ONVIF™
Credential Service Specification

Version 17.06

June 2017

Contents

Contributors

Version 1

ASSA ABLOY	Patrik Björling Rygert
ASSA ABLOY	Mattias Rengstedt
Axis Communications AB	Marcus Johansson
Axis Communications AB	Robert Rosengren
Bosch	Mohane Caliaperoumal
Honeywell	Uvaraj Thangarajan
Honeywell	Neelendra Bhandari
Honeywell	Mayur Salgar
PACOM	Eugene Scully
Siemens Limited	Lokeshwar K
Siemens Limited	Suresh Raman

1 Scope

This specification defines the web service interface for integration with physical access control systems. This includes discovering components and support of the configuration of the credentials components.

Supplementary dedicated services such as access control and access rules services will be defined in separate document.

Web service usage and common ONVIF functionality are outside the scope of this document. Please refer to the ONVIF Core Specification for more information.

1.1 Normative references

- **ONVIF Network Interface Specification Set**

  ONVIF Network Interface Specification Set version 2.6 or later.
  http://www.onvif.org/Documents/Specifications.aspx

- **Core Specification**

  Part of [ONVIF Network Interface Specification Set]

- **Access Control Service Specification**

  ONVIF Access Control Service Specification version 1.0
  Part of [ONVIF Network Interface Specification Set]

- **Access Rules Service Specification**

  ONVIF Access Rules Service Specification version 1.0
  Part of [ONVIF Network Interface Specification Set]

- **ISO IEC Directives**

  ISO/IEC Directives, Part 2 (6th edition)
  Rules for the structure and drafting of international standards.
  http://www.iec.ch/members_experts/refdocs/iec/isoiec-dir2%7Bed6.0%7Den.pdf

- **ISO 16484-5:2014-09 Annex P**

  DIN EN ISO 16484-5:2014-09 Annex P
  BACnet Encoding of Standard Authentication Factor Formats (Normative)
  https://www.iso.org/obp/ui/#!iso:std:63753:en

2 Terms and Definitions

2.1 Conventions

The key words “shall”, “shall not”, “should”, “should not”, “may”, “need not”, “can”, “cannot” in this specification are to be interpreted as described in Annex H of [ISO IEC Directives].

2.2 Definitions

Anti-Passback Operating mode which requires user validation when leaving a security controlled area in order to be able to re-enter and vice versa.

Anti-Passback Violation State A signal stating if the anti-passback rules have been violated for a credential.

Access Profile A collection of access policies. Is used to define role based access.

Credential 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.

Credential Format The credential data can be formatted in many different ways. ONVIF supports the BACnet format types in [ISO 16484-5:2014-09 Annex P].

Credential Holder Associates a credential with a user. Typically it holds a reference to a credential and a reference to a user.

Credential Identifier Card number, unique card information, PIN, fingerprint, or other biometric information, etc., that can be validated in an access point.

Credential Number A sequence of bytes uniquely identifying a credential at an access point.

Credential State The credential state indicates if a credential is enabled or disabled. The state also indicates if anti-passback has been violated or not. The state may also contain a reason why the credential was disabled.

Duress Forcing a person to provide access to a secure area against that person's wishes.

Format Type See Credential Format.

Validity Period From a certain point in time, to a later point in time.

2.3 Abbreviations

PACS Physical Access Control System

3 Overview

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, etc. 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 [Access Rules Service Specification].

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-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 section 4.1.1.1 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 1

