ONVIF™

Schedule Service Specification

Draft

Version

June 2015



 2008-2015 by ONVIF: Open Network Video Interface Forum Inc. All rights reserved.

Recipients of this document may copy, distribute, publish, or display this document so long as this copyright notice, license and disclaimer are retained with all copies of the document. No license is granted to modify this document.

THIS DOCUMENT IS PROVIDED "AS IS," AND THE CORPORATION AND ITS MEMBERS AND THEIR AFFILIATES, MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NON-INFRINGEMENT, OR TITLE; THAT THE CONTENTS OF THIS DOCUMENT ARE SUITABLE FOR ANY PURPOSE; OR THAT THE IMPLEMENTATION OF SUCH CONTENTS WILL NOT INFRINGE ANY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

IN NO EVENT WILL THE CORPORATION OR ITS MEMBERS OR THEIR AFFILIATES BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES, ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT, WHETHER OR NOT (1) THE CORPORATION, MEMBERS OR THEIR AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR (2) SUCH DAMAGES WERE REASONABLY FORESEEABLE, AND ARISING OUT OF OR RELATING TO ANY USE OR DISTRIBUTION OF THIS DOCUMENT.  THE FOREGOING DISCLAIMER AND LIMITATION ON LIABILITY DO NOT APPLY TO, INVALIDATE, OR LIMIT REPRESENTATIONS AND WARRANTIES MADE BY THE MEMBERS AND THEIR RESPECTIVE AFFILIATES TO THE CORPORATION AND OTHER MEMBERS IN CERTAIN WRITTEN POLICIES OF THE CORPORATION.

Contents

[1 Scope 6](#_Toc421875012)

[1.1 Normative references 6](#_Toc421875013)

[2 Terms and Definitions 7](#_Toc421875014)

[2.1 Conventions 7](#_Toc421875015)

[2.2 Definitions 7](#_Toc421875016)

[2.3 Abbreviations 7](#_Toc421875017)

[3 Overview 8](#_Toc421875018)

[3.1 Schedules 8](#_Toc421875019)

[3.2 Special Days 9](#_Toc421875020)

[4 Schedules 12](#_Toc421875021)

[4.1 Service Capabilities 12](#_Toc421875022)

[4.1.1 Data Structures 12](#_Toc421875023)

[4.1.2 GetServiceCapabilities command 13](#_Toc421875024)

[4.2 Schedule Information 13](#_Toc421875025)

[4.2.1 Data Structures 13](#_Toc421875026)

[4.2.2 GetScheduleInfo command 16](#_Toc421875027)

[4.2.3 GetScheduleInfoList command 17](#_Toc421875028)

[4.2.4 GetSchedules command 18](#_Toc421875029)

[4.2.5 GetScheduleList command 19](#_Toc421875030)

[4.2.6 CreateSchedule command 20](#_Toc421875031)

[4.2.7 ModifySchedule command 21](#_Toc421875032)

[4.2.8 DeleteSchedule command 22](#_Toc421875033)

[4.3 Special Day Group Information 22](#_Toc421875034)

[4.3.1 Data Structures 22](#_Toc421875035)

[4.3.2 GetSpecialDayGroupInfo command 23](#_Toc421875036)

[4.3.3 GetSpecialDayGroupInfoList command 24](#_Toc421875037)

[4.3.4 GetSpecialDayGroups command 25](#_Toc421875038)

[4.3.5 GetSpecialDayGroupList command 26](#_Toc421875039)

[4.3.6 CreateSpecialDayGroup command 27](#_Toc421875040)

[4.3.7 ModifySpecialDayGroup command 28](#_Toc421875041)

[4.3.8 DeleteSpecialDayGroup command 29](#_Toc421875042)

[4.4 Schedule Status 29](#_Toc421875043)

[4.4.1 Data Structures 29](#_Toc421875044)

[4.4.2 GetScheduleState command 30](#_Toc421875045)

[5 Notification topics 31](#_Toc421875046)

[5.1 Event overview (informative) 31](#_Toc421875047)

[5.2 Status changes 31](#_Toc421875048)

[5.2.1 Schedule 31](#_Toc421875049)

[5.3 Configuration changes 32](#_Toc421875050)

[5.3.1 Schedule 32](#_Toc421875051)

[5.3.2 Special Days 32](#_Toc421875052)

[Annex A - Examples 33](#_Toc421875053)

[*A.1* *Access 24\*7 for admin staff* 33](#_Toc421875054)

[*A.2* *Access on Monday and Wednesday from 6 AM to 8 PM for cleaning staff* 33](#_Toc421875055)

[*A.3* *Access from Friday 6 PM to Monday 7 AM for maintenance staff* 33](#_Toc421875056)

[*A.4* *Access on Weekdays from 8 AM to 5 PM for employees* 34](#_Toc421875057)

[*A.5* *Access from January 15, 2014, to January 14, 2015, from 9 AM to 6 PM* 34](#_Toc421875058)

[*A.6* *Special Days example 1* 35](#_Toc421875059)

[*A.7* *Special Days example 2* 36](#_Toc421875060)

[Annex B. Revision History 37](#_Toc421875061)

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 |

# Scope

This specification defines the web service interface for interaction with ONVIF devices which support time management features such as schedules and special days (sometimes referred to as holidays).

