

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Alarm and electronic security systems –
Part 11-33: Electronic access control systems – Access control configuration
based on Web services**

**Systèmes d'alarme et de sécurité électroniques –
Partie 11-33: Systèmes de contrôle d'accès électronique – Configuration du
contrôle d'accès en fonction des services Web**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2021 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, ...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC online collection - oc.iec.ch

Discover our powerful search engine and read freely all the publications previews. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org

The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 18 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC -

webstore.iec.ch/advsearchform

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, ...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC online collection - oc.iec.ch

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org

Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Alarm and electronic security systems –
Part 11-33: Electronic access control systems – Access control configuration
based on Web services**

**Systèmes d'alarme et de sécurité électroniques –
Partie 11-33: Systèmes de contrôle d'accès électronique – Configuration du
contrôle d'accès en fonction des services Web**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 13.320

ISBN 978-2-8322-1011-4

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	8
INTRODUCTION.....	10
1 Scope.....	11
2 Normative references	11
3 Terms and definitions	12
4 Overview	15
4.1 General.....	15
4.2 Namespaces	16
4.3 Error handling	17
5 Credential service.....	17
5.1 General.....	17
5.2 Service capabilities	18
5.2.1 General	18
5.2.2 ServiceCapabilities data structure.....	18
5.2.3 GetServiceCapabilities command	19
5.3 Credential information.....	20
5.3.1 General	20
5.3.2 Data structures	20
5.3.3 GetCredentialInfoList command.....	23
5.3.4 GetCredentials command	24
5.3.5 GetCredentialList command.....	25
5.3.6 CreateCredential command	26
5.3.7 SetCredential command.....	28
5.3.8 ModifyCredential command.....	30
5.3.9 DeleteCredential command.....	31
5.3.10 GetCredentialState command	32
5.3.11 EnableCredential command	32
5.3.12 DisableCredential command	33
5.3.13 ResetAntipassbackViolation command.....	33
5.3.14 GetSupportedFormatTypes command.....	34
5.3.15 GetCredentialIdentifiers command	34
5.3.16 SetCredentialIdentifier command	35
5.3.17 DeleteCredentialIdentifier command	36
5.3.18 GetCredentialAccessProfiles command	36
5.3.19 SetCredentialAccessProfiles command.....	37
5.3.20 DeleteCredentialAccessProfiles command.....	37
5.4 Notification topics	38
5.4.1 General	38
5.4.2 Event overview (informative).....	38
5.4.3 Status changes.....	38
5.4.4 Configuration changes	39
6 Access rules service.....	40
6.1 General.....	40
6.2 Service capabilities	41
6.2.1 General	41
6.2.2 ServiceCapabilities data structure.....	41

6.2.3	GetServiceCapabilities command	41
6.3	Access profile information	41
6.3.1	General	41
6.3.2	Data structures	42
6.3.3	GetAccessProfileInfo command	42
6.3.4	GetAccessProfileInfoList command	43
6.3.5	GetAccessProfiles command	44
6.3.6	GetAccessProfileList command	45
6.3.7	CreateAccessProfile command	46
6.3.8	SetAccessProfile command	47
6.3.9	ModifyAccessProfile command	48
6.3.10	DeleteAccessProfile command	49
6.4	Notification topics	50
6.4.1	General	50
6.4.2	Event overview (informative)	50
6.4.3	Configuration changes	50
7	Authentication behaviour service	51
7.1	General	51
7.2	Example	51
7.3	Service capabilities	52
7.3.1	General	52
7.3.2	ServiceCapabilities data structure	52
7.3.3	GetServiceCapabilities command	53
7.4	Authentication profile information	53
7.4.1	General	53
7.4.2	Data structures	54
7.4.3	GetAuthenticationProfileInfo command	55
7.4.4	GetAuthenticationProfileInfoList command	56
7.4.5	GetAuthenticationProfiles command	57
7.4.6	GetAuthenticationProfileList command	58
7.4.7	CreateAuthenticationProfile command	59
7.4.8	SetAuthenticationProfile command	60
7.4.9	ModifyAuthenticationProfile command	61
7.4.10	DeleteAuthenticationProfile command	62
7.5	Security level information	63
7.5.1	General	63
7.5.2	Data structures	64
7.5.3	GetSecurityLevelInfo command	66
7.5.4	GetSecurityLevelInfoList command	66
7.5.5	GetSecurityLevels command	67
7.5.6	GetSecurityLevelList command	68
7.5.7	CreateSecurityLevel command	69
7.5.8	SetSecurityLevel command	70
7.5.9	ModifySecurityLevel command	71
7.5.10	DeleteSecurityLevel command	72
7.6	Notification topics	73
7.6.1	General	73
7.6.2	Event overview (informative)	73
7.6.3	Configuration changes	73

8	Schedule service	74
8.1	General.....	74
8.2	Recurrence	76
8.2.1	General	76
8.2.2	Weekly recurrence.....	76
8.2.3	Extended recurrence	77
8.2.4	Standard schedule recurrence	77
8.2.5	Special day recurrence	77
8.3	Service capabilities	78
8.3.1	General	78
8.3.2	ServiceCapabilities data structure.....	78
8.3.3	GetServiceCapabilities command	79
8.4	Schedule information	79
8.4.1	General	79
8.4.2	Data structures	79
8.4.3	GetScheduleInfo command	82
8.4.4	GetScheduleInfoList command	83
8.4.5	GetSchedules command	84
8.4.6	GetScheduleList command	85
8.4.7	CreateSchedule command.....	86
8.4.8	SetSchedule command	87
8.4.9	ModifySchedule command	88
8.4.10	DeleteSchedule command	89
8.5	Special day group information.....	90
8.5.1	General	90
8.5.2	Data structures	90
8.5.3	GetSpecialDayGroupInfo command	90
8.5.4	GetSpecialDayGroupInfoList command.....	91
8.5.5	GetSpecialDayGroups command	92
8.5.6	GetSpecialDayGroupList command.....	93
8.5.7	CreateSpecialDayGroup command	94
8.5.8	SetSpecialDayGroup command	95
8.5.9	ModifySpecialDayGroup command	96
8.5.10	DeleteSpecialDayGroup command.....	97
8.6	Schedule status	97
8.6.1	ScheduleState data structure.....	97
8.6.2	GetScheduleState command.....	98
8.7	Notification topics	99
8.7.1	General	99
8.7.2	Event overview (informative).....	99
8.7.3	Status changes.....	99
8.7.4	Configuration changes	100
8.8	Examples	101
8.8.1	General	101
8.8.2	Access 24 × 7 for admin staff.....	101
8.8.3	Access on Monday and Wednesday from 06:00 to 20:00 for cleaning staff	101
8.8.4	Access from Friday 18:00 to 07:00 for maintenance staff.....	101
8.8.5	Access on weekdays from 08:00 to 17:00 for employees	101

8.8.6	Access from January 15, 2014, to January 14, 2015, from 09:00 to 18:00	103
8.8.7	Special days example 1	103
8.8.8	Special days example 2	104
8.8.9	Special days example 3	106
Annex A (normative)	Access control interface XML schemata	107
A.1	Credential service WSDL	107
A.2	Access rules service WSDL	127
A.3	Authentication behaviour service WSDL	137
A.4	Schedule service WSDL	155
Annex B (informative)	Mapping of mandatory functions in IEC 60839-11-1	174
Bibliography		182
Figure 1	– Overview of service dependencies	16
Figure 2	– Main data structures in the credential service	18
Figure 3	– Main data structures in the access rules service	40
Figure 4	– Multiple schedules per access point	46
Figure 5	– Result of schedule union	47
Figure 6	– Authentication behaviour example	52
Figure 7	– Related objects of an authentication profile	54
Figure 8	– Related objects of a security level	63
Figure 9	– Security level examples	65
Figure 10	– Main data structures in the schedule service	74
Figure 11	– Recurrence support matrix	76
Figure 12	– Recurring events with an exception	77
Figure 13	– Recurring events with a special day	78
Figure 14	– SpecialDaysSchedule example	81
Figure 15	– Example of special day with time part	81
Figure 16	– Schedule states	98
Table 1	– Defined namespaces in this document	16
Table 2	– Referenced namespaces (with prefix)	16
Table 3	– GetServiceCapabilities command	19
Table 4	– GetCredentialInfo command	23
Table 5	– GetCredentialInfoList command	24
Table 6	– GetCredentials command	25
Table 7	– GetCredentialList command	26
Table 8	– CreateCredential command	27
Table 9	– SetCredential command	29
Table 10	– ModifyCredential command	31
Table 11	– DeleteCredential command	32
Table 12	– GetCredentialState command	32
Table 13	– EnableCredential command	33
Table 14	– DisableCredential command	33

