INTERNATIONAL TELECOMMUNICATION UNION

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

H.282

(05/99)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Related systems aspects

---

Remote device control protocol for multimedia applications

ITU-T Recommendation H.282

(Previously CCITT Recommendation)

ITU-T H-SERIES RECOMMENDATIONS

AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Characteristics of transmission channels used for other than telephone purposes	H.10–H.19
Use of telephone-type circuits for voice-frequency telegraphy	H.20–H.29
Telephone circuits or cables used for various types of telegraph transmission or simultaneous transmission	H.30–H.39
Telephone-type circuits used for facsimile telegraphy	H.40–H.49
Characteristics of data signals	H.50–H.99
CHARACTERISTICS OF VISUAL TELEPHONE SYSTEMS	H.100–H.199
INFRASTRUCTURE OF AUDIOVISUAL SERVICES	
General	H.200–H.219
Transmission multiplexing and synchronization	H.220–H.229
Systems aspects	H.230–H.239
Communication procedures	H.240–H.259
Coding of moving video	H.260–H.279
Related systems aspects	**H.280–H.299**
Systems and terminal equipment for audiovisual services	H.300–H.399
Supplementary services for multimedia	H.450–H.499

For further details, please refer to ITU-T List of Recommendations.

ITU-T RECOMMENDATION H.282

REMOTE DEVICE CONTROL PROTOCOL FOR MULTIMEDIA APPLICATIONS

Summary

This Recommendation describes the core services and protocol for performing configuration and control of a remote device during a multimedia conference. A number of standard devices are defined allowing a node to model a peripheral device such as for example a camera, microphone, VCR or slide projector. This Recommendation also allows the modelling of non-standard devices. This model is presented to the other nodes in the conference to allow these nodes to perform remote control. This Recommendation does not specify the use of a particular transport protocol.

Recommendation T.136 describes its application using T.120 as the transport protocol.

Recommendation H.283 describes its application using the RDC logical channel transport established using H.245.

Future Recommendations may describe its application using other protocols.

Source

ITU-T Recommendation H.282 was prepared by ITU-T Study Group 16 (1997-2000) and was approved under the WTSC Resolution No. 1 procedure on the 27th of May 1999.

FOREWORD

ITU (International Telecommunication Union) is the United Nations Specialized Agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of the ITU. The ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Conference (WTSC), which meets every four years, establishes the topics for study by the ITU-T Study Groups which, in their turn, produce Recommendations on these topics.

The approval of Recommendations by the Members of the ITU-T is covered by the procedure laid down in WTSC Resolution No. 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation the term *recognized operating agency (ROA)* includes any individual, company, corporation or governmental organization that operates a public correspondence service. The terms *Administration, ROA* and *public correspondence* are defined in the *Constitution of the ITU (Geneva, 1992).*

INTELLECTUAL PROPERTY RIGHTS

The ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. The ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, the ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

Recommendation H.282

REMOTE DEVICE CONTROL PROTOCOL FOR MULTIMEDIA APPLICATIONS

(Geneva, 1999)

1 Scope

This Recommendation defines a set of services and a protocol which may be used to perform remote device control in a multimedia conference.

Recommendation T.136 describes how to utilize H.282 using T.120 as the lower layer protocol. Recommendation H.283 describes its application using the RDC logical channel transport established using H.245. Future Recommendations may describe how to utilize H.282 using other lower layer protocols.

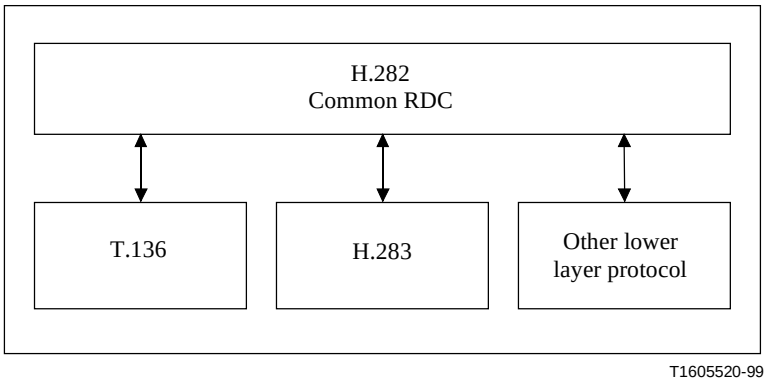

Figure 1/H.282

2 Normative References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published.

– ITU-T Recommendation H.281 (1994), *A far end camera control protocol for video conferences using H.224*.

