ONVIF®

Profile D Specification

RELEASE CANDIDATE

Version RC 1.1

March 2021

REVISION HISTORY

Vers.	Date	Description	Contributors
RC 1.0	June, 2020	Release candidate version 1.0	Refer to Contributors table
RC 1.1	March, 2021	Removed deprecated GetWsdlUrl, Added AccessControl/Denied|AccessGranted/* events implied by ExternalAuthorization Removed reference to Analytics service. Clarified that at least one access point must support ExternalAuthorization. Add Configuration events in Door and Access Point management sections.	Refer to Contributors table

CONTRIBUTORS

Company	Contributors
ASSA ABLOY AB	Patrik Björling Rygert – Working Group chairman, editor Mattias Rengstedt
Axis Communications AB	Emil Selinder Derek Wang Johan Adolfsson
Dahua Technology	Weiming Mao Hui (Hugh) Xu
Hikvision Digital Technology	Shuanglong Liao Yuanyuan (Peggy) Zheng
Honeywell	Vinay Ghule Giri Guntipalli

TABLE OF CONTENTS

1 Scope

This document defines the mandatory and conditional features required by an ONVIF Device and ONVIF Client that support Profile D.

2 Normative references

This section defines the normative references applicable to this specification.

2.1 Normative References

- **ONVIF Profile Policy**

  < http://www.onvif.org/profiles>

- **ONVIF Network Interface Specification Set**

  < https://www.onvif.org/profiles/specifications/ >

3 Terms and Definitions

This section provides common terms and definitions used in this specification.

3.1 Definitions

Profile	See [ONVIF Profile Policy]
ONVIF Device	Networked hardware appliance or software program that exposes one or multiple ONVIF Web Services
ONVIF Client	Networked hardware appliance or software program that uses ONVIF Web Services.
tns1	A prefix for the ONVIF topic namespace "http://www.onvif.org/ver10/topics". This prefix is not part of the standard and an implementation can use any prefix. See [ONVIF Network Interface Specification Set] Core Specification description of Namespaces for details.

4 Technical Specification Version Requirement

Implementation of ONVIF Network Interface Specification Set, version 20.06 or later is required for conformance to Profile D.

5 Requirement Levels

Each feature in this document has a requirement level for Device and Client that claim conformance to Profile D and contains a Function List that states the functions requirement level for Device and Client that implement that feature.

The requirement levels for features are:

- **Mandatory = Feature that shall be implemented by an ONVIF device or ONVIF client.**
- **Conditional = Feature that shall be implemented by an ONVIF device or ONVIF client if it supports that functionality in any way, including any proprietary way. Features that are conditional are marked with "if supported" in a profile specification.**

The requirement levels for functions are:

- **Mandatory = Function that shall be implemented by an ONVIF device or ONVIF client.**
- **Conditional = Function that shall be implemented by an ONVIF device or ONVIF client if it supports that functionality.**
- **Optional = Function that may be implemented by an ONVIF device or ONVIF client.**

Function Lists use the following abbreviations:

- **M = Mandatory**
- **C = Conditional**
- **O = Optional**

All functions shall be implemented as described in the corresponding [ONVIF Network Interface Specification Set].

6 Overview

An ONVIF profile is described by a fixed set of functionalities through a number of services that are provided by the ONVIF standard. A number of services and functionalities are mandatory for each type of ONVIF profile. An ONVIF device and client may support any combination of profiles and other optional services and functionalities.

An ONVIF device conformant with Profile D is an ONVIF device that captures some input credential identifier (carried, memorized or biometric) and passes it on to a client that either grants or denies the access request. The client informs the device about the decision, which performs necessary actions (e.g. unlocking a door). A Profile D device is not an access control unit, and therefore does not need to store access rules, schedules or credentials locally. However, a Profile D device may have the capability to store white- or blacklisted credential identifiers.

An ONVIF client conformant with Profile D is an ONVIF client that configures the device with necessary data such as which door(s) and access point(s) the device is responsible for, and possibly also white- and/or blacklisted credential identifiers. The client is also responsible of receiving access request events from the device, take the access control decision, and the command the device to either grant or deny the access request. The client can also command the device to perform operations on the door(s) it controls, such as locking/unlocking, double locking, blocking, etc.

7 Profile Mandatory Features (normative)

Devices and Clients conformant to Profile D shall support the following features. The requirements represent the minimum functionality that must be implemented for conformance.

7.1 User authentication

This section describes the required method of user authentication.