Table 15 – ResetAntipassbackViolation command	34
Table 16 – GetSupportedFormatTypes command.....	34
Table 17 – GetCredentialIdentifiers command.....	35
Table 18 – SetCredentialIdentifier command.....	35
Table 19 – DeleteCredentialIdentifier command	36
Table 20 – GetCredentialAccessProfiles command	36
Table 21 – SetCredentialAccessProfiles command	37
Table 22 – DeleteCredentialAccessProfiles command.....	38
Table 23 – GetServiceCapabilities command	41
Table 24 – GetAccessProfileInfo command.....	43
Table 25 – GetAccessProfileInfoList command	44
Table 26 – GetAccessProfiles command	45
Table 27 – GetAccessProfileList command	46
Table 28 – CreateAccessProfile command	47
Table 29 – SetAccessProfile command	48
Table 30 – ModifyAccessProfile command	49
Table 31 – DeleteAccessProfile command	50
Table 32 – Example of schedule state to security level mapping	51
Table 33 – GetServiceCapabilities command	53
Table 34 – GetAuthenticationProfileInfo command	56
Table 35 – GetAuthenticationProfileInfoList command	57
Table 36 – GetAuthenticationProfiles command	58
Table 37 – GetAuthenticationProfileList command	59
Table 38 – CreateAuthenticationProfile command	60
Table 39 – SetAuthenticationProfile command	61
Table 40 – ModifyAuthenticationProfile command	62
Table 41 – DeleteAuthenticationProfile command	63
Table 42 – GetSecurityLevelInfo command	66
Table 43 – GetSecurityLevelInfoList command.....	67
Table 44 – GetSecurityLevels command	68
Table 45 – GetSecurityLevelList command	69
Table 46 – CreateSecurityLevel command	70
Table 47 – SetSecurityLevel command	71
Table 48 – ModifySecurityLevel command	72
Table 49 – DeleteSecurityLevel command	72
Table 50 – GetServiceCapabilities command	79
Table 51 – GetScheduleInfo command.....	83
Table 52 – GetScheduleInfoList command	84
Table 53 – GetSchedules command.....	85
Table 54 – GetScheduleList command.....	86
Table 55 – CreateSchedule command.....	87
Table 56 – SetSchedule command.....	88
Table 57 – ModifySchedule command.....	89

Table 58 – DeleteSchedule command	89
Table 59 – GetSpecialDayGroupInfo command	91
Table 60 – GetSpecialDayGroupInfoList command	92
Table 61 – GetSpecialDayGroups command	93
Table 62 – GetSpecialDayGroupList command	94
Table 63 – CreateSpecialDayGroup command	95
Table 64 – SetSpecialDayGroup command	96
Table 65 – ModifySpecialDayGroup command	97
Table 66 – DeleteSpecialDayGroup command	97
Table 67 – GetScheduleState command	99
Table B.1 – Access point interface requirements.....	175
Table B.2 – Indication and annunciation requirements	176
Table B.3 – Recognition requirements	179
Table B.4 – Duress signalling requirements	180
Table B.5 – Overriding requirements.....	181
Table B.6 – System self protection requirements	181

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ALARM AND ELECTRONIC SECURITY SYSTEMS –**Part 11-33: Electronic access control systems –
Access control configuration based on Web services**

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 60839-11-33 has been prepared by IEC technical committee 79: Alarm and electronic security systems. It is an International Standard.

The text of this International Standard is based on the following documents:

FDIS	Report on voting
79/646/FDIS	79/648/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/standardsdev/publications.

A list of all parts in the IEC 60839 series, published under the general title *Alarm and electronic security systems*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The "colour inside" logo on the cover page of this document indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

This document makes it possible to build an alarm and electronic security system with clients, typically a monitoring console, and devices, typically an access control unit, from different manufacturers using common and well defined interfaces.

The document specifies only the data and control flow between a client and the services without reference to any physical device as the services required to implement a compliant electronic access control system (EACS) are not necessarily implemented on a single device, i.e. all services can be run on a control panel, event aggregator software on PC, etc.

This document does not define internal communication between an access control unit and its components if they are implemented on a single device.

This document is based upon work done by the ONVIF open industry forum. The ONVIF Credential specification, ONVIF Access Rules specification, ONVIF Authentication Behaviour specification and ONVIF Schedule specification are compatible with this document.

This document is accompanied by a set of computer readable interface definitions (see Annex A):

- credential service WSDL, see Clause A.1;
- access rules service WSDL, see Clause A.2;
- authentication behaviour service WSDL, see Clause A.3;
- schedule service WSDL, see Clause A.4.

Due to the differences in terminology used in IEC 60839-11-1:2013 and IEC 60839-11-2:2014 and the ONVIF specification that this part of IEC 60839 is based on, a reader should take special notice of the terms and definitions clause.

Additional services needed for monitoring of doors and access points (portal sides) are outside the scope of this document. These services are covered by IEC 60839-11-32.

ALARM AND ELECTRONIC SECURITY SYSTEMS –

Part 11-33: Electronic access control systems – Access control configuration based on Web services

1 Scope

This part of IEC 60839 defines the Web services interface for electronic access control systems. This includes listing electronic access control system components, their logical composition, monitoring their states and controlling them. It also includes a mapping of mandatory and optional requirements in accordance with IEC 60839-11-1:2013, as covered by Annex B.

This document applies to physical security only. Physical security prevents unauthorized personnel, attackers or accidental intruders from physically accessing a building, room, etc.

Web services usage and device management functionality are outside the scope of this document. Refer to IEC 60839-11-31:2016 for more information.

This document does not in any way limit a manufacturer to add other protocols or extend the protocol defined here. For rules on how to accomplish this, refer to IEC 60839-11-31:2016.

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.

IEC 60839-11-1:2013, *Alarm and electronic security systems – Part 11-1: Electronic access control systems – System and components requirements*

IEC 60839-11-2:2014, *Alarm and electronic security systems – Part 11-2: Electronic access control systems – Application guidelines*

IEC 60839-11-31:2016, *Alarm and electronic security systems – Part 11-31: Electronic access control systems – Core interoperability protocol based on Web services*

IEC 60839-11-32:2016, *Alarm and electronic security systems – Part 11-32: Electronic access control systems – Access control monitoring based on Web services*

ISO 16484-5:2017, *Building automation and control systems (BACS) – Part 5: Data communication protocol*

RFC 5545, *Internet Calendaring and Scheduling Core Object Specification (iCalendar)*, (available at <https://tools.ietf.org/html/rfc5545>)

RFC 5234, *Augmented BNF for Syntax Specifications: ABNF*, (available at <https://tools.ietf.org/html/rfc5234>)

3 Terms and definitions

For the purposes of this document the terms and definitions given in IEC 60839-11-1, IEC 60839-11-2, IEC 60839-11-31 and IEC 60839-11-32 and the following apply.

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

- IEC Electropedia: available at <http://www.electropedia.org/>
- ISO Online browsing platform: available at <http://www.iso.org/obp>

NOTE When the IEC term defined in IEC 60839-11-1 and IEC 60839-11-2 differs from the terms used in this document, the IEC term will be given within parentheses in the section headers.

3.1

access level

set of rules used to determine where and when a credential has authorized access to one or more portals and which may include special passage conditions such as specific portal allowed open times

Note 1 to entry: In this document, the term "access rules" is used.

3.2

access point

portal

physical entrance/exit at which access can be controlled by a door, turnstile or other secure barrier

Note 1 to entry: For the purposes of this document the access point is considered to be a logical composition of a physical door and reader(s) controlling access in one direction.

Note 2 to entry: In this document, the term "door" has the same meaning as "access point" or "portal".

3.3

access policy

association of an access point (portal side) and a schedule (time zone)

Note 1 to entry: An access policy defines when an access point (portal side) can be accessed.

3.4

access profile

collection of access policies

Note 1 to entry: Access profiles are used to define "when" and "where" access will be granted.

3.5

anti-passback violation state

state indicating if the anti-passback rules have been violated for a credential

3.6

authentication policy

association of a security level and a schedule (time zone)

Note 1 to entry: An authentication policy defines when a security level is required at an access point (portal side).

3.7

authentication profile

collection of authentication policies

EXAMPLE All entrance access points (portal sides) are configured to require card access during office hours, card and PIN access during nighttime, and no access during holidays.

Note 1 to entry: Authentication profiles are used to define authentication behaviour for a type of access point (portal sides).

