In case all requested parameter settings could be performed, the global result status shall be set to "`rsmpErrSuccess`". Otherwise the global result status shall be set to "`rsmpErrSetErrorGeneral`" and detailed result status shall be returned for every failure that occurred.

10 Usage of FSAP

10.1 General

The "Fast Service Advertisement Protocol" FSAP is specified in [3]. This document specifies usage of FSAP for RSMP; for the general parts of FSAP see [3].

10.2 SAM

The "ITS-AID" field contained in the element for "RMPE advertisement" of the "Service Info Segment" of the "Service Advertisement Message" (SAM) shall contain the registered value of RMPE. RMPE is registered as an ITS Application Class; see ISO 17419. The various contexts of this application class are distinguished by the globally unique identifier ITS-SCU-CMCID of the ITS-SCU configuration management centre (ITS-SCU-CMC) of the ITS-SCU to be managed.

NOTE 1 The registered value of ITS-AID for RMPE is 134.

NOTE 2 ITS-SCU-CMCID is of ASN.1 type `ItsScuCmcID::=OBJECT IDENTIFIER`.

Contexts of the RMPE ITS application class are presented in Annex A.

The "Reply Port Number" field, if present in the element for "RMPE advertisement" of the "Service Info Segment" of a SAM, shall show the registered well-known number PORT_RSMP specified in ISO 17419.

NOTE 3 Presence of the "Reply Port Number" field is not required, as the value contained in it is the registered well-known port number PORT_RSMP.

NOTE 4 The registered value of the well-known port number PORT_RSMP is 32763.

The optional "SAMapplicationData" field of the "Service Info Extensions" field contained in the element for "RMPE advertisement" of the "Service Info Segment" of a "Service Advertisement Message" (SAM) may contain the unique identifier of ASN.1 type `ItsScuCmcID` of the ITS-SCU-CMC specified in ISO 17419 that offers remote station management. In case different ITS-SCU-CMCs are available, this optional field shall not be used.

10.3 SRM

The "ITS-AID" sub-field of the "Context Information field" contained in the element for "RMPE context" presented in a "Service Response Message" (SRM) shall contain the registered value of RMPE.

NOTE The registered value of ITS-AID for RMPE is 134.

The fields "Context ID" and "Context Data" contained in the element for "RMPE context" presented in a "Service Response Message" (SRM) shall contain the applicable RMPE context information of ASN.1 type `SamContext`, see Annex A for more details.

11 Dynamic data

Dynamic data are data that add functionality to this standard, but may be specified in other standards. Details of dynamic data are specified based on the generic definitions in this document, and may become part of this document once the other standard applies for a registration of these dynamic data details.

Dynamic data are:

— Security message formats, see Clause 7, B.2. Security message formats are presented with ASN.1 open types based on `SECRSMP`. Privately defined formats are allowed.

— Data contained in RSMP request and response PDUs, see Clause 7, B.2. RSMP request data are presented with ASN.1 open types based on `RSMPREQ`. RSMP response data are presented with ASN.1 open types based on `RSMPRES`. Privately defined data for RSMP requests and responds are allowed.

— Reason codes indicating why an RMC requests a remote management session, see 9.1.2, B.2. Reason codes are presented with ASN.1 open types based on `RMCREQREASON`. Privately defined reason codes are allowed.

— Protocol maintenance features, see 9.4, B.2. Protocol maintenance features are presented with ASN.1 open types based on `PROTMNT`. Privately defined protocol maintenance features are allowed.

— ITS-S application process maintenance features, see 9.5, B.2. ITS-S application process maintenance features are presented with ASN.1 open types based on `APPMNT`. Privately defined ITS-S application process maintenance features are allowed.

— Configuration parameters, i.e. M-Parameters originally specified in ISO 24102-1, see 9.6, B.2.

12 Conformance

An "Implementation Conformance Statements" (ICS) proforma used to declare elements of an implementation conforming to this document is provided in Annex D.

13 Test methods

A "Test Suite Structure & Test Purposes" (TSS&TP) specification for conformance testing is not yet specified.

An "Abstract Test Suite" (ATS) for conformance testing is not yet specified.