7.1.1 Device requirements

- Device shall authenticate HTTP requests using Digest authentication as described by the **Core Specification**.
- Device shall authenticate RTSP requests using Digest authentication as described by the **Core Specification** if RTSP is supported by the device.
- Device shall authenticate RTSP requests tunneled over HTTP using Digest authentication on the RTSP level as described by the **Core Specification** if RTSP is supported by the device.

7.1.2 Client requirements

- Client shall support authenticating HTTP requests using Digest authentication as described by the **Core Specification**.
- Client shall support authenticating RTSP requests using Digest authentication as described by the **Core Specification** if RTSP is supported by the client.
- Client shall support authenticating RTSP requests tunneled over HTTP using Digest authentication on the RTSP level as described by the **Core Specification** if RTSP is supported by the client.

7.1.3 Function List for Devices

User Authentication		**Device MANDATORY**
Function	**Service**	**Requirement**
Digest authentication	Core	M

7.1.4 Function List for Clients

User Authentication		**Client MANDATORY**
Function	**Service**	**Requirement**
Digest authentication	Core	M

7.2 Capabilities

This section describes the operations related to obtaining the capabilities of a device.

7.2.1 Device requirements

- Device shall support **GetServices** and **GetServiceCapabilities** as detailed in the **Core Specification**.
- Device shall support **GetServiceCapabilities** as detailed in the **Access Control** and **Door Control Service Specifications**.
- If supported, device shall support **GetServiceCapabilities** as detailed in the **Credential Service Specifications**.
- Device shall provide the supported number of access points via the **MaxAccessPoints** capability as detailed in the **Access Control Service Specification**.
- Device shall provide the supported number of doors via the **MaxDoors** capability as detailed in the **Door Control Service Specification**.
- Device shall provide a number larger than zero for at least one of **MaxAccessPoints** and **MaxDoors**.
- Device shall indicate support for at least two pull point subscriptions by returning **MaxPullPoints** set to no less than two in the response to **GetServiceCapabilities** in the **Event Service Specification**.

7.2.2 Client requirements

- Client shall determine the available **Services** using the **GetServices** operation.

7.2.3 Function List for Devices

Capabilities		**Device MANDATORY**
Function	**Service**	**Requirement**
GetServices	Device Management	M
GetServiceCapabilities	Device Management	M
GetServiceCapabilities	Event	M
GetServiceCapabilities	Access Control	M
GetServiceCapabilities	Door Control	M
GetServiceCapabilities	Credential	C

7.2.4 Function List for Clients

Capabilities		Client MANDATORY
Function	**Service**	**Requirement**
GetServices	Device Management	M
GetServiceCapabilities	Device Management	O
GetServiceCapabilities	Event	O
GetServiceCapabilities	Access Control	O
GetServiceCapabilities	Door Control	O
GetServiceCapabilities	Credential	O

7.3 Discovery

This section describes the operations related to device discovery.

7.3.1 Device requirements

- Device shall support **WS-Discovery** as specified in the **Core Specification**.
- Device shall support discovery mode using the operations **GetDiscoveryMode** and **SetDiscoveryMode**.
- Device shall support listing, adding, modifying and removing discovery scopes using the operations **GetScopes**, **AddScopes**, **SetScopes** and **RemoveScopes**
- Device shall support the Profile D-specific scope parameter presented in 7.3.5 **Scope Parameters**.

7.3.2 Client requirements

- Client shall be able to discover a device using **WS-Discovery** as specified in the **Core Specification**.

7.3.3 Function List for Devices

Discovery		Device MANDATORY
Function	**Service**	**Requirement**
WS-Discovery	Core	M
GetDiscoveryMode	Device Management	M
SetDiscoveryMode	Device Management	M
GetScopes	Device Management	M
SetScopes	Device Management	M
AddScopes	Device Management	M
RemoveScopes	Device Management	M

7.3.4 Function List for Clients

Discovery		Client MANDATORY
Function	**Service**	**Requirement**
WS-Discovery	Core	M
GetDiscoveryMode	Device Management	O
SetDiscoveryMode	Device Management	O
GetScopes	Device Management	O
SetScopes	Device Management	O
AddScopes	Device Management	O
RemoveScopes	Device Management	O

7.3.5 Scope Parameters

Discovery		
Category	**Defined Values**	**Description**
Profile	D	The scope indicates if the device is conformant with Profile D. A device conformant with Profile D shall include a scope entry with this value in its scope list.