3.8

credential

information either memorized or held within a token with information either memorized, held within a token or uniquely identified biological (person) or physical (vehicle) trait of the user or credential holder

EXAMPLE A credential number can be stored in the physical credential (token). A PIN code is associated with the token, but will not necessarily be stored in the token. Both the credential number and the PIN code are credential identifiers that can be contained in the same credential.

Note 1 to entry: In this document, the term "credential" represents information that is used to identify a credential holder (user) at an access point (portal side). This information is not necessarily held within a token.

Note 2 to entry: In this document, several pieces of information (credential identifiers) can be part of the same credential if they belong together.

3.9

credential identifier

piece of data that identifies a credential holder (user) at an access point

EXAMPLE Card number, unique card information, PIN, fingerprint, or other biometric information.

Note 1 to entry: A credential identifier also specifies the type of identifier (recognition method) and the format of the value used in the Web service interface.

3.10

credential state

state indicating if a credential is enabled or disabled, or if anti-passback rules have been violated or not

Note 1 to entry: The state may also contain a reason why the credential was disabled.

3.11

format type

format of a credential identifier value in the Web services interface

Note 1 to entry: The format is not necessarily the same as when it is stored in a physical credential.

Note 2 to entry: ONVIF supports the BACnet format types listed in ISO 16484-5:2017.

3.12

iCalendar

industry standard format for exchanging scheduling and activity-recording information electronically

3.13

recognition group

logical OR between the recognition methods in a security level

EXAMPLE One recognition group contains the recognition methods pt:Card and pt:Fingerprint. Another group contains the recognition methods pt:Card and pt:Face. The resulting effect is that the access point will require either Card+Fingerprint, or Card+Face.

3.14 recognition method

method of recognizing a credential holder (user) at an access point (portal side)

EXAMPLE Information held within a physical credential (token) such as a credential number, memorized information such as a PIN, fingerprint, or other biometric information.

3.15 recognition type

recognition method or anonymous entry

3.16 security level

defined level of assurance to be granted access to an access point (portal side)

Note 1 to entry: A security level may be defined as individual recognition methods, combinations of recognition methods (using logical AND or OR), or no recognition methods (open).

3.17 special day group

set of days (or parts of days) that require the regular schedule (time zone) to be overridden

EXAMPLE Holidays, half-days or working Sundays.

3.18 special days' schedule

set of overriding time periods (time slots) for a special day group

3.19 time slot

interval of time between two given moments indicating the beginning and the end of a valid period within a time zone

Note 1 to entry: In this document, the term "time period" is used.

Note 2 to entry: In this document, time periods are only used in special days' schedules, i.e. the time slots that override a regular schedule during special days.

Note 3 to entry: Time periods do not have a date part.

3.20 time zone

one or more time slots combined with calendar information

EXAMPLE The schedule (time zone) "Working hours" can be defined as two time periods (time slots) valid on working days, one from 08:00 to 11:30, and one from 12:30 to 18:00. A set of special days' schedules can be defined for dates the regular time periods do not apply. Each special day's schedule can contain another set of time periods that will override the regular time periods for the special days.

Note 1 to entry: In this document, the term "schedule" is used.

3.21 token

portable device containing a readable unique identifier (credential) that can be associated with a user's data and access rights stored within the electronic access control system

Note 1 to entry: In this document, the term "physical credential" is used.

Note 2 to entry: In this document, the term "token" is the unique identifier of any managed entity.

3.22**user**

person requesting access through an access point

Note 1 to entry: In this document, the term "credential holder" is used.

Note 2 to entry: A credential holder is not necessarily a person; a credential may for instance be assigned to a vehicle; the vehicle credential shall be associated with a person.

3.23**vEvent**

component in iCalendar, specifying the properties of an event

4 Overview**4.1 General**

This document defines four Web service interfaces:

- credential service (see Clause 5);
- access rules service (see Clause 6);
- authentication behaviour service (see Clause 7);
- schedule service (see Clause 8).

The services above relate to the two following Web service interfaces defined in IEC 60839-11-32:2016:

- access control service (see IEC 60839-11-32:2016, Clause 5);
- door control service (see IEC 60839-11-32:2016, Clause 6).

Figure 1 shows how the six Web service interfaces relate to each other.

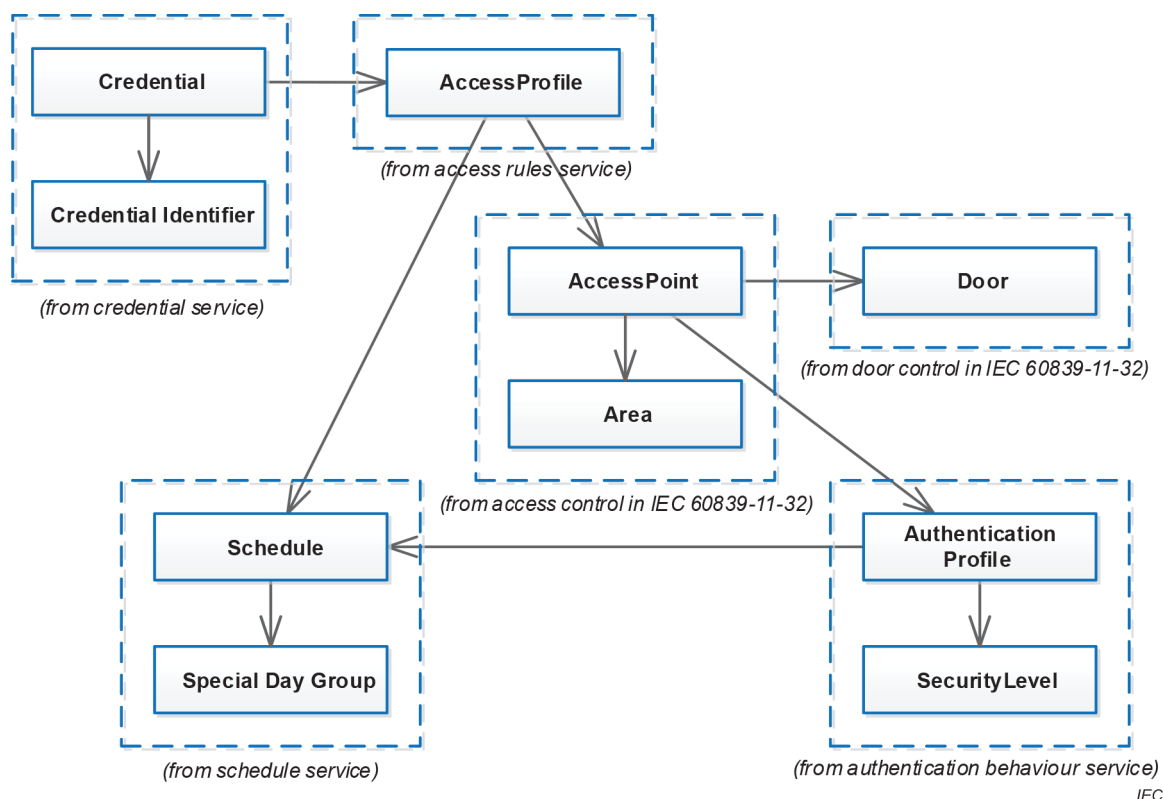


Figure 1 – Overview of service dependencies

Also refer to IEC 60839-11-32:2016, Clause 4, which is relevant for this document.

4.2 Namespaces

In addition to namespaces listed in IEC 60839-11-31:2016, 5.4, this document defines the namespaces listed in Table 1.

Table 1 – Defined namespaces in this document

Prefix	Namespace URI	Description
pt	https://www.onvif.org/ver10/pacs/types.xsd	XML schema descriptions in this document
tac	http://www.onvif.org/ver10/accesscontrol/wsdl	Namespace for the access control service
tdc	http://www.onvif.org/ver10/doorcontrol/wsdl	Namespace for the door control service
tcr	http://www.onvif.org/ver10/credential/wsdl	Namespace for the credential service
tar	http://www.onvif.org/ver10/accessrules/wsdl	Namespace for the access rules service
tab	http://www.onvif.org/ver10/authenticationbehaviour/wsdl	Namespace for the authentication behaviour service.
tsc	http://www.onvif.org/ver10/schedule/wsdl	Namespace for the schedule service

The namespaces listed in Table 2 are referenced by this document.

Table 2 – Referenced namespaces (with prefix)

Prefix	Namespace URI	Description
env	http://www.w3.org/2003/05/soap-envelope	Envelope namespace as defined by SOAP 1.2 [W3C SOAP12-PART11]

4.3 Error handling

Faults listed in this document are specific to each command. Each command may also return generic faults listed in IEC 60839-11-31:2016, 5.12.3.2.