Annex A
(normative)

Contexts of the RMPE ITS application class

The RMPE ITS application class has contexts distinguished by the globally unique ITS-SCU-CMC identifier ITS-SCU-CMCID specified in ISO 17419.

Usage of contexts is specified in the Fast Service Advertisement Protocol specification ISO 22418[3] and the underlying format specification standard ISO/TS 16460. ISO/TS 16460 specifies the ASN.1 type `SamContext` that maps context data to a context identifier of ASN.1 type `ItsAidCtxRef` composed of the ITS-AID of an ITS application class, and the unique INTEGER context identifier of ASN.1 type `CtxRef`.

For the ITS-AID of RMPE the contexts presented in Table A.1 are identified.

Table A.1 — RMPE contexts

Values of context identifier of ASN.1 type `CtxRef` contained in component `ctx` of `ItsAidCtxRef`	Context information
`c-ctxRefNull` = 0	Null-context (= don't know/not applicable context) applicable for all values of ITS-AID.
`c-ctxRefRmpe<X>`, x > 0	Positive INTEGER number mapping to the globally unique ITS-SCU-CMC object identifier ITS-SCU-CMCID of ASN.1 type **`ItsScuCmcID`**[a]. `c-CtxTypeRmpe<X> ItsAidCtxRef::={itsaid extension:content:134, ctx c-ctxRefRmpe}`[b] Extension of `AllsamContexts` in implementations of ISO/TS 16460: `{ ItsScuCmcID IDENTIFIED BY c-CtxTypeRmpe<X> }`

[a] It might be desirable to update ISO/TS 16460, i.e. removing the size constraint from the ASN.1 type CtxRef in order to allow for more than 255 ITS-SCU-CMCs.

[b] <X> to be replaced by a text string uniquely identifying the corresponding INTEGER value x.

At time of writing this document no values were assigned. Future assignments will be published in subsequent amendments or versions to this document.

Annex B
(normative)

ASN.1 modules

B.1 Overview

The following ASN.1 module is specified in this annex:

— ITSSremoteMgmt2 { iso (1) standard (0) calm-management (24102) remote (2) asnm-1 (1) version2 (2)}

B.2 Module ITSSremoteMgmt

This module specifies ASN.1 type definitions together with useful ASN.1 value definitions.