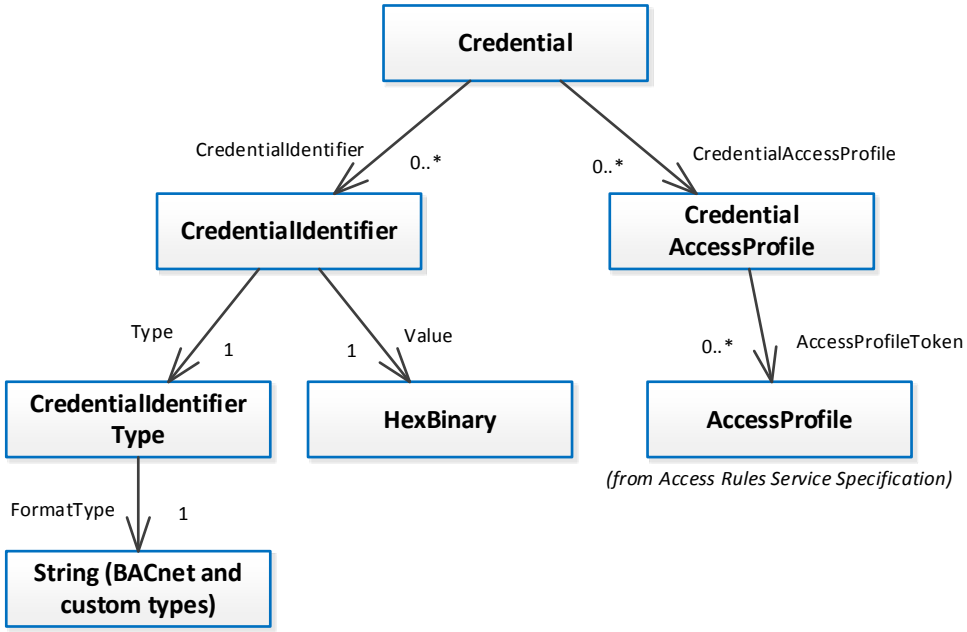

Figure 1: Main data structures in the credential service

4 Credentials

This service offers commands for configuring the credentials.

4.1 Service Capabilities

The device shall provide service capabilities in two ways:

1. With the GetServices method of Device service when IncludeCapability is true. Please refer to [Core Specification] for more details.
2. With the GetServiceCapabilities method.

4.1.1 Data Structures

4.1.1.1 Service Capabilities

The service capabilities reflect optional functionality of a service. The information is static and does not change during 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 credential 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. Supported identifiers starting with the prefix pt: are reserved to define PACS specific identifier types and these reserved identifier types shall all share the "pt:<Name>" syntax.

  - pt:Card Supports Card identifier type
  - 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

- **SupportedExemptionType**

  A list of exemptions that the device supports. Supported exemptions starting with the prefix pt: are reserved to define PACS specific exemption types and these reserved exemption types shall all share "pt:<Name>" syntax.

  - pt:ExemptFromAuthentication Supports ExemptedFromAuthentication in section 4.2.1.3.

4.1.2 GetServiceCapabilities command

This operation returns the capabilities of the credential service.

Table 1 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]**

4.2 Credential Information

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

4.2.1 Data Structures

4.2.1.1 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 1024 characters.

- **ValidFrom**

  The start date/time validity of the credential. If the ValiditySupportsTimeValue capability is set to false, then only date is supported (time is ignored).

- **ValidTo**

  The expiration date/time validity of the credential. If the ValiditySupportsTimeValue capability is set to false, then only date is supported (time is ignored).

4.2.1.2 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.

- **Attributes**

  A list of credential attributes as name value pairs. Key names starting with the prefix pt: are reserved to define PACS specific attributes following the "pt:<Name>" syntax.

4.2.1.3 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.

4.2.1.4 CredentialIdentifierType

Specifies the name of 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 section 4.1.1.1 for supported credential identifier types).

- **FormatType**

  Specifies the format of the credential value for the specified identifier type name. See section 4.2.1.5 below.

4.2.1.5 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 16484-5:2014-09 Annex P]. The BACnet type “CUSTOM” is not used in this specification. Instead device manufacturers can define their own format types.

- **Description**

  User readable description of the credential identifier format type. It shall be up to 1024 characters. For custom types, it is recommended to describe how the octet string is encoded (following the structure in column *Authentication Factor Value Encoding* of [ISO 16484-5:2014-09 Annex P]).

4.2.1.6 CredentialAccessProfile

The association between a credential and an access profile.

- **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 date is supported (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 date is supported (time is ignored).

4.2.1.7 CredentialState

The CredentialState structure contains information about the state of the credential and optionally the reason of 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**

  Predefined ONVIF reasons as mentioned in the section 4.2.1.8 DisabledReasons. For any other reason, free text can be used.

- **AntipassbackState**

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

4.2.1.8 DisabledReasons

The enumerated predefined disabled reasons for the credential.

- **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).

4.2.1.9 Anti-passback State

A structure containing anti-passback related state information.

- **AntipassbackViolated**

  Indicates if anti-passback is violated for the credential.

4.2.2 GetCredentialInfo command

This method returns a list of credential info items matching the given tokens. Only found credentials shall be returned i.e., the returned number of elements can differ from the requested number of elements.

The device shall ignore tokens it cannot resolve and may return an empty list if there are no credentials matching the specified token. If the number of requested items are greater than MaxLimit, a TooManyItems fault shall be returned.