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

## Normative references

* **ONVIF Network Interface Specification Set**

ONVIF Network Interface Specification Set version 2.5 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 1.0.3. Part of [ONVIF Network Interface Specification Set]

* **Access Rules Service Specification**

ONVIF Access Rules Service Specification version 1.0 (IPR Draft). Part of [ONVIF Network Interface Specification Set]

* **Credential Service Specification**

ONVIF Credential Service Specification version 1.0 (IPR Draft). Part of [ONVIF Network Interface Specification Set]

* **RFC 2445, Internet Calendaring and Scheduling Core Object Specification (iCalendar)**

<https://www.ietf.org/rfc/rfc2445.txt>

* **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>

# Terms and Definitions

## 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].

## Definitions

|  |  |
| --- | --- |
| **iCalendar** | An industry standard format for exchanging scheduling and activity-recording information electronically. |
| **Schedule** | A set of time periods, e.g. working hours (weekdays from 8 AM to 6 PM). It may also include one or more Special Days Schedules. |
| **Special Days** | A set of dates that require the regular Schedule to be overridden, e.g. holidays, half-days or working Sundays. |
| **Special Days Schedule** | A schedule that defines time periods for a Special Day List. |
| **Time Period** | A time period has a start time and an end time, e.g. 8 AM to 6 PM. |
| **vEvent** | A component in iCalendar, specifying the properties of an event. |

## Abbreviations

|  |  |
| --- | --- |
| **PACS** | Physical Access Control System |

# Overview

The schedule service provides functions to manage schedules and special days. Schedules and special days are defined by iCalendar vEvents. iCalendar is a very flexible format where daily, weekly, monthly and yearly recurrence can be used.

To briefly explain the iCalendar format, consider the following examples (only parts of the iCalendar structures are shown below):

An iCalendar event starting at 8 PM on November 25, 2014, and ending at 2 AM on November 26, 2014:

BEGIN:VCALENDAR

BEGIN:VEVENT

DTSTART:20141125T200000

DTEND:20141126T020000

END:VEVENT

END:VCALENDAR

A weekly recurring iCalendar event each Tuesday and Thursday from 2 PM to 4 PM, where the first occurrence is on November 25, 2014, and the last occurrence is on December 9, 2014 (5 occurrences):

BEGIN:VCALENDAR

BEGIN:VEVENT

DTSTART:20141125T140000

DTEND:20141125T160000

RRULE:FREQ=WEEKLY;COUNT=5;BYDAY=TU,TH

END:VEVENT

END:VCALENDAR

The following example have exactly the same effect as the previous example, but the end date of the series is specified instead of the number of occurrences:

BEGIN:VCALENDAR

BEGIN:VEVENT

DTSTART:20141125T140000

DTEND:20141125T160000

RRULE:FREQ=WEEKLY;UNTIL:20141209T140000;BYDAY=TU,TH

END:VEVENT

END:VCALENDAR

## Schedules

Schedules can, for example, be used to define when access is granted or when a video camera should record.

Single occurrences in a series of recurring events can be deleted to create exceptions from the recurring events.

The following picture illustrates a recurring series of events. Each vertical bar represents the 24-hour day, and the green parts represent events (time periods). Events occur each Monday, but there is an exception on the fourth Monday, when no event occurs.



Figure 1: Recurring events with an exception

## Special Days

In contrast to exceptions, the schedule service also support special days. Special days are used to define different behaviour during for instance holidays or half working days. Examples of different behaviour includes video recording with a different frame rate or different access times to a facility. Special days are grouped together and given names such as “US holidays”, “UK holidays”, “Swedish half-days”, etc.

The following picture illustrates the same series of recurring events, but instead of an exception, the fourth Monday is treated as per special day with different behaviour..



Figure 2: Recurring events with a special day

In the PACS domain, supporting different behaviour during special days is essential. Most traditional access controllers on the market today only handle weekly recurrence. Generally they do not support exceptions, but they usually support special days.

Each device supporting the ONVIF Schedule Service, can implement full iCalendar support, or support weekly recurrence with special days, or any other combination in-between. This information is published using the service capabilities.

When the ExtendedRecurrenceSupported capability is set to true then full iCalendar recurrence format is supported. Special days can still be used but may not be needed.

For devices not supporting full iCalendar recurrence, the ExtendedRecurrenceSupported capability should be set to false. The iCalendar “FREQ” and “BYDAY” keywords are used to define weekly recurring events per day in the week (see iCalendar examples above). When the ExtendedRecurrenceSupported capability is set to false, then the year of the start date of recurring events shall be set to 1970, and the occurrence count (optionally the until date) shall be omitted (i.e. forever recurring events).

The following table shows the effect of the ExtendedRecurrenceSupported capability settings for schedule and special days respectively.

For instance, the last row of the table shows that if ExtendedRecurrenceSupported is set to true then for Special Days both non-recurring events and all types of recurring events are allowed.

But, as shown in the second last row, if ExtendedRecurrenceSupported is set to false then for Special Days only non-recurring events are allowed.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | **Event** | | | | |
| **Non-recurring events allowed?** | **Recurring events allowed?** | | | |
| **ExtendedRecurrenceSupported** | **Entity Type** | **Daily** | **Weekly** | **Monthly** | **Yearly** |
| FALSE | Schedule | NO | NO | YES | NO | NO |
| TRUE | YES | YES | YES | YES | YES |
| FALSE | Special Days | YES | NO | NO | NO | NO |
| TRUE | YES | YES | YES | YES | YES |

Figure 3: Extended Recurrence supported matrix

A group of special days may be used to define what times the facility is accessible during holidays. Another group of special days may be used to define what times the facility is accessible during half-working days.

Special day groups are typically reused by different schedules. Consider the following example:

A PACS system is shared by two sites; one in Sweden, and one in Poland. All Polish holidays are grouped in a special day group called "Polish holidays". All Swedish holidays are grouped in a special day group called "Swedish holidays". In addition Sweden have half-working days before some of the bank holidays, so another group is defined called "Swedish half-days".