Unaligned packed encoding rules (PER) as specified in ISO/IEC 8825-2 shall be applied for this ASN.1 module.

```
ITSSremoteMgmt2 { iso (1) standard (0) calm-management (24102) remote
(2) asnm-1 (1) version2 (2)}

DEFINITIONS AUTOMATIC TAGS::=BEGIN

IMPORTS

-- C-ITS Data Dictionary (still in ISO 17419)
NullType, Latitude, Longitude, TimeDurationValue, PortNumber FROM
CITSdataDictionary1 {iso(1) standard(0) cits-applMgmt (17419)
dataDictionary (1) version1 (1)}

-- ISO 17419
ITSscuID, ItsScuCmcID, ITSaid, ITSSapPrPr, ITSSapdID, ITSSpPr,
ITSaooID, ITSprotID, ITSSpdID, ITSpoID, ProtocolID, ITSsapiid, ITSsapid
FROM CITSapplMgmtApplReg {iso(1) standard(0) cits-applMgmt (17419)
applRegistry (2) version2 (2)}

-- ISO 24102-1
Param24102, RefMPARAM FROM ITSmanagement { iso (1) standard (0)
calm-management (24102) local (1) asnm-1 (1) version2 (2)}
;

-- End of IMPORTS

-- Types

-- Security header and trailer --
SECRSMP::=CLASS{
   &ref   RefSECRSMP, -- security type identifier
   &SecRSMP
   }

RefSECRSMP::=INTEGER{
   c-noSecurity   (0),
   c-octString    (1)
   }(0..255)

SecRSMPs SECRSMP::={noSecurity | octString, ...}

noSecurity      SECRSMP::={&ref c-noSecurity, &SecRSMP NullType}
octString       SECRSMP::={&ref c-octString, &SecRSMP OctStringSec}
```

```
OctStringSec::=OCTET STRING (SIZE(0..65535))

SecHeader::=SEQUENCE{
   secRef       SECRSMP.&ref({SecRSMPs}),
   secHead      SECRSMP.&SecRSMP({SecRSMPs}{@secRef})
   }

SecTrailer::=SEQUENCE{
   secRef       SECRSMP.&ref({SecRSMPs}),
   secTrail     SECRSMP.&SecRSMP({SecRSMPs}{@secRef})
   }

-- RSMP-Request and RSMP-Response common parts
-- RSMP PDU-ID
RsmpPDUcounter::=INTEGER(0..65535) -- cyclic counter

RsmpSessionID::=INTEGER(0..65535) -- cyclic counter

RSMPPDU::=CLASS{
   &ref         RefRSMPPDU, -- management request type identifier
   &RSMPpdu
   }

RefRSMPPDU::=INTEGER{
   c-requestPDU   (0),
   c-responsePDU  (1)
   } (0..255)

RSMPpdus        RSMPPDU::={requestPDU | responsePDU, ...}

requestPDU      RSMPPDU::={&ref c-requestPDU, &RSMPpdu RequestData}
responsePDU     RSMPPDU::={&ref c-responsePDU, &RSMPpdu ResponseData}

RSMPmessage::=SEQUENCE{
   sessionID    RsmpSessionID,
   pduCounter   RsmpPDUcounter,
   pduID        RSMPPDU.&ref({RSMPpdus}),
   pdu          RSMPPDU.&RSMPpdu({RSMPpdus}{@pduID})
   }

-- RSMP-Request

RSMPREQ::=CLASS{
   &ref         RefRSMPREQ, -- management request type identifier
   &RSMPrequest
   }

-- RqDataIDs
RefRSMPREQ::=INTEGER{
   c-pingRq           (0),
   c-rmcReqSession    (1),
   c-rmsInitSession   (2),
   c-rmsCloseSession  (3),
   c-getMparams       (4),
   c-setMparams       (5),
   c-firmwareUpdate   (6),
   c-protMaintenance  (7),
   c-appMaintenance   (8)
   } (0..255)

RSMPrequests RSMPREQ::={pingRq | rmcReqSession | rmsInitSession |
rmsCloseSession | getMparams | setMparams | firmwareUpdate |
protMaintenance | appMaintenance, ...}

pingRq          RSMPREQ::={&ref c-pingRq, &RSMPrequest RSMPping}
rmcReqSession   RSMPREQ::={&ref c-rmcReqSession, &RSMPrequest
                RMCreqSession}
rmsInitSession  RSMPREQ::={&ref c-rmsInitSession, &RSMPrequest
                RMSinitSession}
rmsCloseSession RSMPREQ::={&ref c-rmsCloseSession, &RSMPrequest
```

```
                    RMScloseSession}
getMparams          RSMPREQ::={&ref c-getMparams, &RSMPrequest
                    GetMparams}
setMparams          RSMPREQ::={&ref c-setMparams, &RSMPrequest
                    SetMparams}
firmwareUpdate     RSMPREQ::={&ref c-firmwareUpdate, &RSMPrequest
                    FirmwareUpdate}
protMaintenance     RSMPREQ::={&ref c-protMaintenance, &RSMPrequest
                    ProtMaintenance}
appMaintenance      RSMPREQ::={&ref c-appMaintenance, &RSMPrequest
                    AppMaintenance}

RSMPping::=OCTET STRING (SIZE(0..255))

-- RMC request for secure session initiation
-- Class to indicate reason for session initiation request by RMC
RMCREQREASON::=CLASS{
   &ref             RefRMCREQREASON, -- reason identifier
   &RMCreqReason
   }

-- IDs of reasons for session initiation request by RMC
RefRMCREQREASON::=INTEGER{
   c-rmcRqNoReason       (0), -- maybe RMS has news?
   c-rmcRqExceptLogFile (1), -- exception occurred. RMS should
                                read logfile.
   c-rmcRqRegulUpdate    (2) -- need regulatory info update.
   } (0..255)

RMCreqReasons RMCREQREASON::={rmcRqNoReason | rmcRqExceptLogFile |