5 Credential service

5.1 General

The credential service specification defines the commands to configure credentials.

A credential holds information that can be validated in an access point, such as unique card information, PIN, biometric information. A credential also holds information on what the credential can access via credential access profiles, which ties the credential to access profiles as described in Clause 6.

A credential is assigned to a person (called credential holder) and has a validity that specifies the period during which the credential can be used to get access.

Consider the following example:

A credential is assigned to a consultant to be temporarily used during one week (April 2nd to April 6th). The start date/time of the validity is set to the morning of April 2nd and the end date/time of the validity is set to the evening of April 6th.

This particular credential is a card with number 987654321 and the consultant is given the personal PIN code 1234. Both pieces of information are stored in the credential. The card number is a credential identifier of type pt:Card and the PIN code is a credential identifier of type pt:PIN (see 5.2.2 for supported identifier types).

The consultant will help out with the installation of a server so he needs access to the server room, and of course also access to the other common facilities at the office. The access profile "IT support access" gives access to the server room during office hours, and the access profile "Staff access" gives access to the rest of the office. The references to both access profiles are stored on the credential in the CredentialAccessProfile structure.

A credential service object model representation is shown in Figure 2.

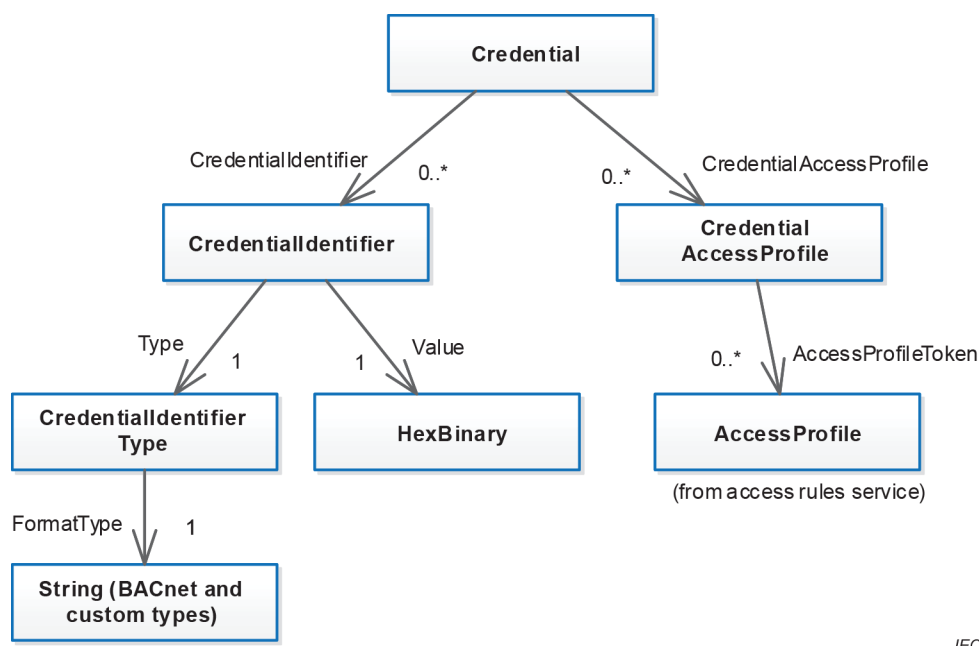


Figure 2 – Main data structures in the credential service

5.2 Service capabilities

5.2.1 General

The device shall provide service capabilities in two ways:

- 1) With the GetServices method of Device service when IncludeCapability is true. Refer to IEC 60839-11-31:2016, 8.1.
- 2) With the GetServiceCapabilities method.

5.2.2 ServiceCapabilities data structure

The service capabilities reflect the optional functionalities of a service. The information is static and does not change during the device operation. The following capabilities are available:

- **MaxLimit**
The maximum number of entries returned by a single Get<Entity>List or Get<Entity> request. The device shall never return more than this number of entities in a single response.
- **CredentialValiditySupported**
Indicates that the device supports credential validity.
- **CredentialAccessProfileValiditySupported**
Indicates that the device supports validity on the association between a credential and an access profile.
- **ValiditySupportsTimeValue**
Indicates that the device supports both date and time value for validity. If set to false, then the time value is ignored.
- **MaxCredentials**
The maximum number of credentials supported by the device.
- **MaxAccessProfilesPerCredential**
The maximum number of access profiles for a credential.
- **ResetAntipassbackSupported**
Indicates the device supports resetting of anti-passback violations and notifying on anti-passback violations.

- **SupportedIdentifierType**

A list of identifier types that the device supports. Identifiers types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined identifier types, free text can be used. The following types are defined by ONVIF:

- pt:Card Supports Card identifier type (including fob, tag and similar).
- pt:PIN Supports PIN identifier type.
- pt:Fingerprint Supports Fingerprint biometric identifier type.
- pt:Face Supports Face biometric identifier type.
- pt:Iris Supports Iris biometric identifier type.
- pt:Vein Supports Vein biometric identifier type.
- pt:Palm Supports Palm biometric identifier type.
- pt:Retina Supports Retina biometric identifier type.

- **SupportedExemptionType**

A list of exemptions that the device supports. Exemption types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined exemption types, free text can be used. The following types are defined by ONVIF:

- pt:ExemptFromAuthentication Supports ExemptedFromAuthentication in 5.3.2.3.

- **ClientSuppliedTokenSupported**

Indicates that the client is allowed to supply the token when creating credentials. To enable the use of the command SetCredential, the value shall be set to true.

- **DefaultCredentialSuspensionDuration**

The default time period that the credential will temporary be suspended (e.g. by using the wrong PIN a predetermined number of times).

The time period is defined as an [ISO 8601] duration string (e.g. "PT5M").

5.2.3 GetServiceCapabilities command

This operation returns the capabilities of the credential service.

A device shall support this command as described in Table 3.

Table 3 – GetServiceCapabilities command

GetServiceCapabilities		Access class: PRE_AUTH
Message name	Description	
GetServiceCapabilitiesRequest	This message shall be empty.	
GetServiceCapabilitiesResponse	This message contains: • "Capabilities": The capability response message contains the requested credential service capabilities using a hierarchical XML capability structure. tcr:ServiceCapabilities Capabilities [1][1]	
Fault codes	Description	
	No command specific faults.	

5.3 Credential information

5.3.1 General

A credential holds information that can be validated in an access point (card information, PIN, etc.) and what the credential can access (access profiles).

5.3.2 Data structures

5.3.2.1 CredentialInfo

The CredentialInfo type represents the credential as a logical object. The structure contains the basic information of a specific credential instance. The device shall provide the following fields for each credential.

- **token**
A service unique identifier of the credential.
- **CredentialHolderReference**
An external reference to a person holding this credential. The reference is a username or used ID in an external system, such as a directory service.

To provide more information, the device may include the following optional fields:

- **Description**
User readable description for the credential. It shall be up to 1 024 characters.
- **ValidFrom**
The start date/time validity of the credential. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).
- **ValidTo**
The expiry date/time validity of the credential. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).

5.3.2.2 Credential

A credential is a physical/tangible object, a piece of knowledge, or a facet of a person's physical being that enables an individual access to a given physical facility or computer-based information system. A credential holds one or more credential identifiers. To gain access one or more identifiers may be required.

The device shall include all properties of the CredentialInfo structure and also a list of credential identifiers and a list of credential access profiles.

- **CredentialIdentifier**
A list of credential identifier structures. At least one credential identifier is required. Maximum one credential identifier structure per type is allowed.

To provide more information, the device may include the following optional fields:

- **CredentialAccessProfile**
A list of credential access profile structures.
- **Attribute**
A list of credential attributes as name value pairs. Key name types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined key name types, free text can be used.
- **ExtendedGrantTime**
A boolean indicating that the credential holder needs extra time to get through the door. ExtendedReleaseTime will be added to ReleaseTime, and ExtendedOpenTime will be added to OpenTime.

5.3.2.3 CredentialIdentifier

A credential identifier is a card number, unique card information, PIN or biometric information such as fingerprint, iris, vein, face recognition, that can be validated in an access point.

- **Type**
Contains the details of the credential identifier type. Is of type CredentialIdentifierType.
- **ExemptedFromAuthentication**
If set to true, this credential identifier is not considered for authentication. For example if the access point requests Card plus PIN, and the credential identifier of type PIN is exempted from authentication, then the access point will not prompt for the PIN.
- **Value**
The value of the identifier in hexadecimal representation.

5.3.2.4 CredentialIdentifierType

Specifies the name of the credential identifier type and its format for the credential value.