– ITU-T Recommendation T.120 (1996), *Data protocols for multimedia conferencing*.

– ITU-T Recommendation T.136 (1999), *Remote device control application protocol*.

– ITU-T Recommendation X.680 (1997), *Information technology – Abstract Syntax Notation One (ASN.1)*: *Specification of basic notation*.

– ITU-T Recommendation X.691 (1997), *Information technology – ASN.1 encoding rules – Specification of Packed Encoding Rules (PER)*.

– ISO/IEC 10646-1 (1993), Information technology – Universal Multiple-Octet Coded Character Set (UCS) – Part 1: Architecture and Basic Multilingual Plane.

3 Definitions

This Recommendation defines the following terms:

3.1 Node ID: A unique identifier that allows a node to be distinguished from other nodes in the conference.

3.2 RDC Service Provider: An entity that implements the services described in this Recommendation.

3.3 User Application: An entity that makes use of the services offered by the RDC Service Provider in order to perform remote device control.

3.4 Controllable Device: A device that has the capability to be controlled remotely in a conference environment.

3.5 Device: An entity which provides a specific function, or set of functions, to the conference. Some devices such as cameras and microphones are capable of generating a source to the conference.

3.6 Node: A distinct entity that is participating in a point-to-point or multipoint conference.

3.7 Source: The origin of a real time media stream.

3.8 Lower Layer Protocol: A protocol that is used to transport the RDC PDUs.

3.9 Low Latency Data Channel: A data channel that allows a PDU to be sent with low latency. This may be achieved by sending the PDU as unreliable data which means its delivery cannot be guaranteed.

3.10 Unicode: Text string format with Multilingual capabilities as defined in ISO/IEC 10646-1.

4 Abbreviations

This Recommendation uses the following abbreviations:

PDU Protocol Data Unit

RDC Remote Device Control

5 Conventions

The primitive parameters of the abstract services defined in this Recommendation use the following key:

M: parameter is mandatory;

C: parameter is conditional;

O: parameter is optional;

Blank: parameter is absent.

(=): value of the parameter is identical to the value of the corresponding parameter of the preceding primitive, where preceding is defined relative to the order: request, indication, response, confirm.

(=RQ): value of the parameter is identical to the value of the corresponding parameter in a preceding primitive, where RQ = request, IN = indication, RS = response, and CF = confirm.

Primitives are categorized in up to four types: Request, Indication, Response, and Confirm. Some primitives support all of these types, while others do not. These four types are defined as follows:

Request primitive: Those that are sourced from a User Application to initiate a certain action.

Indication primitive: Those that are sourced from an RDC Service Provider either as a result of a Request primitive, or as a result of an RDC Service Provider initiated action.

Response primitive: Those that are sourced from a User Application in response to an Indication primitive which is defined to require a response.

Confirm primitive: Those that are sourced from an RDC Service Provider as a result of a Response primitive.

6 Overview of Remote Device Control

RDC provides modelling and classification of devices for the purpose of allowing remote activation and control within a conference environment.

A number of standard device classes and a library of standard device attributes are defined by this Recommendation (see Annex A). These aim to address mandatory or commonly required functionality for use in support of multipoint multimedia communication and in particular conferencing.

A device is normally a physical entity but may also be a software process and will typically reside at a Terminal node. In both cases the RDC control model requires that the host node is the default owner of a given device and has responsibility for servicing remote device requests. It is not necessary for a host node to be co-located with the device it is hosting. A device can be a source or receiver of a real time stream or a device may be both a source and a receiver such as in the case of a video cassette recorder.

Remote device control provides the following facilities to the conference:

– declaration of devices and their associated attributes to enable a conference participant to determine a profile of a device attached to a remote node;
– a mechanism to allow a participant to request exclusive access to a remote device in order to perform device control or configuration;
– mechanisms to allow a participant to control, configure, receive event notification and obtain the status of a remote device;
– the definition of a standard device class which allows a participant to select a set of remote devices which have their outputs combined together to form a single output stream.