rmcRqRegulUpdate, ...}

rmcRqNoReason RMCREQREASON::={&ref c-rmcRqNoReason, &RMCreqReason
NullType}
rmcRqExceptLogFile  RMCREQREASON::={&ref c-rmcRqExceptLogFile,
&RMCreqReason ExceptionID}
rmcRqRegulUpdate    RMCREQREASON::={&ref c-rmcRqRegulUpdate, &RMCreqReason
RegulUpdateRq}

ExceptionID::=INTEGER{
   exceptUnknown    (0) -- add privately specified exception codes
   }

RegulUpdateRq::=SEQUENCE{
   -- ID of regulatory issue
   lat      Latitude, -- latitude of RMC position
   lon      Longitude -- longitude of RMC position
   }

RMCreqReas::=SEQUENCE{
   rmcRqReasonRef   RMCREQREASON.&ref({RMCreqReasons}),
   reason RMCREQREASON.&RMCreqReason ({RMCreqReasons}{@rmcRqReasonRef})
   }

RMCreqSession::=SEQUENCE{
   itsscuID          ITSscuID, -- ITS-SCU-ID
   itsScuCmcID       ItsScuCmcID, -- unique ID of intended RMS
   reqReason         RMCreqReas, -- reason for session request
   prevSessionID     RsmpSessionID -- previous sessionID with intended
                     RMS
   }

RMSinitSession::=SEQUENCE{
   itsScuCmcID       ItsScuCmcID, -- unique ID of RMS
   itsscuID          ITSscuID, -- ITS-SCU-ID
   replyPort         PortNumber -- port number for reply by RMC
   }

RMScloseSession::=SEQUENCE{
   itsScuCmcID       ItsScuCmcID, -- unique ID of intended RMS
   itsscuID          ITSscuID -- ITS-SCU-ID
```

```
   }

GetMparams::=SEQUENCE OF RefMPARAM

SetMparams::=SEQUENCE OF Param24102

FirmwareUpdate::=SEQUENCE{
   firmware   OCTET STRING -- specific to management centre
   }

PROTMNT::=CLASS{
   &ref       RefMNT,
   &PROTmntc
   }

ProtMaintenance::=SEQUENCE{
   protMntRef      PROTMNT.&ref({ProtMntRqs}),
   protMgmtTask    PROTMNT.&PROTmntc({ProtMntRqs}{@protMntRef})
   }

RefMNT::=INTEGER{
   c-MntInstall   (0),
   c-MntUpdate    (1),
   c-MntDelete    (2)
   }

ProtMntRqs PROTMNT::={installProtRq | updateProtRq | deleteProtRq, ...}

installProtRq PROTMNT::={&ref c-MntInstall, &PROTmntc
ProtMgmtInstallRq}
updateProtRq PROTMNT::={&ref c-MntUpdate, &PROTmntc ProtMgmtUpdateRq}
deleteProtRq PROTMNT::={&ref c-MntDelete, &PROTmntc ProtMgmtDeleteRq}

ProtMgmtInstallRq::=SEQUENCE{
   protID       ITSprotID,    -- ITS protocol identifier
   itspoID      ITSpoID,      -- ITS protocol owner
   itsspdID     ITSSpdID,     -- ITS-S protocol developer
   itsspPr      ITSSpPr,      -- ITS-S protocol provisioner
   protCode     ProtCode      -- software to be installed
   }

ProtCode::=OCTET STRING

ProtMgmtUpdateRq::=SEQUENCE{
   protID       ProtocolID,   -- of ITS-S protocol to be updated
   itspoID      ITSpoID,      -- ITS protocol owner
   itsspdID     ITSSpdID,     -- ITS-S protocol developer
   itsspPr      ITSSpPr,      -- ITS-S protocol provisioner
   protPackage  ProtCode      -- software to be installed
   }

ProtMgmtDeleteRq::=ProtocolID -- of ITS-S protocol to be deleted

APPMNT::=CLASS{
   &ref   RefMNT,
   &APPmntc
   }

AppMaintenance::=SEQUENCE{
   appMntRef    APPMNT.&ref({AppMntRqs}),
   appMgmtTask  APPMNT.&APPmntc({AppMntRqs}{@appMntRef})
   }

AppMntRqs APPMNT::={installAppRq | updateAppRq | deleteAppRq, ...}

installAppRq APPMNT::={&ref c-MntInstall, &APPmntc AppMgmtInstallRq}
updateAppRq APPMNT::={&ref c-MntUpdate, &APPmntc AppMgmtUpdateRq}
deleteAppRq APPMNT::={&ref c-MntDelete, &APPmntc AppMgmtDeleteRq}

AppMgmtInstallRq::=SEQUENCE{
   appID        ITSaid,       -- ITS application identifier
```

```
   itssapid     ITSsapid,    -- ITS application process identifier
   itsaooID     ITSaooID,    -- ITS application object owner
   itssapdID    ITSSapdID,   -- ITS-S application process developer
   itssapPrPr   ITSSapPrPr,  -- ITS-S application process provisioner
   appPackage   AppCode      -- software to be installed
   }

AppMgmtUpdateRq::=SEQUENCE{
   appID        ITSsapiid,   -- ID of instance of ITS-S application
                                process
   itsaooID     ITSaooID,    -- ITS application object owner
   itssapdID    ITSSapdID,   -- ITS-S application process developer
   itssapPrPr   ITSSapPrPr,  -- ITS-S application process provisioner
   appCode      AppCode      -- software to be installed
   }

AppCode::=OCTET STRING

AppMgmtDeleteRq::=ITSsapiid, -- ID of instance of ITS-S application process

RequestData::=SEQUENCE{
   rsmpRqRef    RSMPREQ.&ref({RSMPrequests}),
   request      RSMPREQ.&RSMPrequest({RSMPrequests}{@rsmpRqRef})
   }
```