- **Name**
The name of the credential identifier type, such as pt:Card, pt:PIN, etc. (see 5.2.2 for supported credential identifier types).
- **FormatType**
Specifies the format of the credential value for the specified identifier type name. See 5.3.2.5 below.

5.3.2.5 CredentialIdentifierFormatTypeInfo

Contains information about a format type.

- **FormatType**
A format type supported by the device. A list of supported format types is provided in ISO 16484-5:2017. The BACnet type "CUSTOM" is not used in this document. Instead device manufacturers can define their own format types.
- **Description**
User readable description of the credential identifier format type. It shall be up to 1 024 characters. For custom types, it is recommended to describe how the octet string is encoded (following the structure given in the column Authentication Factor Value Encoding of ISO 16484-5:2017, Table P-1).

5.3.2.6 CredentialAccessProfile

The association between a credential and an access profile. Refer to 6.3.2.2.

- **AccessProfileToken**
The reference token of the associated access profile.

The device may include the following optional fields:

- **ValidFrom**
The start date/time of the validity for the association between the credential and the access profile. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).
- **ValidTo**
The end date/time of the validity for the association between the credential and the access profile. If the ValiditySupportsTimeValue capability is set to false, then only the date is supported (the time is ignored).

5.3.2.7 CredentialState

The CredentialState structure contains information about the state of the credential and optionally the reason why the credential was disabled.

- **Enabled**

True if the credential is enabled or false if the credential is disabled.

The device may include the following optional fields:

- **Reason**

The reason for disabling the credential. Reason types starting with the prefix pt: are reserved to define ONVIF-specific types. For custom defined reason types, free text can be used. The following types are defined by ONVIF:

- pt:CredentialLockedOut Access is denied due to credential locked out.
- pt:CredentialBlocked Access is denied because the credential has deliberately been blocked by the operator.
- pt:CredentialLost Access is denied due to the credential being reported as lost.
- pt:CredentialStolen Access is denied due to the credential being reported as stolen.
- pt:CredentialDamaged Access is denied due to the credential being reported as damaged.
- pt:CredentialDestroyed Access is denied due to the credential being reported as destroyed.
- pt:CredentialInactivity Access is denied due to credential inactivity.
- pt:CredentialExpired Access is denied because the credential has expired.
- pt:CredentialRenewalNeeded Access is denied because the credential requires a renewal (e.g. new PIN or fingerprint enrollment).

- **AntipassbackState**

A structure indicating the anti-passback state. This field shall be supported if the ResetAntipassbackSupported capability is set to true.

5.3.2.8 Anti-passback State

A structure containing anti-passback related state information.

- **AntipassbackViolated**

Indicates if anti-passback is violated for the credential.

5.3.2.9 GetCredentialInfo command

This operation requests a list of CredentialInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 4.

Table 4 – GetCredentialInfo command

GetCredentialInfo		Access class: READ_SYSTEM
Message name	Description	
GetCredentialInfoRequest	<i>This message contains:</i> • "Token": Tokens of CredentialInfo items to get. pt:ReferenceToken Token [1][unbounded]	
GetCredentialInfoResponse	<i>This message contains:</i> • "CredentialInfo": List of CredentialInfo items. tcr:CredentialInfo CredentialInfo [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgs ter:TooManyItems	Too many items were requested, see MaxLimit capability.	

5.3.3 GetCredentialInfoList command

This operation requests a list of all CredentialInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 5.

Table 5 – GetCredentialInfoList command

GetCredentialInfoList		Access class: READ_SYSTEM
Message name	Description	
GetCredentialInfoListRequest	<i>This message contains:</i> • <i>"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.</i> • <i>"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.</i> xs:int Limit [0][1] xs:string StartReference [0][1]	
GetCredentialInfoListResponse	<i>This message contains:</i> • <i>"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.</i> • <i>"CredentialInfo": List of CredentialInfo items.</i> xs:string NextStartReference [0][1] tcr:CredentialInfo CredentialInfo [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:InvalidStartReference	<i>StartReference is invalid or has timed out. Client needs to start fetching from the beginning.</i>	

5.3.4 GetCredentials command

This operation requests a list of Credential items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 6.

Table 6 – GetCredentials command

GetCredentials		Access class: READ_SYSTEM_SECRET
Message name	Description	
GetCredentialsRequest	<i>This message contains:</i> • "Token": Tokens of Credentials to get. pt:ReferenceToken Token [1][unbounded]	
GetCredentialsResponse	<i>This message contains:</i> • "Credential": List of Credential items. tcr:Credential Credential [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgs ter:TooManyItems	Too many items were requested, see MaxLimit capability.	

5.3.5 GetCredentialList command

This operation requests a list of all Credential items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 7.

Table 7 – GetCredentialList command

GetCredentialList		Access class: READ_SYSTEM_SECRET
Message name	Description	
GetCredentialListRequest	<i>This message contains:</i> • <i>"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.</i> • <i>"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.</i> xs:int Limit [0][1] xs:string StartReference [0][1]	
GetCredentialListResponse	<i>This message contains:</i> • <i>"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.</i> • <i>"Credential": List of Credential items.</i> xs:string NextStartReference [0][1] tcr:Credential Credential [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:InvalidStartReference	<i>StartReference is invalid or has timed out. Client needs to start fetching from the beginning.</i>	

5.3.6 CreateCredential command

This operation creates the specified credential in the device.

A call to this method takes a Credential structure and a CredentialState structure as input parameters. The credential state can be created in disabled or enabled state.

The token field of the Credential structure shall be empty and the device shall allocate a token for the credential. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

A device shall support this command as described in Table 8.

Table 8 – CreateCredential command

CreateCredential		Access class: WRITE_SYSTEM
Message name	Description	
CreateCredentialRequest	<i>This message contains:</i> • "Credential": The credential to create. • "State": The state of the credential. tcr:Credential Credential [1][1] tcr:CredentialState State [1][1]	
CreateCredentialResponse	<i>This message contains:</i> • "Token": The token of the created credential. pt:ReferenceToken Token [1][1]	
Fault codes	Description	
env:Sender ter:CapabilityViolated ter:MaxCredentials	There is not enough space to create a new credential, see the MaxCredentials capability.	
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.	
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	Credential validity is not supported by the device, see the CredentialValiditySupported capability.	
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.	
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	Specified identifier type is not supported by the device, see the SupportedIdentifierType capability.	
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	The same identifier type was used more than once.	
env:Sender ter:InvalidArgVal ter:InvalidFormatType	Specified identifier format type is not supported by the device.	
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	Specified identifier value is not in accordance with the FormatType definition.	
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	The same combination of identifier type, format and value was used more than once (some devices may not support duplicate identifier values).	
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	A referred entity token is not found (some devices may not validate referred entities).	
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	Specified exemption type is not supported by the device, see the SupportedExemptionType capability.	

5.3.7 SetCredential command

This method is used to synchronize a credential in a client with the device.

A call to this method takes a Credential structure and a CredentialState structure as input parameters. The token field of the credential shall not be empty.

If a credential with the specified token does not exist in the device, the credential is created. The credential state can be created in disabled or enabled state.

If a credential with the specified token exists in the device, then the credential is modified. The credential state will be set according to the specified state (this behaviour differs from the ModifyCredential command). All existing credential identifiers and credential access profiles are removed and replaced with the specified entities.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 9.

Table 9 – SetCredential command

SetCredential		Access class: WRITE_SYSTEM
Message name	Description	
SetCredentialRequest	<i>This message contains:</i> • "Credential": The credential to create. • "State": The state of the credential. tcr:Credential Credential [1][1] tcr:CredentialState State [1][1]	
SetCredentialResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	<i>The device does not support that the client supplies the token.</i>	
env:Sender ter:CapabilityViolated ter:MaxCredentials	<i>There is not enough space to create a new credential, see the MaxCredentials capability.</i>	
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	<i>There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.</i>	
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	<i>Credential validity is not supported by the device, see the CredentialValiditySupported capability.</i>	
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	<i>Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.</i>	
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	<i>Specified identifier type is not supported by the device, see the SupportedIdentifierType capability.</i>	
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	<i>The same identifier type was used more than once.</i>	
env:Sender ter:InvalidArgVal ter:InvalidFormatType	<i>Specified identifier format type is not supported by the device.</i>	
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	<i>Specified identifier value is not in accordance with the FormatType definition.</i>	
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	<i>The same combination of identifier type, format and value was used more than once (some devices may not support duplicate identifier values).</i>	
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	<i>A referred entity token is not found (some devices may not validate referred entities).</i>	
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	<i>Specified exemption type is not supported by the device, see the SupportedExemptionType capability.</i>	

5.3.8 **ModifyCredential command**

This operation modifies the specified credential.