7.4 Network Configuration

This section describes the operations related to the configuration of network settings on the device.

7.4.1 Device requirements

- Device shall support listing and configuring the device hostname using the **GetHostName** and **SetHostName** operations.

- Device shall support listing and configuring the DNS values using the **GetDNS** and **SetDNS** operations.
- Device shall support listing and configuring supported network interfaces on the device using the **GetNetworkInterfaces** and **SetNetworkInterfaces** operations.
- Device shall support listing and configuring supported network protocols on the device using the **GetNetworkProtocols** and **SetNetworkProtocols** operations.
- Device shall support listing and configuring the default gateway of the device using the **GetNetworkDefaultGateway** and **SetNetworkDefaultGateway** operations.

7.4.2 Client requirements

- Client shall be able to list and configure supported network interfaces on the device using the **GetNetworkInterfaces** and **SetNetworkInterfaces** operations.
- Client shall be able to list and set the default gateway of the device using the **GetNetworkDefaultGateway** and **SetNetworkDefaultGateway** operations.

7.4.3 Function List for Devices

Network Configuration		**Device MANDATORY**
Function	**Service**	**Requirement**
GetHostName	Device Management	M
SetHostName	Device Management	M
GetDNS	Device Management	M
SetDNS	Device Management	M
GetNetworkInterfaces	Device Management	M
SetNetworkInterfaces	Device Management	M
GetNetworkProtocols	Device Management	M
SetNetworkProtocols	Device Management	M
GetNetworkDefaultGateway	Device Management	M
SetNetworkDefaultGateway	Device Management	M

7.4.4 Function List for Clients

Network Configuration		Client MANDATORY
Function	**Service**	**Requirement**
GetHostName	Device Management	O
SetHostName	Device Management	O
GetDNS	Device Management	O
SetDNS	Device Management	O
GetNetworkInterfaces	Device Management	M
SetNetworkInterfaces	Device Management	M
GetNetworkProtocols	Device Management	O
SetNetworkProtocols	Device Management	O
GetNetworkDefaultGateway	Device Management	M
SetNetworkDefaultGateway	Device Management	M

7.5 System

This section describes the operations related to obtaining device information and the configuration of device settings.

7.5.1 Device requirements

- Device shall support the listing of device information such as manufacturer, model and firmware version using the **GetDeviceInformation** operation.
- Device shall support listing and configuring the date and time on the device using the **GetSystemDateAndTime** and **SetSystemDateAndTime** operations.
- Device shall support the ability to return to factory settings using the **SetSystemFactoryDefault** operation.
- Device shall support rebooting using the **SystemReboot** operation.

7.5.2 Client requirements (if supported)

- Client shall be able to get device information such as manufacturer, model and firmware version using the **GetDeviceInformation** operation.

7.5.3 Function List for Devices

System		Device MANDATORY
Function	**Service**	**Requirement**
GetDeviceInformation	Device Management	M
GetSystemDateAndTime	Device Management	M
SetSystemDateAndTime	Device Management	M
SetSystemFactoryDefault	Device Management	M
SystemReboot	Device Management	M

7.5.4 Function List for Clients

System		Client CONDITIONAL
Function	**Service**	**Requirement**
GetDeviceInformation	Device Management	M
GetSystemDateAndTime	Device Management	O
SetSystemDateAndTime	Device Management	O
SetSystemFactoryDefault	Device Management	O
SystemReboot	Device Management	O

7.6 User Handling

This section describes the operations related to managing users on the device.

7.6.1 Device requirements

- Device shall support creating, listing, modifying and deleting users from the device using the **CreateUsers**, **GetUsers**, **SetUser** and **DeleteUsers** operations.

7.6.2 Client requirements (if supported)

- Client shall be able to create, list, modify and delete users from the device using the **CreateUsers**, **GetUsers**, **SetUser** and **DeleteUsers** operations.

7.6.3 Function List for Devices

User Handling	Device MANDATORY	
Function	**Service**	**Requirement**
GetUsers	Device Management	M
CreateUsers	Device Management	M
DeleteUsers	Device Management	M
SetUser	Device Management	M

7.6.4 Function List for Clients

User Handling	Client CONDITIONAL	
Function	**Service**	**Requirement**
GetUsers	Device Management	M
CreateUsers	Device Management	M
DeleteUsers	Device Management	M
SetUser	Device Management	M

7.7 Event Handling

This section describes the operations related to retrieving and filtering events. The Real-time Pull-Point Notification Interface as covered by the **Core Specification** is Mandatory for Profile D conformance.