-- RSMP-Response

```
RSMPRES::=CLASS{
   &ref         RefRSMPRES, -- management response type identifier
   &RSMPresponse
   }

-- RsDataIDs
RefRSMPRES::=INTEGER{
   c-pingRs              (0),
   c-nullResponse        (1), -- RMCrequestSession is without response
   c-rmsInitSessionRs    (2),
   c-rmsCloseSessionRs   (3),
   c-getMparamsRs        (4),
   c-setMparamsRs        (5),
   c-firmwareUpdateRs    (6),
   c-protMaintenanceRs   (7),
   c-appMaintenanceRs    (8)
   }(0..255)

RSMPresponses RSMPRES::={pingRs | nullResponse | rmsInitSessionRs |
rmsCloseSessionRs | getMparamsRs | setMparamsRs | firmwareUpdateRs |
protMaintenanceRs | appMaintenanceRs, ...}

pingRs               RSMPRES::={&ref c-pingRs, &RSMPresponse RSMPping}
nullResponse         RSMPRES::={&ref c-nullResponse, &RSMPresponse
                     NullType}
rmsInitSessionRs     RSMPRES::={&ref c-rmsInitSessionRs, &RSMPresponse
                     RMSinitSessionRs}
rmsCloseSessionRs    RSMPRES::={&ref c-rmsCloseSessionRs, &RSMPresponse
                     RMScloseSessionRs}
getMparamsRs         RSMPRES::={&ref c-getMparamsRs, &RSMPresponse
                     GetMparamsRs}
setMparamsRs         RSMPRES::={&ref c-setMparamsRs, &RSMPresponse
                     SetMparamsRs}
firmwareUpdateRs     RSMPRES::={&ref c-firmwareUpdateRs, &RSMPresponse
                     FirmwareUpdateRs}
protMaintenanceRs    RSMPRES::={&ref c-protMaintenanceRs, &RSMPresponse
                     ProtMaintenanceRs}
appMaintenanceRs     RSMPRES::={&ref c-appMaintenanceRs, &RSMPresponse
                     AppMaintenanceRs}

RMSinitSessionRs::=SEQUENCE{
   itsScuCmcID       ItsScuCmcID, -- unique ID of RMS
   itsscuID          ITSscuID, -- ITS-SCU-ID
   resultStatus      RSMPErrStatus -- (success / rejected)
```

```
   }

RMScloseSessionRs::=SEQUENCE{
   itsScuCmcID       ItsScuCmcID, -- unique ID of intended RMS
   itsscuID          ITSscuID, -- ITS-SCU-ID
   resultStatus      RSMPErrStatus -- (success / rejected)
   }

GetMparamsRs::=SEQUENCE OF Param24102

SetMparamsRs::=SEQUENCE{
   globalStat        RSMPErrStatus, -- success or setErrorGeneral
   detailStat        DetailStatusSetMparams -- present in case of errors
   }

DetailStatusSetMparams::=SEQUENCE (SIZE(0..255)) OF SetMparamStatus

SetMparamStatus::=SEQUENCE{
   paramNo           RefMPARAM,
   resultCode        RSMPErrStatus
   }

FirmwareUpdateRs::=SEQUENCE{
   resultStatus      RSMPErrStatus -- (success / rejected)
   }

ProtMaintenanceRs::=SEQUENCE{
   protMntRef        PROTMNT.&ref({ProtMntRss}),
   protMgmtTask      PROTMNT.&PROTmntc({ProtMntRss}{@protMntRef}),
   resultStatus      RSMPErrStatus -- (success /
                     accessViolation/protUnknown)
   }

ProtMntRss PROTMNT::={installProtRs | updateProtRs | deleteProtRs, ...}

installProtRs PROTMNT::={&ref c-MntInstall, &PROTmntc ProtMgmtInstallRs}
updateProtRs PROTMNT::={&ref c-MntUpdate, &PROTmntc ProtMgmtUpdateRs}
deleteProtRs PROTMNT::={&ref c-MntDelete, &PROTmntc ProtMgmtDeleteRs}

ProtMgmtInstallRs::=ProtocolID -- with instance = 0 in case of failure

ProtMgmtUpdateRs::=ProtocolID -- with instance = 0 in case of failure

ProtMgmtDeleteRs::=ProtocolID -- with instance = 0 in case of success

AppMaintenanceRs::=SEQUENCE{
   appMntRef         APPMNT.&ref({AppMntRss}),
   appMgmtTask       APPMNT.&APPmntc({AppMntRss}{@appMntRef}),