The token of the credential to modify is specified in the token field of the Credential structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified credential.

When an existing credential is modified, the state is not modified explicitly. The only way for a client to change the state of a credential is to explicitly call the EnableCredential, DisableCredential or ResetAntipassback command.

All existing credential identifiers and credential access profiles are removed and replaced with the specified entities.

A device shall support this command as described in Table 10.

Table 10 – ModifyCredential command

ModifyCredential		Access class: WRITE_SYSTEM
Message name	Description	
ModifyCredentialRequest	<i>This message contains:</i> "Credential": Details of the credential. tcr:Credential Credential [1][1]	
ModifyCredentialResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	<i>There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.</i>	
env:Sender ter:CapabilityViolated ter:CredentialValiditySupported	<i>Credential validity is not supported by the device, see the CredentialValiditySupported capability.</i>	
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	<i>Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.</i>	
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	<i>Specified identifier type not is supported by the device, see the SupportedIdentifierType capability.</i>	
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierType	<i>The same identifier type was used more than once.</i>	
env:Sender ter:InvalidArgVal ter:InvalidFormatType	<i>Specified identifier format type is not supported by the device.</i>	
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	<i>Specified identifier value is not in accordance with the FormatType definition.</i>	
env:Sender ter:InvalidArgVal ter:DuplicatedIdentifierValue	<i>The same combination of identifier type, format and value was used more than once (some devices may not support duplicate identifier values).</i>	
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	<i>A referred entity token is not found (some devices may not validate referred entities).</i>	
env:Sender ter:CapabilityViolated ter:SupportedExemptionType	<i>Specified exemption type is not supported by the device, see the SupportedExemptionType capability.</i>	

5.3.9 DeleteCredential command

This method deletes the specified credential.

If it is associated with one or more entities some devices may not be able to delete the credential, and consequently a ReferenceInUse fault shall be generated.

If no token was specified in the request, the device shall return InvalidArgs as a generic fault code.

A device shall support this command as described in Table 11.

Table 11 – DeleteCredential command

DeleteCredential		Access class: WRITE_SYSTEM
Message name	Description	
DeleteCredentialRequest	<i>This message contains:</i> "Token": The token of the credential to delete. pt:ReferenceToken Token [1][1]	
DeleteCredentialResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	

5.3.10 GetCredentialState command

This method returns the state for the specified credential.

If the capability ResetAntipassbackSupported is set to true, then the device shall supply the anti-passback state in the returned CredentialState structure.

A device shall support this command as described in Table 12.

Table 12 – GetCredentialState command

GetCredentialState		Access class: READ_SYSTEM_SENSITIVE
Message name	Description	
GetCredentialStateRequest	<i>This message contains:</i> "Token": Token of Credential. pt:ReferenceToken Token [1][1]	
GetCredentialStateResponse	<i>This message contains:</i> "State": State of the credential. tcr:CredentialState State [1][1]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	

5.3.11 EnableCredential command

This method is used to enable a credential.

A device shall support this command as described in Table 13.

Table 13 – EnableCredential command

EnableCredential		Access class: ACTUATE
Message name	Description	
EnableCredentialRequest	<i>This message contains:</i> • "Token": The token of the credential. • "Reason": Reason for enabling the credential. pt:ReferenceToken Token [1][1] pt:Name Reason [0][1]	
EnableCredentialResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	

5.3.12 DisableCredential command

This method is used to disable a credential.

A device shall support this command as described in Table 14.

Table 14 – DisableCredential command

DisableCredential		Access class: ACTUATE
Message name	Description	
DisableCredentialRequest	<i>This message contains:</i> • "Token": The token of the credential. • "Reason": Reason for disabling the credential. pt:ReferenceToken Token [1][1] pt:Name Reason [0][1]	
DisableCredentialResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	

5.3.13 ResetAntipassbackViolation command

This method is used to reset anti-passback violations for a specified credential.

A device that signals support for the ResetAntipassbackSupported capability shall support this method as described in Table 15.

Table 15 – ResetAntipassbackViolation command

ResetAntipassbackViolation		Access class: ACTUATE
Message name	Description	
ResetAntipassbackViolationRequest	<i>This message contains:</i> "CredentialToken": Token of Credential. pt:ReferenceToken CredentialToken [1][1]	
ResetAntipassbackViolationResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	

5.3.14 GetSupportedFormatTypes command

This method returns all the supported format types of a specified identifier type that is supported by the device.

A device shall support this command as described in Table 16.

Table 16 – GetSupportedFormatTypes command

GetSupportedFormatTypes		Access class: READ_SYSTEM
Message name	Description	
GetSupportedFormatTypesRequest	<i>This message contains:</i> "CredentialIdentifierTypeName": Name of the credential identifier type. xs:string CredentialIdentifierTypeName	
GetSupportedFormatTypesResponse	<i>This message contains:</i> "FormatTypeInfo": Identifier format types. tcr:CredentialIdentifierFormatTypeInfo FormatTypeInfo [1][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Identifier type is not found.</i>	

5.3.15 GetCredentialIdentifiers command

This method returns all the credential identifiers for a credential.

A device shall support this command as described in Table 17.

Table 17 – GetCredentialIdentifiers command

GetCredentialIdentifiers		Access class: READ_SYSTEM_SECRET
Message name	Description	
GetCredentialIdentifiersRequest	<i>This message contains:</i> "CredentialToken": Token of Credential. pt:ReferenceToken CredentialToken [1][1]	
GetCredentialIdentifiersResponse	<i>This message contains:</i> "CredentialIdentifier": Identifiers of the credential tcr:CredentialIdentifier CredentialIdentifier [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	Credential token is not found.	

5.3.16 SetCredentialIdentifier command

This operation creates or updates a credential identifier for a credential.

If the type of specified credential identifier already exists, the current credential identifier of that type is replaced. Otherwise the credential identifier is added.

A device shall support this command as described in Table 18.

Table 18 – SetCredentialIdentifier command

SetCredentialIdentifier		Access class: WRITE_SYSTEM
Message name	Description	
SetCredentialIdentifierRequest	<i>This message contains:</i> "CredentialToken": Token of Credential. "CredentialIdentifier": Identifier of the credential. pt:ReferenceToken CredentialToken [1][1] tcr:CredentialIdentifier CredentialIdentifier [1][1]	
SetCredentialIdentifierResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	Credential token is not found.	
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	Specified identifier type is not supported by the device, see the SupportedIdentifierType capability.	
env:Sender ter:InvalidArgVal ter:InvalidFormatType	Specified identifier format type is not supported by the device.	
env:Sender ter:InvalidArgVal ter:InvalidIdentifierValue	Credential identifier value is not in accordance with the FormatType definition.	

5.3.17 DeleteCredentialIdentifier command

This method deletes all the identifier values for the specified type. However, if the identifier type name does not exist in the device, it will be silently ignored without any response.

A device shall support this command as described in Table 19.

Table 19 – DeleteCredentialIdentifier command

DeleteCredentialIdentifier		Access class: WRITE_SYSTEM
Message name	Description	
DeleteCredentialIdentifierRequest	<i>This message contains:</i> "CredentialToken": Token of Credential. "CredentialIdentifierTypeName": Identifier type name of a credential. pt:ReferenceToken CredentialToken [1][1] pt:Name CredentialIdentifierTypeName [1][1]	
DeleteCredentialIdentifierResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	
env:Receiver ter:ConstraintViolated ter:MinIdentifiersPerCredential	<i>At least one credential identifier is required.</i>	

5.3.18 GetCredentialAccessProfiles command

This method returns all the credential access profiles for a credential.

A device shall support this command as described in Table 20.

Table 20 – GetCredentialAccessProfiles command

GetCredentialAccessProfiles		Access class: READ_SYSTEM
Message name	Description	
GetCredentialAccessProfilesRequest	<i>This message contains:</i> "CredentialToken": Token of Credential. pt:ReferenceToken CredentialToken [1][1]	
GetCredentialAccessProfilesResponse	<i>This message contains:</i> "CredentialAccessProfile": Access profiles of the credential. tcr:CredentialAccessProfile CredentialAccessProfile [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	

5.3.19 SetCredentialAccessProfiles command

This operation adds or updates the credential access profiles for a credential.

The device shall update the credential access profile if the access profile token in the specified credential access profile matches. Otherwise the credential access profile is added.

A device shall support this command as described in Table 21.