7.7.1 Device requirements

- Device shall support event handling with a pull point using the **SetSynchronizationPoint**, **CreatePullPointSubscription** and **PullMessage** operations.
- Device shall support retrieval of supported filter dialects and topics using the **GetEventProperties** operation.
- Device shall support event filtering using **Message Content Filter** and **Topic Filter** as described in the **Core Specification**.
- Device shall return the following **MessageContentFilterDialect** in response to **GetEventProperties**:
  - http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter
- Device shall support subscription management using the **Unsubscribe** operation.
- Device shall support at least two concurrent pull point subscriptions.

7.7.2 Client requirements

- Client shall implement event handling with a pull point using the **SetSynchronizationPoint**, **CreatePullPointSubscription** and **PullMessage** operations.

7.7.3 Function List for Devices

Event Handling		**Device MANDATORY**
Function	**Service**	**Requirement**
SetSynchronizationPoint	Event	M
CreatePullPointSubscription	Event	M
PullMessages	Event	M
GetEventProperties	Event	M
Unsubscribe	Event	M
Filter parameter of CreatePullPointSubscriptionRequest	Event	M
MessageContentFilterDialect http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter	Event	M

7.7.4 Function List for Clients

Event Handling		**Client MANDATORY**
Function	**Service**	**Requirement**
SetSynchronizationPoint	Event	M
CreatePullPointSubscription	Event	M
PullMessages	Event	M
GetEventProperties	Event	O
Unsubscribe	Event	O
Filter parameter of CreatePullPointSubscriptionRequest	Event	O
MessageContentFilterDialect http://www.onvif.org/ver10/tev/messageContentFilter/ItemFilter	Event	C*

*Client shall support this dialect if Message Content Filter is supported.

7.8 Access Point Information

This section describes the operations and events related to access point information.

7.8.1 Device requirements

- Device shall support returning paginated lists of access points using the **GetAccessPointInfoList** and **GetAccessPointList** operations. If the capability **MaxAccessPoints** is set to zero, then an empty list shall be returned.

- Device shall support returning specific access points using the **GetAccessPointInfo** and **GetAccessPoints** operations. If the capability **MaxAccessPoints** is set to zero, then an empty list shall be returned.
- If the device supports creating or modifying of access points in any way, the device shall generate an event whenever an **Access Point** is changed.
- If the device supports removing of access points in any way, the device shall generate an event whenever an **Access Point** is removed.

7.8.2 Client requirements

- Client shall be able to retrieve paginated lists of access points using the **GetAccessPointInfoList** operation.
- Client shall be able to receive **Access Point** changed events.
- Client shall be able to receive **Access Point** removed events.

7.8.3 Function List for Devices

Access Point Information		**Device MANDATORY**
Function	**Service**	**Requirement**
GetAccessPointInfoList	Access Control	M
GetAccessPointInfo	Access Control	M
GetAccessPointList	Access Control	M
GetAccessPoints	Access Control	M
tns1:Configuration/AccessPoint/Changed	Event	C
tns1:Configuration/AccessPoint/Removed	Event	C

7.8.4 Function List for Clients

Access Point Information		**Client MANDATORY**
Function	**Service**	**Requirement**
GetAccessPointInfoList	Access Control	M
GetAccessPointInfo	Access Control	O
GetAccessPointList	Access Control	O
GetAccessPoints	Access Control	O
tns1:Configuration/AccessPoint/Changed	Event	M
tns1:Configuration/AccessPoint/Removed	Event	M

7.9 Access Point State

This section describes the operations and events related to access point states.

7.9.1 Device requirements (if access points are supported)

- Device shall support returning an **Access Point State** using the **GetAccessPointState** operation.
- If the device supports changing access point state in any way, the device shall generate an event whenever an **Access Point State** is changed.

7.9.2 Client requirements

- Client shall be able to retrieve an **Access Point State** using the **GetAccessPointState** operation.
- Client shall be able to receive **Access Point State** changed events.

7.9.3 Function List for Devices

Access Point State		**Device CONDITIONAL**
Function	**Service**	**Requirement**
GetAccessPointState	Access Control	M
tns1:AccessPoint/State/Enabled	Event	C

7.9.4 Function List for Clients

Access Point State		**Client MANDATORY**
Function	**Service**	**Requirement**
GetAccessPointState	Access Control	M
tns1:AccessPoint/State/Enabled	Event	M

7.10 Access Control

This section describes the operations and events related to access control.