   resultStatus      RSMPErrStatus -- (success /
                     accessViolation/appUnknown)
   }

AppMntRss APPMNT::={installAppRs | updateAppRs | deleteAppRs, ...}

installAppRs APPMNT::={&ref c-MntInstall, &APPmntc AppMgmtInstallRs}
updateAppRs APPMNT::={&ref c-MntUpdate, &APPmntc AppMgmtUpdateRs}
deleteAppRs APPMNT::={&ref c-MntDelete, &APPmntc AppMgmtDeleteRs}

AppMgmtInstallRs::=ITSsapiid -- with instance = 0 in case of failure

AppMgmtUpdateRs::=ITSsapiid -- with instance = 0 in case of failure

AppMgmtDeleteRs::=ITSsapiid -- with instance = 0 in case of success

ResponseData::=SEQUENCE{
   rsmpRsRef         RSMPRES.&ref({RSMPresponses}),
   response          RSMPRES.&RSMPresponse({RSMPresponses}{@rsmpRsRef}),
   error             RSMPErrStatus
   }

-- Error Status in RSMP - share values with ErrStatus in ISO 24102-3
```

```
RSMPErrStatus::=INTEGER{
   rsmpErrSuccess            (0), -- = accepted
   rsmpErrUnspecFailure      (1),
   rsmpErrParamNoUnknown     (2),
   rsmpErrAccessViolation    (3),
   rsmpErrRejected           (64),
   rsmpErrPduUnknown         (65),
   rsmpErrSetErrorGeneral    (66),
   rsmpErrAppUnknown         (67),
   rsmpErrProtUnknown        (68)
   } (0..255)

-- Complements SAP functions

-- MF-SAP Command.request --
-- send an RSMP message
RSMPmessageTX::=SEQUENCE{
   rsmpRequest      RSMPmessage
   }

RSMPmessageTXconf::=NullType

-- MF-SAP Request.request --
-- receive an RSMP message
RSMPmessageRX::=SEQUENCE{
   rsmpRequest      RSMPmessage
   }

RSMPmessageRXconf::=NullType

-- SF-SAP Request.request --
SecRSMPmessageTX::=SEQUENCE{
   rsmpRequestTX    RSMPmessageTX
   }

SecRMCHrX::=SEQUENCE{
   rmchRX           RMCHmessage -- something (request or response)
                    is received
   }

RMCHmessage::=SEQUENCE{
   header           SecHeader,
   rmchMsg          OCTET STRING, -- result of security procedure
   trailer          SecTrailer
   }

-- Management parameters Param24102 --
-- extends ASN.1 module from ISO 24102-1

RSMPtimeout::=TimeDurationValue

/*
   The ASN.1 specification has been checked for conformance to the
   ASN.1
   standards by OSS ASN.1 Syntax Checker, and by OSS
   ASN-1STEP
*/

END

-- Dynamic data to be requested properly

rsmpMessageTX       MFSAP-CR::={&mxref c-rsmpMessageTX, &MXParam
                    RSMPmessageTX}

c-rsmpMessageTX     RefMFSAP-CR::=<unique number 'a' to be assigned
                    in ISO 24102-3>

rsmpMessageTXconf   MFSAP-CC::={&mxref c-rsmpMessageTX, &MXParam
                    RSMPmessageTXconf}
```

```
rsmpMessageRX          MFSAP-RR::={&mxref c-rsmpMessageRX, &MXParam
                       RSMPmessageRX}

c-rsmpMessageRX        RefMFSAP-RR::=::=<unique number 'b' to be
                       assigned in ISO 24102-3>

rsmpMessageRXconf      MFSAP-RC::={&mxref c-rsmpMessageRX, &MXParam
                       RSMPmessageRXconf}


secRSMPmessageTX       SFSAP-RR::={&mxref c-secRSMPmessageTX,
                       &MXParam SecRSMPmessageTX}
secRMCHRX              SFSAP-RR::={&mxref c-secRMCHrX, &MXParam
                       SecRMCHrX}

c-secRSMPmessageTX     RefSFSAP-RR::=<unique number 'c' to be assigned
                       in ISO 24102-3>

c-secRMCHrX            RefSFSAP-RR::=<unique number 'd' to be assigned
                       in ISO 24102-3>
```