A schedule called "Polish employee schedule" is created, defining access to the office for Polish employees during all days of the week, including the weekend (e.g. access on Mondays-Fridays from 7 AM to 8 PM, and access on Saturday-Sunday from 10AM to 2PM). The schedule is linked to the "Polish holidays" group defining the access to work during these days (e.g. no access during holidays).

Similarly, a schedule called "Swedish employee schedule" is created, defining access to the office for all Swedish employees during all days of the week, including the weekend. The schedule is linked to both the "Swedish holidays" group (e.g. no access) and the "Swedish half-days" group (e.g. access from 7 AM to 1 PM).

Only a certain subset of the iCalendar specification is required to be supported to get a usable service. Support for additional features within the iCalendar specification is reflected in the ServiceCapabilities.

The schedule service supports schedules in both local time and UTC time. Typically a schedule is in local time.

The following picture shows the main data structures involved in the schedule service:



Figure 4: Main data structures in the schedule service

# Schedules

## 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.

### Data Structures

#### 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.

* **MaxSchedules**

Indicates the maximum number of schedules the device supports. The device shall support at least one schedule.

* **MaxTimePeriodsPerDay**

Indicates the maximum number of time periods per day the device supports in a schedule including special days schedule. The device shall support at least one time period per day.

* **MaxSpecialDayGroups**

Indicates the maximum number of special day group entities the device supports. The device shall support at least one ‘SpecialDayGroup’ entity.

* **MaxDaysInSpecialDayGroup**

Indicates the maximum number of days per ‘SpecialDayGroup’ entity the device supports. The device shall support at least one day per ‘SpecialDayGroup’ entity.

* **MaxSpecialDaysSchedules**

Indicates the maximum number of ‘SpecialDaysSchedule’ entities referred by a schedule that the device supports.

* **SpecialDaysSupported**

If this capability is supported, then the device shall support special days.

* **ExtendedRecurrenceSupported**

***For schedules****:*

If this capability is supported, then all iCalendar recurrence types shall be supported by the device. The device shall also support the start and end dates (or iCalendar occurrence count) in recurring events (see iCalendar examples in section 3).

If this capability is not supported, then only the weekly iCalendar recurrence type shall be supported. Non-recurring events and other recurring types are not supported. The device shall only accept a start date with the year ‘1970’ (the month and day is needed to reflect the week day of the recurrence) and will not accept an occurrence count (or iCalendar until date) in recurring events.

***For special days*** *(only applicable if SpecialDaysSupported is set to true):*

If this capability is supported, then all iCalendar recurrence types shall be supported by the device. The device shall also support the start and end dates (or occurrence count) in recurring events.

If this capability is not supported, then only non-recurring special days are supported.

* **StateReportingSupported**

If this capability is set to true, the device shall implement the GetScheduleState command, and shall notify subscribing clients whenever schedules become active or inactive.

### GetServiceCapabilities command

This operation returns the capabilities of the schedule 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 schedule service capabilities using a hierarchical XML capability structure.*   tsc:ServiceCapabilities **Capabilities [1][1]** | |

## Schedule Information

The schedule is a set of time periods which may also include one or more special day’s schedules.

### Data Structures

#### ScheduleInfo

The ScheduleInfo type represents the schedule as a physical object. The structure contains information of a specific schedule instance.

The device shall provide the following fields for each ScheduleInfo instance:

* **token**

A service unique identifier of the schedule.

* **Name**

A user readable name. It shall be up to 64 characters.

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

* **Description**

User readable description for the schedule. It shall be up to 1024 characters.

#### Schedule

The schedule structure shall include all properties of the ScheduleInfo structure and also the standard events (iCalendar format) and a list of SpecialDaysSchedule instances.

The device shall provide the following fields for each schedule instance:

* **Standard**

An iCalendar structure that defines a number of events. Events can be recurring or non-recurring. The events can, for instance, be used to control when a camera should record or when a facility is accessible.

Some devices might not be able to fully support all the features of iCalendar. Setting the service capability ExtendedRecurrenceSupported to false will enable more devices to be ONVIF compliant. Is of type string (but contains an iCalendar structure).

* **SpecialDays**

For devices that are not able to support all the features of iCalendar, supporting special days is essential. Each SpecialDaysSchedule instance defines an alternate set of time periods that overrides the regular schedule for a specified list of special days. Is of type SpecialDaysSchedule.

#### SpecialDaysSchedule

A override schedule that defines alternate time periods for a group of special days.

The picture below shows the use of SpecialDaysSchedule instances. A Schedule instance called “Employee schedule” holds an iCalendar structure defining the regular access, and also holds a number of SpecialDaysSchedule instances. Each SpecialDaysSchedule holds a token to a SpecialDayGroup and zero or more TimePeriod instances.

In this example, the first SpecialDaysSchedule instance holds the token to the “Holidays” special day group, but holds no time periods. The result is that no access is granted on the special days listed in the “Holidays” special day group.

The second SpecialDaysSchedule instance holds the token to the “Half-days” special day group, and also holds one time period “7 AM to 1 PM”, resulting in access granted during those hours for special days listed in the “Half-days” special day group.



Figure 5: SpecialDaysSchedule

The device shall provide the following fields for each SpecialDaysSchedule instance:

* **GroupToken**

Indicates the list of special days in a schedule.

* **TimeRange**

Indicates the alternate time periods for the list of special days (overrides the regular schedule). For example, the regular schedule indicates that it is active from 8AM to 5PM on Mondays. However, this particular Monday is a special day, and the alternate time periods state that the schedule is active from 9 AM to 11 AM and 1 PM to 4 PM.

If no time periods are defined, then no access is allowed. Is of type TimePeriod.