Table 21 – SetCredentialAccessProfiles command

SetCredentialAccessProfiles	Access class: ACTUATE
Message name	Description
SetCredentialAccessProfilesRequest	<i>This message contains:</i> • "CredentialToken": Token of Credential. • "CredentialAccessProfile": Access profiles of the credential. pt:ReferenceToken CredentialToken [1][1] tcr:CredentialAccessProfile CredentialAccessProfile [1][unbounded]
SetCredentialAccessProfilesResponse	<i>This message shall be empty.</i>
Fault codes	Description
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>
env:Sender ter:CapabilityViolated ter:MaxAccessProfilesPerCredential	<i>There are too many access profiles per credential, see the MaxAccessProfilesPerCredential capability.</i>
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	<i>Credential access profile validity is not supported by the device, see the CredentialAccessProfileValiditySupported capability.</i>
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	<i>A referred entity token is not found (some devices may not validate referred entities).</i>

5.3.20 DeleteCredentialAccessProfiles command

This method deletes credential access profiles for the specified credential token.

However, if no matching credential access profiles are found, the corresponding access profile tokens are silently ignored without any response.

A device shall support this command as described in Table 22.

Table 22 – DeleteCredentialAccessProfiles command

DeleteCredentialAccessProfiles		Access class: ACTUATE
Message name	Description	
DeleteCredentialAccessProfilesRequest	<i>This message contains:</i> • <i>"CredentialToken": Token of Credential.</i> • <i>"AccessProfileToken": Tokens of Access Profiles.</i> pt:ReferenceToken CredentialToken [1][1] pt:ReferenceToken AccessProfileToken [1][unbounded]	
DeleteCredentialAccessProfilesResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:NotFound	<i>Credential token is not found.</i>	

5.4 Notification topics

5.4.1 General

Subclause 5.4 defines notification topics specific to the credential service.

5.4.2 Event overview (informative)

The credential service specifies events when the credential state changes and when credentials are changed.

The main topics for status changes are:

- tns1:Credential/State/Enabled
- tns1:Credential/State/ApbViolation

The main topics for configuration change notifications are:

- tns1:Configuration/Credential/Changed
- tns1:Configuration/Credential/Removed

5.4.3 Status changes

5.4.3.1 General

The device shall provide the status change events to inform subscribed clients when credential entity status is changed. The device shall use the topics defined in 5.4.3 associated with the respective message description.

5.4.3.2 Credential

Whenever the credential state (enabled or disabled) is changed, the device shall provide the following event (the Reason field is one of the fields listed in 5.3.2.7, e.g. "pt:CredentialLost"):

```
Topic: tns1:Credential/State/Enabled
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                             Type="pt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="State"
                             Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="Reason"
                             Type="xs:string"/>
    <tt:SimpleItemDescription Name="ClientUpdated"
                             Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

ClientUpdated is set to true if the state change was initiated by the client. ClientUpdated is set to false if the device initiated the state change (e.g. because the wrong PIN was entered three times in a row).

The device shall provide the following event whenever there is any anti-passback violation:

```
Topic: tns1:Credential/State/ApbViolation
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                             Type="pt:ReferenceToken"/>
  </tt:Source>
  <tt:Data>
    <tt:SimpleItemDescription Name="ApbViolation"
                             Type="xs:boolean"/>
    <tt:SimpleItemDescription Name="ClientUpdated"
                             Type="xs:boolean"/>
  </tt:Data>
</tt:MessageDescription>
```

5.4.4 Configuration changes

5.4.4.1 General

Whenever configuration data has been changed, added or removed, the device shall provide these events to inform subscribed clients.

5.4.4.2 Credential

Whenever configuration data for a credential (including credential identifiers and credential access profiles) is changed, or if a credential is added, the device shall provide the following event:

```
Topic: tns1:Configuration/Credential/Changed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
                             Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

Whenever a credential is removed, the device shall provide the following event:

```
Topic: tns1:Configuration/Credential/Removed
<tt:MessageDescription IsProperty="false">
  <tt:Source>
    <tt:SimpleItemDescription Name="CredentialToken"
      Type="pt:ReferenceToken"/>
  </tt:Source>
</tt:MessageDescription>
```

6 Access rules service

6.1 General

The access rules service defines the access profile and its access policies. The credentials are associated with an access profile for access authorization to a facility.

The access rules service defines WHEN and WHERE credentials have access. Each credential is associated with access profiles, where each access profile defines the access for a group of people. For example, employees will have access to office doors during office hours. Another example is access to an apartment by one family during all hours. Each access profile consists of a number of access policies, where each access policy defines when access is possible to an access point.

The service is flexible in such a way that it is possible to give access to something other than an access point (by setting EntityType to a QName other than AccessPointInfo).

Figure 3 shows the main data structures involved in the access rules service.

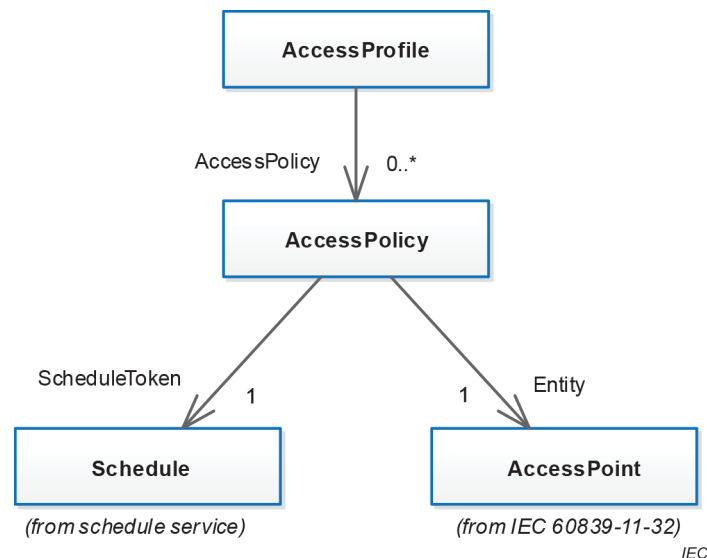


Figure 3 – Main data structures in the access rules service

6.2 Service capabilities

6.2.1 General

An ONVIF compliant device shall provide service capabilities in two ways:

- 1) With the `GetServices` method of `Device` service when `IncludeCapability` is `true`. Refer to the ONVIF Core Specification for more details.
- 2) With the `GetServiceCapabilities` method.

6.2.2 ServiceCapabilities data structure

The service capabilities reflect the optional functionalities of a service. The information is static and does not change during the device operation. The following capabilities are available:

- **MaxLimit**
The maximum number of entries returned by a single `Get<Entity>List` or `Get<Entity>` request. The device shall never return more than this number of entities in a single response.
- **MaxAccessProfiles**
Indicates the maximum number of access profiles supported by the device.
- **MaxAccessPoliciesPerAccessProfile**
Indicates the maximum number of access policies per access profile supported by the device.
- **MultipleSchedulesPerAccessPointSupported**
Indicates whether or not several access policies can refer to the same access point in an access profile.
- **ClientSuppliedTokenSupported**
Indicates that the client is allowed to supply the token when creating access profiles. To enable the use of the command `SetAccessProfile`, the value shall be set to `true`.

6.2.3 GetServiceCapabilities command

This operation returns the capabilities of the access rules service.

A device shall support this command as described in Table 23.

Table 23 – GetServiceCapabilities command

GetServiceCapabilities		Access class: PRE_AUTH
Message name	Description	
<code>GetServiceCapabilitiesRequest</code>	<i>This message shall be empty.</i>	
<code>GetServiceCapabilitiesResponse</code>	<i>This message contains:</i> • <i>"Capabilities": The capability response message contains the requested access rules service capabilities using a hierarchical XML capability structure.</i> tar:ServiceCapabilities Capabilities [1][1]	
Fault codes	Description	
	<i>No command specific faults.</i>	

6.3 Access profile information

6.3.1 General

Access profiles define who can access what and when.

6.3.2 Data structures

6.3.2.1 AccessProfileInfo

The AccessProfileInfo structure contains basic information about an access profile. The device shall provide the following fields for each access profile instance.

- **token**
A service unique identifier of the access profile.
- **Name**
A user readable name. It shall be up to 64 characters.

To provide more information, the device may include the following optional field:

- **Description**
User readable description for the access profile. It shall be up to 1 024 characters.

6.3.2.2 AccessProfile

The AccessProfile structure contains information about the collection of access policies. The device shall include all properties of the AccessProfileInfo structure and also a list of access policies.

- **AccessPolicy**
A list of access policy structures, where each access policy defines during which schedule an access point can be accessed.

6.3.2.3 AccessPolicy

The access policy is an association of an access point and a schedule. It defines when an access point can be accessed using an access profile which contains this access policy. If an access profile contains several access policies specifying different schedules for the same access point, this will result in a union of the schedules.