7 RDC Infrastructure

7.1 Capabilities

At each node, each device is assigned a device class identifier, according to the type of device, and a unique numeric identifier to distinguish it from other devices with the same class identifier. The device class and identifier taken together with the unique node ID of the node that owns the device provides a unique handle for each device within the conference.

7.1.1 Device List

A node may advertise a device that it will allow to be remotely accessed and controlled via a device profile. A device list is made up of the device profiles of the same device class. Hence a device list is constructed for each device class that is supported by the node. Each device list is provided to the

lower layer protocol which it distributes unchanged to all other nodes in the conference using its capability exchange mechanism. A standard capability ID is used to identify the device class of the devices profiles contained in the list (see 7.1.3). A non-standard capability ID may be used to identify a list of non-standard devices.

A device profile only provides a limited amount of device information. A remote node must still interrogate the local node using the RDC-Device-Attributes request primitive to obtain the full device details before control or configuration of that device commences. A device does not require the existence of peer devices to be used in a conference.

Each device profile in the list contains the following information:

Table 1/H.282 – Content of Device Profile Entry

Parameter	Description	Type/Value
Device Identifier	A locally unique number used to distinguish multiple devices of the same type at a node.	INTEGER (0 .. 127)
Audio Source Flag	Flag indicating whether the device can source audio.	BOOLEAN
Audio Sink Flag	Flag indicating whether the device can sink audio.	BOOLEAN
Video Source Flag	Flag indicating whether the device can source video.	BOOLEAN
Video Sink Flag	Flag indicating whether the device can sink video.	BOOLEAN
Remote Control Flag	This flag is set to TRUE if the device has any attributes which may be used for device control. If this flag is FALSE then the device is only available to be selected as a stream source.	BOOLEAN
Device Lock Flag	This flag is set to TRUE if the device can be locked for exclusive use by a single node.	BOOLEAN
Instance Number	The instance number for this Device Profile Entry. This is an 8-bit number which is incremented modulo 2^8 each time the set of attributes for this device change.	INTEGER (0 .. 255)
Device Name	This is an optional Unicode text string, limited to a maximum length of 32 characters, which contains the name of the device.	TextString

If a device is declared as having attributes, these must be obtained by a remote node before control or configuration can commence.

A device that is declared as having an output source is available for source selection.

It is permissible for a device to be declared as controllable but not be available for source selection or for a device to be available for source selection but is not controllable.

If any of the attributes declared by a device change, a node must re-advertise that device profile with a new instance number via the capability mechanisms of the lower layer protocol. As the device profile is distributed, peer nodes will see the device profile instance number change and hence determine that a new set of attributes is now available. A node will then be required to interrogate a host node to obtain the new device details before control or configuration of that device commences. A device profile may not be removed or re-advertised with a new instance number during the period that the device is locked by a remote node.

7.1.2 Stream List

A node may advertise the video and audio streams to which a device may be connected to provide an output to that stream. A stream profile list containing one or more stream profiles is provided to the lower layer protocol which it distributes unchanged to all other nodes in the conference using its capability exchange mechanism. A standard capability ID is used to identify the stream list (see 7.1.3). The information in this stream profile may then be used by remote nodes to perform source selection.

Each stream profile in the list contains the following information:

Table 2/H.282 – Content of Stream Profile Entry

Parameter	Description	Type/Value
Stream Identifier	A locally unique number used to distinguish multiple streams output by this node.	INTEGER (0 .. 65535)
Video Stream	Flag indicating whether the stream is a video stream. If this flag is set to FALSE then the stream is an audio stream.	BOOLEAN
Source Change Event Notification	Flag indicating whether the node supports the sending of source change events for this stream.	BOOLEAN
Stream Name	This is an optional Unicode text string, limited to a maximum length of 32 characters, which contains the name of the stream.	TextString

7.1.3 Standard Capability Identifiers

This table defines the standard capability identifiers.