#### TimePeriod

A time period defines a start and end time. For full day access, the start time =”00:00:00” with no defined end time. For a time period with no end time, the schedule runs until midnight. The end time must always be greater than the start time, otherwise an InvalidArgVal error messages is generated by the device.

The device shall provide the following fields for each TimePeriod instance:

* **From**

Indicates the start time.

* **Until**

Indicates the end time. Is optional, if omitted, the period ends at midnight. The end time is exclusive, meaning that that exact moment in time is not part of the period. To determine if a moment in time (t) is part of a time period, the formula StartTime ≤ t < EndTime is used.

### GetScheduleInfo command

This method returns a list of schedule info items, specified in the request. Only found schedules shall be returned, i.e., the returned numbers of elements can differ from the requested element.

The device shall ignore tokens it cannot resolve and shall return an empty list if there are no items matching the specified tokens.

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

Table 2 GetScheduleInfo command

|  |  |  |
| --- | --- | --- |
| **GetScheduleInfo** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetScheduleInfoRequest | *This message contains:*   * "*Token": Tokens of ScheduleInfo items to get.*   pt:ReferenceToken **Token [1][unbounded]** | |
| GetScheduleInfoResponse | *This message contains:*   * *"ScheduleInfo": List of ScheduleInfo items.*   tsc:ScheduleInfo **ScheduleInfo [0][unbounded]** | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgs  ter:TooManyItems | *Too many items were requested, see MaxLimit capability.* | |

### GetScheduleInfoList command

This operation requests a list of all of ScheduleInfo 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 3 GetScheduleInfoList command

|  |  |  |
| --- | --- | --- |
| **GetScheduleInfoList** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetScheduleInfoListRequest | *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]** | |
| GetScheduleInfoListResponse | *This message contains:*   * *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* * *"ScheduleInfo": List of ScheduleInfo items.*   xs:string **NextStartReference [0][1]** tsc:ScheduleInfo **ScheduleInfo [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.* | |

### GetSchedules command

This operation returns the specified schedule item 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.

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

Table 4 GetSchedules command

|  |  |  |
| --- | --- | --- |
| **GetSchedules** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetScheduleRequest | *This message contains:*   * "*Token": Tokens of Schedule items to get*   pt:ReferenceToken **Token [1][unbounded]** | |
| GetScheduleResponse | *This message contains:*   * *"Schedule": List of schedule items.*   tsc:Schedule **Schedule [0][unbounded]** | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgs  ter:TooManyItems | *Too many items were requested, see MaxLimit capability.* | |

### GetScheduleList command

This operation requests a list of all of Schedule 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 GetScheduleList command

|  |  |  |
| --- | --- | --- |
| **GetScheduleList** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetScheduleListRequest | *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]** | |
| GetScheduleListResponse | *This message contains:*   * *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* * *"Schedule": List of Schedule items.*   xs:string **NextStartReference [0][1]** tsc:Schedule **Schedule [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.* | |

### CreateSchedule command

This operation creates the specified schedule. The token field of the schedule structure shall be empty, the device shall allocate a token for the schedule. 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 CreateSchedule command

|  |  |  |
| --- | --- | --- |
| **CreateSchedule** | | Access Class: WRITE\_SYSTEM |
| **Message name** | **Description** | |
| CreateScheduleRequest | *This message contains:*   * *"Schedule": The Schedule to create*   tsc:Schedule **Schedule [1][1]** | |
| CreateScheduleResponse | *This message contains:*   * *"Token": The token of created Schedule*   pt:ReferenceToken **Token[1][1]** | |
| **Fault codes** | **Description** | |
| env:Receiver  ter:CapabilityViolated  ter:MaxSchedules | *There is not enough space to add new schedule, see MaxSchedules capability* | |
| env:Sender  ter: CapabilityViolated  ter:MaxSpecialDaysSchedules | *There are too many* SpecialDaysSchedule entities *referred in this schedule, see MaxSpecialDays­Schedules capability.* | |
| env:Sender  ter: CapabilityViolated  ter:MaxTimePeriodsPerDay | *There are too many TimePeriods in a day schedule, see MaxTimePeriodsPerDay capability.* | |
| env:Sender  ter:InvalidArgVal   ter:ReferenceNotFound | *A referred entity token is not found (some devices may not validate referred entities).* | |

### ModifySchedule command

This operation modifies or updates the specified schedule.

Table 7 ModifySchedule command

|  |  |  |
| --- | --- | --- |
| **ModifySchedule** | | Access Class: WRITE\_SYSTEM |
| **Message name** | **Description** | |
| ModifyScheduleRequest | *This message contains:*   * *"Schedule": The Schedule to modify/update*   tsc:Schedule **Schedule [1][1]** | |
| ModifyScheduleResponse | *This message shall be empty* | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgVal  ter:NotFound | *The specified token not found.* | |
| env:Sender  ter:CapabilityViolated  ter:MaxSpecialDaysSchedules | *There are too many SpecialDaysSchedule entities referred in this schedule, see MaxSpecialDaysSchedules capability.* | |
| env:Sender  ter: CapabilityViolated  ter:MaxTimePeriodsPerDay | *There are too many TimePeriods in a day schedule, see MaxTimePeriodsPerDay capability.* | |
| env:Sender  ter:InvalidArgVal   ter:ReferenceNotFound | *A referred entity token is not found (some devices may not validate referred entities).* | |

### DeleteSchedule command

This operation will delete the specified schedule.