Table 2 GetCredentialInfo command

GetCredentialInfo	Access Class: READ_SYSTEM
Message name	**Description**
GetCredentialInfoRequest	*This message contains:* • *"Token": Tokens of CredentialInfo items to get.* pt:ReferenceToken **Token [1][unbounded]**
GetCredentialInfoResponse	*This message contains:* • *"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.*

4.2.3 GetCredentialInfoList command

This operation requests a list of all credential info 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. Please refer section 4.8.3 in [Access Control Service Specification] for more details.

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

Table 3 GetCredentialInfoList command

GetCredentialInfoList	Access Class: READ_SYSTEM
Message name	**Description**
GetCredentialInfoListRequest	*This message contains:* • *"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.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetCredentialInfoListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"CredentialInfo": List of CredentialInfo items.* xs:string **NextStartReference [0][1]** tcr:CredentialInfo**CredentialInfo [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

4.2.4 GetCredentials command

This operation returns the specified 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 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.

Table 4 GetCredentials command

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

4.2.5 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. Please refer section 4.8.3 in [Access Control Service Specification] for more details. The number of items returned shall not be greater the Limit parameter.

Table 5 GetCredentialList command

GetCredentialList	Access Class: READ_SYSTEM_SECRET
Message name	**Description**
GetCredentialListRequest	*This message contains:* • *"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.* • *"StartReference": Start returning entries from this start reference. If not specified, entries shall start from the beginning of the dataset.* xs:int **Limit [0][1]** xs:string **StartReference [0][1]**
GetCredentialListResponse	*This message contains:* • *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* • *"Credential": List of Credential items.* xs:string **NextStartReference [0][1]** tcr:Credential **Credential [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:InvalidStartReference	*StartReference is invalid or has timed out. Client needs to start fetching from the beginning.*

4.2.6 CreateCredential command

This operation creates a credential. A call to this method takes a credential structure and a credential state structure as input parameters. The credential state can be created in disabled or enabled state. The token field of the credential shall be empty, 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 generic fault code.

Table 6 CreateCredential command

CreateCredential	Access Class: WRITE_SYSTEM
Message name	**Description**
CreateCredentialRequest	*This message contains:* • *"Credential": The credential to create.* • *"State": The state of the credential.* tcr:Credential **Credential [1][1]** tcr:CredentialState **State [1][1]**
CreateCredentialResponse	*This message contains:* • *"Token": The token of the created credential* pt:ReferenceToken **Token [1][1]**
Fault codes	**Description**
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 device, see the CredentialValiditySupported capability*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*Credential access profile validity is not supported by the device, see the CredentialAccessProfileValidity-Supported capability.*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Specified identifier type is not supported by 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 as per 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:InvalidArgVal ter:ExemptFromAuthenticationSupported	*Exempt from authentication is not supported by the device. See the SupportedExemptionType capability.*

env:Receiver ter:CapabilityViolated ter:MaxCredentials	*There is not enough space to create a new credential, see the MaxCredentials capability*

4.2.7 ModifyCredential command

This operation modifies 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.

Table 7 ModifyCredential command

ModifyCredential	Access Class: WRITE_SYSTEM
Message name	**Description**
ModifyCredentialRequest	*This message contains:* • *"Credential": Details of the credential.* tcr:Credential **Credential [1][1]**
ModifyCredentialResponse	*This message shall be empty*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*
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 device, see the CredentialValiditySupported capability*
env:Sender ter:CapabilityViolated ter:CredentialAccessProfileValiditySupported	*Credential access profile validity is not supported by device, see the CredentialAccessProfileValidity-Supported capability*
env:Sender ter:CapabilityViolated ter:SupportedIdentifierType	*Specified identifier type not is supported by 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 as per 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:InvalidArgVal ter:ExemptFromAuthenticationSupported	*Exempt from authentication is not supported by the device. See the SupportedExemptionType capability.*

4.2.8 DeleteCredential command

This method deletes the specified credential.

If it is referred to by another entity some devices may not be able to delete the credential, and consequently a ReferenceInUse fault shall be generated.

Table 8 DeleteCredential command

DeleteCredential	Access Class: WRITE_SYSTEM
Message name	**Description**
DeleteCredentialRequest	*This message contains:* • *"Token": The token of the credential to delete.* pt:ReferenceToken **Token [1][1]**
DeleteCredentialResponse	*This message shall be empty*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*
env:Sender ter:InvalidArgVal ter:ReferenceInUse	*Failed to delete, credential token is in use*

4.2.9 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 credential state structure.