Table 3/H.282 – Standard Capability Identifiers

Capability Name	Capability Description	ID	List
Camera	A list of cameras available for remote control	1	Device list
Microphone	A list of microphones available for remote control	2	Device list
Stream Player Recorder	A list of stream player recorders available for remote control	3	Device list
Slide Projector	A list of slide projectors available for remote control	4	Device list
Light Source	A list of light sources available for remote control	5	Device list
Source Combiner	A list of source combiners available for remote control	6	Device list
Stream	A list of output streams	7	Stream list

7.2 Conducted Mode Behaviour

The RDC Service Provider has two modes of operation one for non-conducted mode and one for conducted mode, as described in this subclause.

In conducted mode, a service provider must obtain permission from the conductor node before making new use of certain functions. The details of this mode of operation are described in Recommendation T.136 when using T.120 as the lower layer protocol.

8 RDC Service Definition

8.1 Device Attributes

An RDC application wishing to control, configure, obtain the status or receive events from a specific device at another node will first be required to obtain the device attributes using the RDC-Device-Attributes request primitive. This primitive is issued by the RDC application and is directed to the node at which the device is located. If the request is successful the RDC-Device-Attributes confirm primitive will contain the entire list of device attributes. Certain attributes will be accompanied with information describing the range of values supported by the parameters used by that attribute. See Annex A for more details on device classes and their attributes.

8.1.1 RDC-Device-Attributes

Device attributes are divided into two main types: control attributes and event attributes. A control attribute may allow a peer application to only set, only read or both set or read a device parameter. Control attributes which allow the value of a device attribute to be set may be used to control a remote device using the Device-Control primitive (see 8.2.4) Control attributes which allow a device parameter to be read may be used to obtain the status of a remote device using the Device-Status primitive (see 8.4.1). Event attributes may be used to request notification of events from a remote device (see 8.5.1).

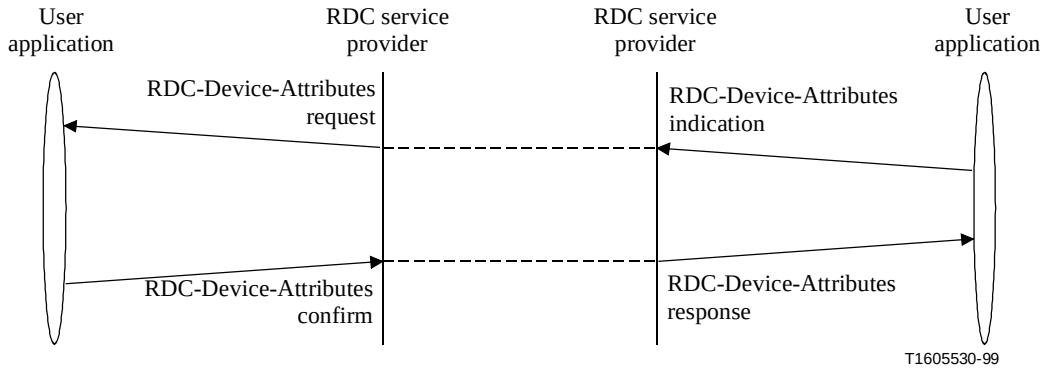

Figure 2/H.282 – RDC-Device-Attributes – Sequence of primitives

Table 4/H.282 – RDC-Device-Attributes

Content	Request	Indication	Response	Confirm
Request Handle	M	M(=)	M(=)	M(=)
Requesting Node		M	M(=)	
Device Owner	M			
Device Class	M	M(=)		
Device Identifier	M	M(=)		
Instance Number			M	M(=)
List of device attributes			M	M(=)
Result			M	M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Unique node ID of the requesting node.

Device Owner: Unique node ID of the node at which the device is located.

Device Class: This parameter identifies the type of device for which a list of attributes is required. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class at the source owner node. The Device Identifier is extracted from the appropriate Device Profile.

Instance Number: This is the current instance number for this set of device attributes which is advertised in the Device Profile.

List of device attributes: This is a list of device attributes. The format of these attributes is defined in Annex A.

Result: Indicates whether the attribute collection from the selected device was successful. It contains one of the following results: successful, request denied, unknown device.

8.2 Device Control

The Device Lock flag in a device profile indicates if the device may be locked prior to performing remote control. Device locking is an optional mechanism that allows a device to be granted exclusive access to one node at a time. In this case, a User Application must lock a remote device using the RDC-Device-Lock request primitive if exclusive access is required when performing remote control. Similarly, if a node requires exclusive access to a local device which it is hosting in order to perform local control then it must lock this device. In this way, a device will only be controlled by one application at one time.