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

Table 8 DeleteSchedule command

|  |  |  |
| --- | --- | --- |
| **DeleteSchedule** | | Access Class: WRITE\_SYSTEM |
| **Message name** | **Description** | |
| DeleteScheduleRequest | *This message contains:*   * *"Token": The token of the schedule to delete.*   pt:ReferenceToken **Token [1][1]** | |
| DeleteScheduleResponse | *This message shall be empty* | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgVal  ter:NotFound | *Schedule token is not found.* | |
| env:Sender  ter:InvalidArgVal   ter:ReferenceInUse | *Failed to delete, Schedule token is in use* | |

## Special Day Group Information

A special day group are days (or parts of days) that require the regular schedule to be overridden with an alternate schedule. For example holidays, half-days, working Sundays , etc.

### Data Structures

#### SpecialDayGroupInfo

The SpecialDayGroupInfo structure contains the basic information about the special days list.

The device shall provide the following fields for each SpecialDayGroupInfo instance:

* **token**

A service-unique identifier of the special day group.

* **Name**

User readable name. It shall be up to 64 characters.

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

* **Description**

User readable description for the special days. It shall be up to 1024 characters.

#### SpecialDayGroup

The special day group structure shall include all properties of the SpecialDayGroupInfo structure and also a set of special days. A special day group are days (or parts of days) that require the regular schedule to be overridden with an alternate schedule. For example holidays, half-days, working Sundays, etc.

The device shall provide the following fields for each SpecialDayGroup instance:

* **Days**

An iCalendar structure that contains a group of special days. Is of type string (containing an iCalendar structure).

### GetSpecialDayGroupInfo command

This operation requests a list of SpecialDayGroupInfo 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 9 GetSpecialDayGroupInfo command

|  |  |  |
| --- | --- | --- |
| **GetSpecialDayGroupInfo** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetSpecialDayGroupInfoRequest | *This message contains:*   * "*Token": Tokens of SpecialDayGroupInfo items to get.*   pt:ReferenceToken **Token [1][unbounded]** | |
| GetSpecialDayGroupInfoResponse | *This message contains:*   * *"SpecialDayGroupInfo": List of SpecialDayGroupInfo items.*   tsc:SpecialDayGroupInfo **SpecialDayGroupInfo [0][unbounded]** | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgs  ter:TooManyItems | *Too many items were requested, see MaxLimit capability.* | |

### GetSpecialDayGroupInfoList command

This operation requests a list of all of SpecialDayGroupInfo 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.

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

Table 10 GetSpecialDayGroupInfoList command

|  |  |  |
| --- | --- | --- |
| **GetSpecialDayGroupInfoList** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetSpecialDayGroupInfoListRequest | *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]** | |
| GetSpecialDayGroupInfoListResponse | *This message contains:*   * *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* * *"SpecialDayGroupInfo": List of SpecialDayGroupInfo items.*   xs:string **NextStartReference [0][1]** tsc:SpecialDayGroupInfo **SpecialDayGroupInfo [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.* | |

### GetSpecialDayGroups command

This operation returns the specified special day group item matching the given token.

Table 11 GetSpecialDayGroups command

|  |  |  |
| --- | --- | --- |
| **GetSpecialDayGroups** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetSpecialDayGroupsRequest | *This message contains:*   * "*Token": Tokens of the SpecialDayGroup items to get.*   pt:ReferenceToken **Token [1][unbounded]** | |
| GetSpecialDayGroupsResponse | *This message contains:*   * *"SpecialDayGroup": List of SpecialDayGroup items.*   tsc:SpecialDayGroup **SpecialDayGroup [0][unbounded]** | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgs  ter:TooManyItems | *Too many items were requested, see MaxLimit capability.* | |

### GetSpecialDayGroupList command

This operation requests a list of all of SpecialDayGroupList 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 12 GetSpecialDayGroupList command

|  |  |  |
| --- | --- | --- |
| **GetSpecialDayGroupList** | | Access Class: READ\_SYSTEM |
| **Message name** | **Description** | |
| GetSpecialDayGroupListRequest | *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]** | |
| GetSpecialDayGroupListResponse | *This message contains:*   * *"NextStartReference": StartReference to use in next call to get the following items. If absent, no more items to get.* * *"SpecialDayGroup": List of SpecialDayGroup items.*   xs:string **NextStartReference [0][1]** tsc:SpecialDayGroup **SpecialDayGroup [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.* | |

### CreateSpecialDayGroup command

This operation creates the specified special day group. The token field of the SpecialDayGroup structure shall be empty, the device shall allocate a token for the special day group. The allocated token shall be returned in the response. If there is any value in the token field, the device shall return InvalidArgVal as generic fault code.

Table 13 CreateSpecialDayGroup command

|  |  |  |
| --- | --- | --- |
| **CreateSpecialDayGroup** | | Access Class: WRITE\_SYSTEM |
| **Message name** | **Description** | |
| CreateSpecialDayGroupRequest | *This message contains:*   * *"SpecialDayGroup": The special day group to create.*   tsc:SpecialDayGroup **SpecialDayGroup [1][1]** | |
| CreateSpecialDayGroupResponse | *This message contains:*   * *"Token": The token of created special day group.*   pt:ReferenceToken **Token [1][1]** | |
| **Fault codes** | **Description** | |
| env:Receiver  ter: CapabilityViolated  ter:MaxSpecialDayGroups | *There is not enough space to add new SpecialDayGroups, see the MaxSpecialDayGroups capability* | |
| env:Sender  ter: CapabilityViolated  ter:MaxDaysInSpecialDayGroup | *There are too many special days in a SpecialDayGroup, see MaxDaysInSpecialDayGroup capability.* | |

### ModifySpecialDayGroup command

This operation updates the specified special day group.