Table 9 GetCredentialState command

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

4.2.10 EnableCredential command

This method is used to enable a credential.

Table 10 EnableCredential command

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

4.2.11 DisableCredential command

This method is used to disable a credential.

Table 11 DisableCredential command

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

4.2.12 ResetAntipassbackViolation command

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

Table 12 ResetAntipassbackViolation command

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

4.2.13 GetSupportedFormatTypes command

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

Table 13 GetSupportedFormatTypes command

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

4.2.14 GetCredentialIdentifiers command

This method returns all the credential identifiers for a credential.

Table 14 GetCredentialIdentifiers command

GetCredentialIdentifiers	Access Class: READ_SYSTEM_SECRET
Message name	**Description**
GetCredentialIdentifersRequest	*This message contains:* • *"CredentialToken": Token of Credential* pt:ReferenceToken **CredentialToken [1][1]**
GetCredentialIdentifersResponse	*This message contains:* • *"CredentialIdentifier": Identifiers of the credential* tcr:CredentialIdentifier **CredentialIdentifier [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

4.2.15 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.

Table 15 SetCredentialIdentifier command

SetCredentialIdentifier	Access Class: WRITE_SYSTEM
Message name	**Description**
SetCredentialIdentiferRequest	*This message contains:* • *"CredentialToken": Token of Credential* • *“CredentialIdentifier”: Identifier of the credential* pt:ReferenceToken **CredentialToken [1][1]** tcr:CredentialIdentifier **CredentialIdentifier [1][1]**
SetCredentialIdentiferResponse	*This message shall be empty*
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 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 as per the FormatType definition.*
env:Sender ter:InvalidArgVal ter:ExemptFromAuthenticationSupported	*Exempt from authentication is not supported by the device. See the SupportedExemptionType capability.*

4.2.16 DeleteCredentialIdentifier command

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

Table 16 DeleteCredentialIdentifier command

DeleteCredentialIdentifier	Access Class: WRITE_SYSTEM
Message name	**Description**
DeleteCredentialIdentiferRequest	*This message contains:* • *"CredentialToken": Token of Credential* • *"CredentialIdentifierTypeName": Identifier type name of acredential* pt:ReferenceToken **CredentialToken [1][1]** pt:Name **CredentialIdentifierTypeName [1][1]**
DeleteCredentialIdentiferResponse	*This message shall be empty*
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*
env:Receiver ter:ConstraintViolated ter:MinIdentifiersPerCredential	*At least one credential identifier is required*

4.2.17 GetCredentialAccessProfiles command

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

Table 17 GetCredentialAccessProfiles command

GetCredentialAccessProfiles	Access Class: READ_SYSTEM
Message name	**Description**
GetCredentialAccessProfilesRequest	*This message contains:* • *"CredentialToken": Token of Credential* pt:ReferenceToken **CredentialToken [1][1]**
GetCredentialAccessProfilesResponse	*This message contains:* • *"CredentialAccessProfile": Access Profiles of the credential* tcr:CredentialIlAccessProfile **CredentialAccessProfile [0][unbounded]**
Fault codes	**Description**
env:Sender ter:InvalidArgVal ter:NotFound	*Credential token is not found.*

4.2.18 SetCredentialAccessProfiles command

This operation add 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.

Table 18 SetCredentialAccessProfiles command

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

4.2.19 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.

Table 19 DeleteCredentialAccessProfiles command

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

5 Notification topics

This section defines notification topics specific to the credential service.

5.1 Event overview (informative)

The credential service specifies events when 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

5.2 Status changes

5.2.1 Credential

Whenever the credential state (enabled or disabled) is changed, the device shall provide the following event:

```
Topic: tns1:Credential/State/Enabled

<tt:MessageDescriptionIsProperty=”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:MessageDescriptionIsProperty=”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>
```

5.3 Configuration changes

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

5.3.1 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:MessageDescriptionIsProperty=”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:MessageDescriptionIsProperty=”false”>
  <tt:Source>
    <tt:SimpleItemDescription Name=”CredentialToken”
                              Type=”pt:ReferenceToken”/>
  </tt:Source>
</tt:MessageDescription>
```

Annex B. Revision History

Vers.	Date	Author	Description
1.0	Jun 2015	PACS WG	First release
17.06	Jun 2017	Hiroyuki Sano	Change request 2068, 2069, 2071, 2079, 2080, 2081, 2098, 2112, 2111, 2113