The device shall provide the following fields for each access policy instance.

- **ScheduleToken**
Reference to the schedule used by the access policy.
- **Entity**
Reference to the entity used by the rule engine. The entity type may be specified by the optional EntityType field explained below but is typically an access point.

To provide more information, the device may include the following optional field:

- **EntityType**
Optional entity type; if missing, an access point type as defined by the ONVIF Access Control service should be assumed. This can also be represented by the QName value "tac:AccessPoint" where tac is the namespace of [Access Control Service Specification]. This field is provided for future extensions; it will allow an access policy being extended to cover entity types other than access points as well.

6.3.3 GetAccessProfileInfo command

This operation requests a list of AccessProfileInfo items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 24.

Table 24 – GetAccessProfileInfo command

GetAccessProfileInfo		Access class: READ_SYSTEM
Message name	Description	
GetAccessProfileInfoRequest	<i>This message contains:</i> "Token": Tokens of AccessProfileInfo items to get. pt:ReferenceToken Token [1][unbounded]	
GetAccessProfileInfoResponse	<i>This message contains:</i> "AccessProfileInfo": List of AccessProfileInfo items. tar:AccessProfileInfo AccessProfileInfo [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgs ter:TooManyItems	Too many items were requested, see MaxLimit capability.	

6.3.4 GetAccessProfileInfoList command

This operation requests a list of all AccessProfileInfo items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 25.

Table 25 – GetAccessProfileInfoList command

GetAccessProfileInfoList		Access class: READ_SYSTEM
Message name	Description	
GetAccessProfileInfoListRequest	<i>This message contains:</i> • <i>"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.</i> • <i>"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.</i> xs:int Limit [0][1] xs:string StartReference [0][1]	
GetAccessProfileInfoListResponse	<i>This message contains:</i> • <i>"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.</i> • <i>"AccessProfileInfo": List of AccessProfileInfo items.</i> xs:string NextStartReference [0][1] tar:AccessProfileInfo AccessProfileInfo [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:InvalidStartReference	<i>StartReference is invalid or has timed out. Client needs to start fetching from the beginning.</i>	

6.3.5 GetAccessProfiles command

This operation requests a list of AccessProfile items matching the given tokens.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens. The device shall not return a fault in this case.

If the number of requested items is greater than MaxLimit, a TooManyItems fault shall be returned.

A device shall support this command as described in Table 26.

Table 26 – GetAccessProfiles command

GetAccessProfiles		Access class: READ_SYSTEM
Message name	Description	
GetAccessProfileRequest	<i>This message contains:</i> • "Token": Tokens of Access Profile items to get. pt:ReferenceToken Token [1][unbounded]	
GetAccessProfileResponse	<i>This message contains:</i> • "AccessProfile": List of Access Profile items. tar:AccessProfile AccessProfile [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgs ter:TooManyItems	Too many items were requested, see MaxLimit capability.	

6.3.6 GetAccessProfileList command

This operation requests a list of all AccessProfile items provided by the device.

A call to this method shall return a StartReference when not all data is returned and more data is available. The reference shall be valid for retrieving the next set of data. Refer to IEC 60839-11-32:2016, 4.7.3 for more details.

The number of items returned shall not be greater than the Limit parameter.

A device shall support this command as described in Table 27.

Table 27 – GetAccessProfileList command

GetAccessProfileList		Access class: READ_SYSTEM
Message name	Description	
GetAccessProfileListRequest	<i>This message contains:</i> • <i>"Limit": Maximum number of entries to return. If not specified, less than one or higher than what the device supports, the number of items is determined by the device.</i> • <i>"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.</i> xs:int Limit [0][1] xs:string StartReference [0][1]	
GetAccessProfileListResponse	<i>This message contains:</i> • <i>"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.</i> • <i>"AccessProfile": List of Access Profile items.</i> xs:string NextStartReference [0][1] tar:AccessProfile AccessProfile [0][unbounded]	
Fault codes	Description	
env:Sender ter:InvalidArgVal ter:InvalidStartReference	<i>StartReference is invalid or has timed out. Client needs to start fetching from the beginning.</i>	

6.3.7 CreateAccessProfile command

This operation creates the specified access profile in the device.

The token field of the AccessProfile structure shall be empty and the device shall allocate a token for the access profile. The allocated token shall be returned in the response.

If the client sends any value in the token field, the device shall return InvalidArgVal as a generic fault code.

If several access policies in one access profile are specifying different schedules for the same access point, then it will result in a union of the schedules.

Figure 4 shows an example of several schedules used at one access point. Each row in the figure below corresponds to one access policy. The first and second lines are two access policies using different schedules for the same access point.

Time periods of the individual schedule					Access policies	
08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	Schedule	Access point
					Schedule 1	Room A Entry
					Schedule 2	Room A Entry
					Schedule 1	Room B Entry

Figure 4 – Multiple schedules per access point

Since both Schedule 1 and Schedule 2 define when Room A Entry can be accessed, a credential holder will experience a union of Schedule 1 and Schedule 2, as seen in the first row in Figure 5 below.

Time periods of the schedule union					Access policies	
08:00-09:00	09:00-10:00	10:00-11:00	11:00-12:00	12:00-13:00	Schedule	Access point
					Schedule 1 and Schedule 2 union	Room A Entry
					Schedule 1	Room B Entry

Figure 5 – Result of schedule union

A device shall support this command as described in Table 28.

Table 28 – CreateAccessProfile command

CreateAccessProfile		Access class: WRITE_SYSTEM
Message name	Description	
CreateAccessProfileRequest	<i>This message contains:</i> "AccessProfile": The AccessProfile to create. tar:AccessProfile AccessProfile [1][1]	
CreateAccessProfileResponse	<i>This message contains:</i> "Token": The Token of created AccessProfile. pt:ReferenceToken Token [1][1]	
Fault codes	Description	
env:Receiver ter:CapabilityViolated ter:MaxAccessProfiles	There is not enough space to add a new AccessProfile, see the MaxAccessProfiles capability.	
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	There are too many AccessPolicies in an AccessProfile, see MaxAccessPoliciesPerAccessProfile capability.	
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	Multiple AccessPoints are not supported for the same schedule, see MultipleSchedulesPerAccessPointSupported capability.	
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	A referred entity token is not found (some devices may not validate referred entities).	

6.3.8 SetAccessProfile command

This method is used to synchronize an access profile in a client with the device.

If an access profile with the specified token does not exist in the device, the access profile is created. If an access profile with the specified token exists, then the access profile is modified.

A call to this method takes an access profile structure as input parameter. The token field of the access profile shall not be empty.

A device that signals support for the ClientSuppliedTokenSupported capability shall implement this command.

If no token was specified in the request, the device shall return `InvalidArgs` as a generic fault code.

A device shall support this command as described in Table 29.

Table 29 – SetAccessProfile command

SetAccessProfile		Access class: WRITE_SYSTEM
Message name	Description	
SetAccessProfileRequest	<i>This message contains:</i> "AccessProfile": The AccessProfile item to create or modify. tar:AccessProfile AccessProfile [1][1]	
SetAccessProfileResponse	<i>This message shall be empty.</i>	
Fault codes	Description	
env:Receiver ter:CapabilityViolated ter:ClientSuppliedTokenSupported	<i>The device does not support that the client supplies the token.</i>	
env:Receiver ter:CapabilityViolated ter:MaxAccessProfiles	<i>There is not enough space to add a new AccessProfile, see the MaxAccessProfiles capability.</i>	
env:Sender ter:CapabilityViolated ter:MaxAccessPoliciesPerAccessProfile	<i>There are too many AccessPolicies in an AccessProfile, see MaxAccessPoliciesPerAccessProfile capability.</i>	
env:Sender ter:CapabilityViolated ter:MultipleSchedulesPerAccessPointSupported	<i>Multiple AccessPoints are not supported for the same schedule, see MultipleSchedulesPerAccessPoint-Supported capability.</i>	
env:Sender ter:InvalidArgVal ter:ReferenceNotFound	<i>A referred entity token is not found (some devices may not validate referred entities).</i>	

6.3.9 ModifyAccessProfile command

This operation modifies the specified access profile.

The token of the access profile to modify is specified in the token field of the AccessProfile structure and shall not be empty. All other fields in the structure shall overwrite the fields in the specified access profile.

If several access policies specify different schedules for the same access point, this will result in a union of the schedules. See Figure 4 and Figure 5 above.

If no token was specified in the request, the device shall return `InvalidArgs` as a generic fault code.

A device shall support this command as described in Table 30.