Table 14 ModifySpecialDayGroup command

|  |  |  |
| --- | --- | --- |
| **ModifySpecialDayGroup** | | Access Class: WRITE\_SYSTEM |
| **Message name** | **Description** | |
| ModifSpecialDayGroupRequest | *This message contains:*   * *"SpecialDayGroup": The special day group to modify/update.*   tsc:SpecialDayGroup **SpecialDayGroup [1][1]** | |
| ModifySpecialDayGroupResponse | *This message shall be empty* | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgVal  ter:NotFound | *The specified token not found.* | |
| env:Sender  ter: CapabilityViolated  ter:MaxDaysInSpecialDayGroup | *There are too many special days in a SpecialDayGroup, see MaxDays­In­SpecialDay­Group capability.* | |

### DeleteSpecialDayGroup command

This method deletes the specified special day group.

If it is associated with one or more schedules some devices may not be able to delete the special day group, and consequently a ReferenceInUse fault must be generated.

Table 15 DeleteSpecialDayGroup command

|  |  |  |
| --- | --- | --- |
| **DeleteSpecialDayGroup** | | Access Class: WRITE\_SYSTEM |
| **Message name** | **Description** | |
| DeleteSpecialDayGroupRequest | *This message contains:*   * *"Token": The token of the special day group item to delete.*   pt:ReferenceToken **Token [1][1]** | |
| DeleteSpecialDayGroupResponse | *This message shall be empty* | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgVal  ter:NotFound | *Special day group token is not found.* | |
| env:Sender  ter:InvalidArgVal   ter:ReferenceInUse | *Failed to delete, Special day group token is in use* | |

## Schedule Status

### Data Structures

#### ScheduleState

The ScheduleState contains state information for a schedule.

The device shall provide the following fields for each ScheduleState instance:

* **Active**

Indicates that the current time is within the boundaries of the schedule or its special days schedules’ time periods. For example, if this schedule is being used for triggering automatic recording on a video source, the Active flag will be true when the schedule-based recording is supposed to record.

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

* **SpecialDay**

Indicates that the current time is within the boundaries of its special days schedules’ time periods. For example, if this schedule is being used for recording at a lower frame rate on a video source during special days, the SpecialDay flag will be true. If special days are not supported by the device, this field may be omitted and interpreted as false by the client.

### GetScheduleState command

This operation requests the ScheduleState for the schedule instance specified by the given token.

Table 16 GetScheduleState command

|  |  |  |
| --- | --- | --- |
| GetScheduleState | | Access Class: READ\_SYSTEM\_SENSITIVE |
| **Message name** | **Description** | |
| GetScheduleStateRequest | *This message contains:*   * *"Token": Token of schedule instance to get ScheduleState.*   pt:ReferenceToken **Token [1][1]** | |
| GetScheduleStateResponse | *This message contains:*   * *"ScheduleState": ScheduleState item.*   tac:ScheduleState **ScheduleState [1][1]** | |
| **Fault codes** | **Description** | |
| env:Sender  ter:InvalidArgVal  ter:NotFound | *Schedule token is not found* | |

# Notification topics

This section defines notification topics specific to the schedule service.

## Event overview (informative)

The schedule service specifies events when schedules or its special days are active and when schedules or special days are changed.

The main topic for status changes are:

* tns1:Schedule/State/Active

The main topics for configuration change notifications are:

* tns1:Configuration/Schedule/Changed
* tns1:Configuration/Schedule/Removed
* tns1:Configuration/SpecialDays/Changed
* tns1:Configuration/SpecialDays/Removed

## Status changes

### Schedule

If the StateReportingSupported capability is set to true then the service/device shall be capable of generating the following event whenever a schedule or its special days becomes active or inactive based on the device time. It’s a property event that indicates if the schedule is active or not.

Topic: tns1:Schedule/State/Active

<tt:MessageDescriptionIsProperty="true">

<tt:Source>

<tt:SimpleItemDescription Name="ScheduleToken"

Type="tt:ReferenceToken"/>

<tt:SimpleItemDescription Name="Name"

Type="xs:string"/>

</tt:Source>

<tt:Data>

<tt:SimpleItemDescription Name="Active"

Type="xs:boolean"/>

<tt:SimpleItemDescription Name="SpecialDay"

Type="xs:boolean"/>

</tt:Data>

</tt:MessageDescription>

## Configuration changes

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

### Schedule

Whenever the configuration data for a schedule is changed (including SpecialDaysSchedule) or if a schedule is added, the device shall provide the following event:

Topic: tns1:Configuration/Schedule/Changed

<tt:MessageDescriptionIsProperty=”false”>

<tt:Source>

<tt:SimpleItemDescription Name=”ScheduleToken”

Type=”pt:ReferenceToken”/>

</tt:Source>

</tt: MessageDescription>

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

Topic: tns1:Configuration/Schedule/Removed

<tt:MessageDescriptionIsProperty=”false”>

<tt:Source>

<tt:SimpleItemDescription Name=”ScheduleToken”

Type=”pt:ReferenceToken”/>

</tt:Source>

</tt:MessageDescription>

### Special Days

Whenever the configuration data for a SpecialDays item is changed or added, the device shall provide the following event:

Topic: tns1:Configuration/SpecialDays/Changed

<tt:MessageDescriptionIsProperty=”false”>

<tt:Source>

<tt:SimpleItemDescription Name=”SpecialDaysToken”

Type=”pt:ReferenceToken”/>

</tt:Source>

</tt:MessageDescription>

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

Topic: tns1:Configuration/SpecialDays/Removed

<tt:MessageDescriptionIsProperty=”false”>

<tt:Source>

<tt:SimpleItemDescription Name=”SpecialDaysToken”

Type=”pt:ReferenceToken”/>

</tt:Source>

</tt:MessageDescription>

1. - Examples

Note that all iCalendar structures in the examples below are not complete. Only the parts that are explanatory are included.