-- SF-SAP Request.conf --

```
secRSMPmessageTXConf     SFSAP-RC::={&mxref c-secRSMPmessageTX,
                         &MXParam RMCHmessage}

secRMCHrxConf            SFSAP-RC::={&mxref c-secRMCHrX, &MXParam
                         RSMPmessage}

c-RSMPtimeout            RefMPARAM::=12

rsmpTimeout              MPARAM::={&paramRef c-RSMPtimeout, &Parameter
                         RSMPtimeout}
```

Annex C
(informative)

Communication service parameters

Table C.1 illustrates the communication service parameters specified in ISO 17423 applicable for RSMP in the direct server initiation mode and in the client initiation mode. Communication service parameters applicable for the FSAP-based server initiation are those applicable for FSAP specified in ISO 22418[3].

Table C.1 — Communication service parameters

Communication service parameter	Value	Comment
Operational Communication service parameters		
CSP_LogicalChannelType	RsmpLCH	ITS-SCU configuration management channel
CSP_SessionCont	TRUE	Continuous connectivity needed for management session.
CSP_AvgADUrate	n.a.	No repetition
CSP_FlowType	n.a.	No well-known flow type presented
CSP_MaxPrio	am allowed value	As allowed by regulation

CSP_PortNo	RMS direct session request by RMC:
	{ receive; PORT_RSMP}
	else
	{ receive; PORT_DYN of RMC}
	RMS direct session init:
	{ transmit; PORT_RSMP}
	else:
	{ transmit; PORT_DYN}
	
	RMC direct session request:
	{ transmit; PORT_RSMP}
	else
	{ transmit; PORT_DYN}
	RMC direct session initialisation by RMS:
	{ receive; PORT_RMS}
	else
	{ receive; PORT_DYN}	

CSP_ExpFlowLifetime	value to be defined by implementation	Indicating the expected necessary time of a remote management session
Destination Communication service parameters		
CSP_DestinationType	4	Address based, unicast
CSP_DestinationDomain	16 2	Global for direct remote management sessions Local for usage with FSAP
CSP_CommDistance	n.a.	