7.10.1 Device requirements (if access points are supported)

- Device shall support externally taken access granted and denied decisions using the **ExternalAuthorization** operation for at least one access point.

- Device shall be able to send anonymous and credential related access granted and denied events (implied by ExternalAuthorization support).
    - Note that even if a device sends a Request/Identifier event, the client may respond with an anonymous ExternalAuthorization call if the client could not identify the credential holder, e.g. if the identifier is a door PIN.
- If the device supports at least one feedback type, device shall support the **Feedback** operation.
- Device shall generate an event whenever a credential identifier access request is raised at an **Access Point**.
- If anonymous access is supported, device shall generate an event whenever an anonymous access request is raised at an **Access Point**.
- Device shall generate an event whenever an access request times out at an **Access Point**.

7.10.2 Client requirements

- Client shall be able to send access decisions using the **ExternalAuthorization** operation.
- Client shall be able to receive anonymous and credential related access granted and denied events (implied by ExternalAuthorization support)
- Client shall be able to send feedback indications using the **Feedback** operation.
- Client shall be able to receive credential identifier access request events.
- Client shall be able to receive anonymous access request events.
- Client shall be able to receive access request timeout events.

7.10.3 Function List for Devices

Access Control		**Device CONDITIONAL**
Function	**Service**	**Requirement**
ExternalAuthorization	Access Control	M
tns1:AccessControl/AccessGranted/Anonymous	Event	M
tns1:AccessControl/AccessGranted/Credential	Event	M
tns1:AccessControl/Denied/Anonymous	Event	M
tns1:AccessControl/Denied/Credential	Event	M
Feedback	Access Control	C
tns1:AccessControl/Request/Identifier	Event	M
tns1:AccessControl/Request/Anonymous	Event	C
tns1:AccessControl/Request/Timeout	Event	M

7.10.4 Function List for Clients

Access Control		Client MANDATORY
Function	**Service**	**Requirement**
ExternalAuthorization	Access Control	M
tns1:AccessControl/AccessGranted/Anonymous	Event	M
tns1:AccessControl/AccessGranted/Credential	Event	M
tns1:AccessControl/Denied/Anonymous	Event	M
tns1:AccessControl/Denied/Credential	Event	M
Feedback	Access Control	M
tns1:AccessControl/Request/Identifier	Event	M
tns1:AccessControl/Request/Anonymous	Event	M
tns1:AccessControl/Request/Timeout	Event	M

7.11 Access Taken

This section describes the events related to access taken.

7.11.1 Device requirements (if access points and access taken are supported)

- Device shall generate an event whenever a credential identifier access was taken at an **Access Point**.
- Device shall generate an event whenever a credential identifier access was not taken at an **Access Point**.
- If anonymous access is supported, device shall generate an event whenever an anonymous access was taken at an **Access Point**.
- If anonymous access is supported, device shall generate an event whenever an anonymous access was not taken at an **Access Point**.

7.11.2 Client requirements

- Client shall be able to receive taken access events when credential identifier was used.
- Client shall be able to receive not taken access events when credential identifier was used.
- Client shall be able to receive anonymously taken access events.
- Client shall be able to receive anonymously not taken access events.

7.11.3 Function List for Devices

Access Taken	Device CONDITIONAL	
Function	**Service**	**Requirement**
tns1:AccessControl/AccessTaken/Identifier	Event	M
tns1:AccessControl/AccessNotTaken/Identifier	Event	M
tns1:AccessControl/AccessTaken/Anonymous	Event	C
tns1:AccessControl/AccessNotTaken/Anonymous	Event	C

7.11.4 Function List for Clients

Access Taken	Client MANDATORY	
Function	**Service**	**Requirement**
tns1:AccessControl/AccessTaken/Identifier	Event	M
tns1:AccessControl/AccessNotTaken/Identifier	Event	M
tns1:AccessControl/AccessTaken/Anonymous	Event	M
tns1:AccessControl/AccessNotTaken/Anonymous	Event	M

7.12 Door Information

This section describes the operations and events related to door information.

7.12.1 Device requirements

- Device shall support returning paginated lists of doors using the **GetDoorInfoList** and **GetDoorList** operations. If the capability **MaxDoors** is set to zero, then an empty list shall be returned.
- Device shall support returning specific doors using the **GetDoorInfo** and **GetDoors** operations. If the capability **MaxDoors** is set to zero, then an empty list shall be returned.
- If the device supports creating or modifying of doors in any way, the device shall generate an event whenever a **Door** is changed.
- If the device supports removing of doors in any way, the device shall generate an event whenever a **Door** is removed.