## Access 24\*7 for admin staff

In the following example, the values of SpecialDaysSupported and ExtendedRecurrenceSupported are irrelevant.

<Schedule token=”402”>

<Name>Access 24\*7</Name>

<Standard>

BEGIN:VCALENDAR

BEGIN:VEVENT

SUMMARY:Access 24\*7

DTSTART:19700101T000000

DTEND:19700102T000000

RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR,SA,SU

END:VEVENT

END:VCALENDAR

</Standard>

</Schedule>

## Access on Monday and Wednesday from 6 AM to 8 PM for cleaning staff

In the following example, the values of SpecialDaysSupported and ExtendedRecurrenceSupported are irrelevant.

<Schedule token=”402”>

<Name>Access on Monday and Wednesday from 6 AM to 8 PM</Name>

<Standard>

BEGIN:VCALENDAR

BEGIN:VEVENT

SUMMARY:Access on Monday and Wednesday from 6 AM to 8 PM

DTSTART:19700105T060000

DTEND:19700105T200000

RRULE:FREQ=WEEKLY;BYDAY=MO,WE

END:VEVENT

END:VCALENDAR

</Standard>

</Schedule>

## Access from Friday 6 PM to Monday 7 AM for maintenance staff

This example can be realized in two different ways depending on the capabilities of the schedule service exposed by the device.

**Case A**

The following example is valid for ExtendedRecurrenceSupported=true. The value of SpecialDaysSupported is irrelevant.

Note that if ExtendedRecurrenceSupported=false for the example below, the device will throw a fault because the year in DTSTART is not ‘1970’.

<Schedule token=”402”>

<Name>Access from Friday 6 PM to Monday 7 AM for maintenance staff</Name>

<Standard>

BEGIN:VCALENDAR

BEGIN:VEVENT

SUMMARY:Access from Friday 6 PM to Monday 7 AM for maintenance staff

DTSTART:20140523T180000

DTEND:20140526T070000

RRULE:FREQ=WEEKLY;BYDAY=FR

END:VEVENT

END:VCALENDAR

</Standard>

</Schedule>

**Case B**

The following example is valid for SpecialDaysSupported=false. The value of ExtendedRecurrenceSupported is irrelevant.

<Schedule token=”402”>

<Name>Access from Friday 6 PM to Monday 7 AM for maintenance staff</Name>

<Standard>

BEGIN:VCALENDAR

BEGIN:VEVENT

SUMMARY:Access from Friday 6 PM to Friday midnight

DTSTART:19700102T180000

DTEND:19700103T000000

RRULE:FREQ=WEEKLY;BYDAY=FR

END:VEVENT

BEGIN:VEVENT

SUMMARY:Access for all Saturdays and all Sundays

DTSTART:19700103T000000

DTEND:19700104T000000

RRULE:FREQ=WEEKLY;BYDAY=SA,SU

END:VEVENT

BEGIN:VEVENT

SUMMARY:Access from Sunday midnight to Monday 7 AM

DTSTART:19700105T000000

DTEND:19700105T070000

RRULE:FREQ=WEEKLY;BYDAY=MO

END:VEVENT

END:VCALENDAR

</Standard>

</Schedule>

## Access on Weekdays from 8 AM to 5 PM for employees

In the following example, the values of SpecialDaysSupported and ExtendedRecurrenceSupported are irrelevant.

<Schedule token=”402”>

<Name>Access on weekdays from 8 AM to 5 PM for employees</Name>

<Standard>

BEGIN:VCALENDAR

BEGIN:VEVENT

SUMMARY:Access on weekdays from 8 AM to 5 PM for employees

DTSTART:19700101T080000

DTEND:19700102T170000

RRULE:FREQ=WEEKLY;BYDAY=MO,TU,WE,TH,FR

END:VEVENT

END:VCALENDAR

</Standard>

</Schedule>

## Access from January 15, 2014, to January 14, 2015, from 9 AM to 6 PM

This example can be realized in two different ways depending on the capabilities of the schedule service exposed by the device.

**Case A**

The following example is valid for ExtendedRecurrenceSupported=true. The value of SpecialDaysSupported is irrelevant.

<Schedule token=”402”>

<Name>Access from Jan 15, 2014, to Jan 14, 2015, from 9 AM to 6 PM</Name>

<Standard>

BEGIN:VCALENDAR

BEGIN:VEVENT

SUMMARY:Access from Jan 15, 2014, to Jan 14, 2015, from 9 AM to 6 PM

DTSTART:20140115T090000

DTEND:20150114T180000

RRULE:FREQ=DAILY

END:VEVENT

END:VCALENDAR

</Standard>

</Schedule>

**Case B**

The following example is valid for SpecialDaysSupported=false and ExtendedRecurrenceSupported=false.

**Note:** This scenario cannot be directly implemented in the device using only the schedule if ExtendedRecurrenceSupported capability is not supported. PACS uses start and end dates in access policies which refer schedules. Alternately, the client or the host would need to break the schedule into weekly schedule (whatever recurrence the device supports) and download those schedules to the device whenever the schedule changes (January 15, 2014 and January 14, 2015).

## Special Days example 1

An employee has access defined as per the schedule, refer example A.4. However the site admin wants to restrict access to employees on Christmas day (Thursday) as it is a national holiday. The site admin has a special days entity called ‘National Holidays’ with December 25 added to it. The site admin modifies the schedule to add a special days schedule for the special days entity ‘National Holidays’ to override the access and disable employee from accessing the facility.

The following example is valid for SpecialDaysSupported=true. The value of ExtendedRecurrenceSupported is irrelevant.