Table C.1 *(continued)*

Communication service parameter	Value	Comment
CSP_Directivity	n.a.	
Performance Communication service parameters		
CSP_Resilience	required	completeness of message
CSP_MinThP	n.a. or tbd by implementation	
CSP_MaxLat	n.a. or tbd by implementation	
CSP_MaxADU	<max message size>	Depends on management task. May require fragmentation and TCP-like transport protocol to ensure completeness of fragments.
Security Communication service parameters		
CSP_DataConfidentiality	n.a.	Performed by RMCH
CSP_DataIntegrity	n.a.	Performed by RMCH
CSP_NonPepudiation	n.a.	Performed by RMCH
CSP_SourceAuthentication	n.a.	Performed by RMCH
Protocol Communication service parameters		
CSP_Protocol	n.a. or tbd by implementation	
CSP_SpecificCommsProts	n.a. or tbd by implementation	

Annex D
(normative)

Implementation conformance statement (ICS) proforma

Users of this document may

— freely reproduce the ICS proforma in this annex so that it can be used for its intended purposes and

— may further publish the completed ICS.

D.1 Guidance for completing the ICS proforma

D.1.1 Purposes and structure

The purpose of this Implementation Conformance Statement (ICS) proforma is to provide a mechanism whereby a supplier of an implementation of the requirements defined in this document may provide information about the implementation in a standardized manner.

The ICS proforma is subdivided into clauses for the following categories of information:

— guidance for completing the ICS proforma;

— identification of the implementation;

— identification of the implementation;

— global statement of conformance.

D.1.2 Abbreviations and conventions

The ICS proforma contained in this annex is comprised of information in tabular form in accordance with the guidelines presented in ISO/IEC 9646-7[6].

Item column	The item column contains a number which identifies the item in the table.
Item description column	The item description column describes in free text each respective item (e.g. parameters).
Status column	The notations defined in ISO/IEC 9646-7[6] are used for the status column: m mandatory – the capability is required to be supported. o optional – the capability may be supported or not. n/a not applicable – in the given context, it is impossible to use the capability. x prohibited (excluded) – there is a requirement not to use this capability in the given context. o.i qualified optional – for mutually exclusive or selectable options from a set. "i" is an integer which identifies an unique group of related optional items and the logic of their selection which is defined immediately following the table.

	ci conditional – the requirement on the capability ("m", "o", "x" or "n/a") depends on the support of other optional or conditional items. "i" is a string containing the respective Table number, followed by a dash, followed by a sequential number identifying a unique conditional status expression which is defined immediately following the respective Table. r as specified in the related referenced standard of the CI access technology.
Reference column	The reference column makes reference to this document, except where explicitly stated otherwise.
Support column	The support column shall be filled in by the supplier of the implementation. The following common notations, defined in ISO/IEC 9646-7[6], are used for the support column: Y or y supported by the implementation. N or n not supported by the implementation. N/A, n/a, or no answer required (allowed only if the status is n/a, directly or after evaluation of a conditional status).
Values allowed column	The values allowed column contains the type, the list, the range, or the length of values allowed. The following notations are used: — range of values: <min value> .. <max value> — list of values: <value1>, <value2>, ..., <valueN> — list of named values: <name1>(<val1>), <name2>(<val2>), ..., <nameN>(<valN>) — length: size (<min size> .. <max size>)
Values supported column	The values supported column shall be filled in by the supplier of the implementation. In this column, the values or the ranges of values supported by the implementation shall be indicated.
References to items	For each possible item answer (answer in the support column) within the ICS proforma a unique reference exists, used, for example, in the conditional expressions. It is defined as the table identifier, followed by a solidus character "/", followed by the item number in the table. If there is more than one support column in a table, the columns are discriminated by letters (a, b, etc.), respectively.
Prerequisite line	A prerequisite line takes the form: Prerequisite: <predicate>. A prerequisite line after a clause or table title indicates that the whole clause or the whole table is not required to be completed if the predicate is FALSE.

D.1.3 Instructions for completing the ICS proforma

The supplier of the implementation shall complete the ICS proforma. In particular, an explicit answer shall be entered using the notation described in D.1.2.

D.2 Identification of the Implementation

Identification of the Implementation Under Test (IUT) and the system in which it resides, i.e. the System Under Test (SUT), shall be filled in so as to provide as much detail as possible regarding version numbers and configuration options.

The product supplier information and client information shall both be filled in if they are different.