7.12.2 Client requirements

- Client shall be able to retrieve paginated lists of doors using the **GetDoorInfoList** operation.
- Client shall be able to receive **Door** changed events.

- Client shall be able to receive **Door** removed events.

7.12.3 Function List for Devices

Door Information		**Device MANDATORY**
Function	**Service**	**Requirement**
GetDoorInfoList	Door Control	M
GetDoorInfo	Door Control	M
GetDoorList	Door Control	M
GetDoors	Door Control	M
tns1:Configuration/Door/Changed	Event	C
tns1:Configuration/Door/Removed	Event	C

7.12.4 Function List for Clients

Door Information		**Client MANDATORY**
Function	**Service**	**Requirement**
GetDoorInfoList	Door Control	M
GetDoorInfo	Door Control	O
GetDoorList	Door Control	O
GetDoors	Door Control	O
tns1:Configuration/Door/Changed	Event	M
tns1:Configuration/Door/Removed	Event	M

7.13 Door State

This section describes the operations and events related to door states.

7.13.1 Device requirements (if doors are supported)

- Device shall support returning a **Door State** using the **GetDoorState** operation.
- Device shall generate an event whenever a **Door Mode State** is changed.
- If door monitoring is supported, device shall generate an event whenever a **Door Physical State** is changed.
- If lock monitoring is supported, device shall generate an event whenever a **Lock Physical State** is changed.
- If double lock monitoring is supported, device shall generate an event whenever a **Double Lock Physical State** is changed.

- If door alarm detection is supported, device shall generate an event whenever a **Door Alarm State** is changed.
- If door tamper detection is supported, device shall generate an event whenever a **Door Tamper State** is changed.
- If door fault detection is supported, device shall generate an event whenever a **Door Fault State** is changed.

7.13.2 Client requirements

- Client shall be able to retrieve a **Door State** using the **GetDoorState** operation.
- Client shall be able to receive **Door Mode State** changed events.
- Client shall be able to receive **Door Physical State** changed events.
- Client shall be able to receive **Lock Physical State** changed events.
- Client shall be able to receive **Double Lock Physical State** changed events.
- Client shall be able to receive **Door Alarm State** changed events.
- Client shall be able to receive **Door Tamper State** changed events.
- Client shall be able to receive **Door Fault State** changed events.

7.13.3 Function List for Devices

Door State		**Device CONDITIONAL**
Function	**Service**	**Requirement**
GetDoorState	Door Control	M
tns1:Door/State/DoorMode	Event	M
tns1:Door/State/DoorPhysicalState	Event	C
tns1:Door/State/LockPhysicalState	Event	C
tns1:Door/State/DoubleLockPhysicalState	Event	C
tns1:Door/State/DoorAlarm	Event	C
tns1:Door/State/DoorTamper	Event	C
tns1:Door/State/DoorFault	Event	C

7.13.4 Function List for Clients

Door State		Client MANDATORY
Function	**Service**	**Requirement**
GetDoorState	Door Control	M
tns1:Door/State/DoorMode	Event	M
tns1:Door/State/DoorPhysicalState	Event	M
tns1:Door/State/LockPhysicalState	Event	M
tns1:Door/State/DoubleLockPhysicalState	Event	M
tns1:Door/State/DoorAlarm	Event	M
tns1:Door/State/DoorTamper	Event	M
tns1:Door/State/DoorFault	Event	M

7.14 Door Control

This section describes the operations and events related to door control.

7.14.1 Device requirements (if doors are supported)

- Device shall support momentarily accessing a door using the **AccessDoor** operation.
- Device shall support locking a door using the **LockDoor** operation.
- Device shall support unlocking a door using the **UnlockDoor** operation.
- If double locking a door is supported, device shall support the **DoubleLockDoor** operation.
- If blocking a door is supported, device shall support the **BlockDoor** operation.
- If locking down a door is supported, device shall support the **LockDownDoor** and **LockDownReleaseDoor** operations.
- If locking open a door is supported, device shall support the **LockOpenDoor** and **LockOpenReleaseDoor** operations.

7.14.2 Client requirements

- Client shall be able to momentarily access a door using the **AccessDoor** operation.
- Client shall be able to lock a door using the **LockDoor** operation.
- Client shall be able to unlock a door using the **UnlockDoor** operation.