<SpecialDayGroup token=”8765”>

<Name>National Holidays</Name>

<Days>

BEGIN:VCALENDAR

PRODID:VERSION:2.0

BEGIN:VEVENT

SUMMARY:Christmas day

DTSTART:20141225T000000

DTEND:20141226T000000

UID:Holiday@ONVIF.com

END:VEVENT

END:VCALENDAR

</Days>

<SpecialDayGroup>

<Schedule token=”402”>

<Name>Access on Monday and Wednesday from 8 AM to 5 PM. No access on Christmas day</Name>

<Standard>

BEGIN:VCALENDAR

PRODID:VERSION:2.0

BEGIN:VEVENT

SUMMARY:Access on Monday and Wednesday from 8 AM to 5 PM

DTSTART:19700105T080000

DTEND:19700105T170000

RRULE:FREQ=WEEKLY;BYDAY=MO,WE

UID:Event@ONVIF.com

END:VEVENT

END:VCALENDAR

</Standard>

<SpecialDaysSchedule>

<SpecialDaysToken>8765</SpecialDaysToken>

<!-- No time period indicates no access on referred special days -->

</SpecialDaysSchedule>

</Schedule>

## Special Days example 2

A janitor has access defined as per schedule refer example A.4, however the site admin wants to allow access to cleaning staff on Christmas day (Thursday) as it is a national holiday. The site admin has a special days entity called ‘National Holidays’ with December 25 added to it. The site admin modifies the schedule to add a special day schedule for the special days entity ‘National Holidays’ to override the access and allow janitor to access the facility.

The following example is valid for SpecialDaysSupported=true. The value of ExtendedRecurrenceSupported is irrelevant.

<SpecialDayGroup token=”8765”>

<Name>National Holidays</Name>

<Days>

BEGIN:VCALENDAR

PRODID:VERSION:2.0

BEGIN:VEVENT

SUMMARY:Christmas day

DTSTART:20141225T000000

DTEND:20141226T000000

UID:Holiday@ONVIF.com

END:VEVENT

END:VCALENDAR

</Days>

<SpecialDayGroup>

<Schedule token=”402”>

<Name>Access on Monday and Wednesday from 8 AM to 5 PM. No access on Christmas day</Name>

<Standard>

BEGIN:VCALENDAR

PRODID:VERSION:2.0

BEGIN:VEVENT

SUMMARY:Access on Monday and Wednesday from 8 AM to 5 PM

DTSTART:19700105T080000

DTEND:19700105T170000

RRULE:FREQ=WEEKLY;BYDAY=MO,WE

UID:Event@ONVIF.com

END:VEVENT

END:VCALENDAR

</Standard>

<SpecialDaysSchedule>

<SpecialDaysToken>8765</SpecialDaysToken>

<StartTime>00:00:00</StartTime>

</SpecialDaysSchedule>

</Schedule>

Annex B. Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Vers.** | **Date** | **Author** | **Description** |
| 0.1 | May-2014 | Suresh R | Draft version 0.1 |
| 0.2 | June-2014 | Suresh R | Updates after the Stuttgart F2F meeting |
| 0.3 | July-2014 | Suresh R | Updates implemented, review from Patrik Björling Rygert and Mohane Caliaperoumal |
| 0.4 | Aug-2014 | Suresh R | Updates implemented on the daily, monthly and yearly recurrence type.  Removed the session 3.1 ‘Retrieving system configuration’  Added an new image for schedule object model. |
| 0.5 | Aug-2014 | Patrik Björling Rygert | Overview section rewritten. All command tables gone through and updated. Missing commands added. Explanatory pictures added. Format fixes, and automatic table numbering fixed. |
| 0.6 | Sep 2014 | Suresh R | Minor editorial changes and also accepted all the changes. |
| 0.7 | Sep 2014 | Suresh R | Changes incorporated based on ticket #1485 |
| 0.8 | Sep 2014 | Suresh R | Updates incorporated based on telco meeting |
| 0.9 | Sep 2014 | Patrik Björling Rygert,  Mohane Caliaperoumal | Added faults, fixed spelling errors, modified SpecialDaysSchedule image |
| 0.10 | Nov 2014 | Suresh R | Updates incorporated to include separate API for create, modify and also the example of iCal |
| 0.11 | Nov 2014 | Suresh R | Updates based on Fort Collins F2F, closure of ticket #1501 |
| 0.12 | Nov 2014 | Patrik Björling Rygert | Refining and rephrasing document. Rewrites due to updated and new capabilities. |
| 0.13 | Dec 2014 | Uvaraj Thangarajan  Suresh R | Schedule service examples updated with iCalendar data.  Updated review comments from Robert |
| 0.14 | Jan 2015 | Suresh R | Convert to TC format and updated review comments from Patrik |
| 0.15 | Jan 2015 | Patrik Björling Rygert | Updates after discussions in telco. |
| 0.16 | Jan 2015 | Mohane Caliaperoumal Patrik Björling Rygert | Updates after discussions in telco. |
| 1.0 | Jan-2015 | Hans Busch | Minor editorial changes  Add reference to RFC 2445 |
| 1.01 | Mar 2015 | Patrik Björling Rygert | Added ReferenceNotFound fault. |
| 1.02 | May 2015 | Uaraj Thangarajan | Updated based on telco (12 may, 2015), closure of ticket [#1657](https://wush.net/trac/onvif/ticket/1657) |
| 1.03 | May 2015 | Suresh R | Updated based on telco, section 4.4.11, |
| 1.0.4 | Jun 2015 | PACS WG | Updates on section 4.4.1.1, for the ticket #941 (external)  Updates on section 4.2.1.4, for the ticket #955 (external)  Minor editorial changes |