7.14.3 Function List for Devices

Door Control		Device CONDITIONAL
Function	**Service**	**Requirement**
AccessDoor	Door Control	M
LockDoor	Door Control	M
UnlockDoor	Door Control	M
DoubleLockDoor	Door Control	C
BlockDoor	Door Control	C
LockDownDoor	Door Control	C
LockDownReleaseDoor	Door Control	C
LockOpenDoor	Door Control	C
LockOpenReleaseDoor	Door Control	C

7.14.4 Function List for Clients

Door Control		Client MANDATORY
Function	**Service**	**Requirement**
AccessDoor	Door Control	M
LockDoor	Door Control	M
UnlockDoor	Door Control	M
DoubleLockDoor	Door Control	O
BlockDoor	Door Control	O
LockDownDoor	Door Control	O
LockDownReleaseDoor	Door Control	O
LockOpenDoor	Door Control	O
LockOpenReleaseDoor	Door Control	O

8 Profile Conditional Features (normative)

The Profile Conditional Features section lists the features that shall be implemented if the device or client supports the feature. The requirements represent the minimum functionality that must be implemented for conformance.

8.1 Access Point Management

This section describes the operations related to access point management.

8.1.1 Device requirements (if supported)

- If access point management is supported, device shall support creating, modifying and deleting **Access Points** using the **CreateAccessPoint**, **ModifyAccessPoint** and **DeleteAccessPoint** operations.
- If client-supplied token is supported, device shall support creating/modifying **Access Points** using the **SetAccessPoint** operation.
- Device shall generate an event whenever an **Access Point** is changed.
- Device shall generate an event whenever an **Access Point** is removed.

8.1.2 Client requirements (if supported)

- If access point management is supported in any way, client shall be able to create, modify and delete **Access Points** using the **CreateAccessPoint**, **ModifyAccessPoint** and **DeleteAccessPoint** operations.
- Client shall be able to receive **Access Point** changed events.
- Client shall be able to receive **Access Point** removed events.

8.1.3 Function List for Devices

Access Point Management		**Device CONDITIONAL**
Function	**Service**	**Requirement**
CreateAccessPoint	Access Control	M
ModifyAccessPoint	Access Control	M
DeleteAccessPoint	Access Control	M
SetAccessPoint	Access Control	M
tns1:Configuration/AccessPoint/Changed	Event	C
tns1:Configuration/AccessPoint/Removed	Event	C

8.1.4 Function List for Clients

Access Point Management		Client CONDITIONAL
Function	**Service**	**Requirement**
CreateAccessPoint	Access Control	M
ModifyAccessPoint	Access Control	M
DeleteAccessPoint	Access Control	M
SetAccessPoint	Access Control	O
tns1:Configuration/AccessPoint/Changed	Event	M
tns1:Configuration/AccessPoint/Removed	Event	M

8.2 Access Point Control

This section describes the operations related to access point control.

8.2.1 Device requirements (if supported)

- If disable access point is supported, device shall support enabling and disabling **Access Points** using the **EnableAccessPoint** and **DisableAccessPoint** operations.

8.2.2 Client requirements (if supported)

- If access point control is supported in any way, client shall be able to enable and disable **Access Points** using the **EnableAccessPoint** and **DisableAccessPoint** operations.

8.2.3 Function List for Devices

Access Point Control		Device CONDITIONAL
Function	**Service**	**Requirement**
EnableAccessPoint	Access Control	M
DisableAccessPoint	Access Control	M

8.2.4 Function List for Clients

Access Point Control		Client CONDITIONAL
Function	**Service**	**Requirement**
EnableAccessPoint	Access Control	M
DisableAccessPoint	Access Control	M

8.3 Door Management

This section describes the operations related to door management.

8.3.1 Device requirements (if supported)

- If door management is supported, device shall support creating, modifying and deleting **Doors** using the **CreateDoor**, **ModifyDoor** and **DeleteDoor** operations.
- If client-supplied token is supported, device shall support creating/modifying **Doors** using the **SetDoor** operation.
- If the device supports creating or modifying of doors in any way, the device shall generate an event whenever a **Door** is changed.
- If the device supports removing of doors in any way, the device shall generate an event whenever a **Door** is removed.

8.3.2 Client requirements (if supported)

- If door management is supported in any way, client shall be able to create, modify and delete **Doors** using the **CreateDoor**, **ModifyDoor** and **DeleteDoor** operations.
- Client shall be able to receive **Door** changed events.
- Client shall be able to receive **Door** removed events.

8.3.3 Function List for Devices

Door Management		**Device CONDITIONAL**
Function	**Service**	**Requirement**
CreateDoor	Door Control	M
ModifyDoor	Door Control	M
DeleteDoor	Door Control	M
SetDoor	Door Control	M
tns1:Configuration/Door/Changed	Event	C
tns1:Configuration/Door/Removed	Event	C

8.3.4 Function List for Clients

Door Management		Client CONDITIONAL
Function	**Service**	**Requirement**
CreateDoor	Door Control	M
ModifyDoor	Door Control	M
DeleteDoor	Door Control	M
SetDoor	Door Control	O
tns1:Configuration/Door/Changed	Event	M
tns1:Configuration/Door/Removed	Event	M

8.4 Credential Format Types

This section describes the operations related to credential format types.

8.4.1 Device requirements (if white- or blacklisting is supported)

- Device shall support returning the supported credential format types using the **GetSupportedFormatTypes** operation.

8.4.2 Client requirements (if white- or blacklisting is supported)

- Client shall be able to retrieve the supported credential format types using the **GetSupportedFormatTypes** operation.

8.4.3 Function List for Devices

Credential Format Types		Device CONDITIONAL
Function	**Service**	**Requirement**
GetSupportedFormatTypes	Credential	M

8.4.4 Function List for Clients

Credential Format Types		Client CONDITIONAL
Function	**Service**	**Requirement**
GetSupportedFormatTypes	Credential	M

8.5 Credential Whitelisting

This section describes the operations related to whitelisting of credential identifiers.

8.5.1 Device requirements (if supported)

- Device shall support getting, modifying and deleting **Whitelists** using the **GetWhitelist**, **AddToWhitelist**, **RemoveFromWhitelist** and **DeleteWhitelist** operations.
- Device shall generate an event whenever a credential identifier access has been granted using the **Whitelist**.

8.5.2 Client requirements (if supported)

- If whitelisting is supported in any way, client shall be able to manage **Whitelists** using the **GetWhitelist**, **AddToWhitelist**, **RemoveFromWhitelist** and **DeleteWhitelist** operations.
- Client shall be able to receive access granted events when credential identifier was whitelisted.

8.5.3 Function List for Devices

Credential Whitelisting		**Device CONDITIONAL**
Function	**Service**	**Requirement**
GetWhitelist	Credential	M
AddToWhitelist	Credential	M
RemoveFromWhitelist	Credential	M
DeleteWhitelist	Credential	M
tns1:AccessControl/AccessGranted/Identifier	Event	M

8.5.4 Function List for Clients

Credential Whitelisting		**Client CONDITIONAL**
Function	**Service**	**Requirement**
GetWhitelist	Credential	M
AddToWhitelist	Credential	M
RemoveFromWhitelist	Credential	M
DeleteWhitelist	Credential	M
tns1:AccessControl/AccessGranted/Identifier	Event	M

8.6 Credential Blacklisting

This section describes the operations related to blacklisting of credential identifiers.

8.6.1 Device requirements (if supported)

- Device shall support getting, modifying and deleting **Blacklists** using the **GetBlacklist**, **AddToBlacklist**, **RemoveFromBlacklist** and **DeleteBlacklist** operations.
- Device shall generate an event whenever a credential identifier access has been denied using the **Blacklist**.

8.6.2 Client requirements (if supported)

- If blacklisting is supported in any way, client shall be able to manage **Blacklists** using the **GetBlacklist**, **AddToBlacklist**, **RemoveFromBlacklist** and **DeleteBlacklist** operations.
- Client shall be able to receive access denied events when credential identifier was blacklisted.

8.6.3 Function List for Devices

Credential Blacklisting		**Device CONDITIONAL**
Function	**Service**	**Requirement**
GetBlacklist	Credential	M
AddToBlacklist	Credential	M
RemoveFromBlacklist	Credential	M
DeleteBlacklist	Credential	M
tns1:AccessControl/Denied/Identifier	Event	M

8.6.4 Function List for Clients

Credential Blacklisting		**Client CONDITIONAL**
Function	**Service**	**Requirement**
GetBlacklist	Credential	M
AddToBlacklist	Credential	M
RemoveFromBlacklist	Credential	M
DeleteBlacklist	Credential	M
tns1:AccessControl/Denied/Identifier	Event	M