If the Device Lock Flag is set to false, a node does not support device locking for that device and cannot therefore give exclusive access to a remote node.

A User Application always has ultimate authority over a lock placed on any device that it is hosting. It may choose to terminate a lock on a hosted device if access to the device is required and then inform the remote node that it has lost exclusive access. Alternatively, it may interrogate the remote node to determine if the lock is still required and then terminate the lock if it is no longer required.

A User Application that is hosting a device is free to accept control requests from any node whilst it is not locked. Hence, a User Application may still control a device using an RDC-Device-Control request primitive without locking the device. This may be desirable when low latency is required and the overhead incurred to lock the device prior to control is unacceptable. However, the requesting application will not have exclusive access to the hosting application which will ignore the control request if it has already been locked by another RDC application. The hosting application may also receive conflicting control requests from other RDC applications at the same time. This may lead to control requests being ignored although it is a local matter as to how the hosting application arbitrates between such requests.

If a User Application, after locking a remote device, then leaves the conference without releasing the lock, then that lock will automatically be released.

A device does not need to be locked in order to obtain status information or to configure a device with an event list for event notification.

8.2.1 RDC-Device-Lock

The RDC-Device-Lock request primitive is issued by the User Application to configure the lock state of a particular device. The request is directed to the node at which the device is located. A device cannot be locked if it has already been locked by another node. A User Application may only request a device be unlocked, if it originally locked that device.

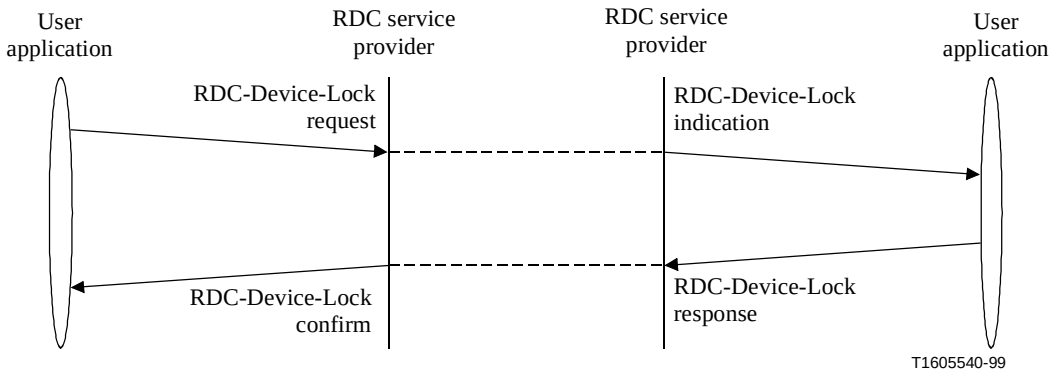

Figure 3/H.282 – RDC-Device-Lock – Sequence of primitives

Table 5/H.282 – RDC-Device-Lock

Content	Request	Indication	Response	Confirm
Request Handle	M	M(=)	M(=)	M(=)
Requesting Node		M	M(=)	
Device Owner	M			
Device Class	M	M(=)		
Device Identifier	M	M(=)		
Lock/Unlock	M	M(=)		
Result			M	M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Node ID of the requesting node.

Device Owner: Unique node ID of the node at which the device is located.

Device Class: This parameter identifies the type of device required to be locked or unlocked. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class at the source owner node. The Device Identifier is extracted from the appropriate Device Profile.

Lock/Unlock: Set to TRUE if the request is for the device to be locked otherwise FALSE in order to unlock the device.

Result: Indicates whether the lock operation on the selected device was successful. It contains one of the following results: successful, request denied, unknown device, device does not support locking, device already locked by another user.

8.2.2 RDC-Device-Lock-Inquire

If a User Application requires a lock to be removed from one of its devices it may first wish to interrogate the remote node that locked the device using the RDC-Device-Lock-Inquire request primitive to determine if the lock is still required. If the remote RDC application indicates that the device lock is no longer required then the device may be unlocked immediately.

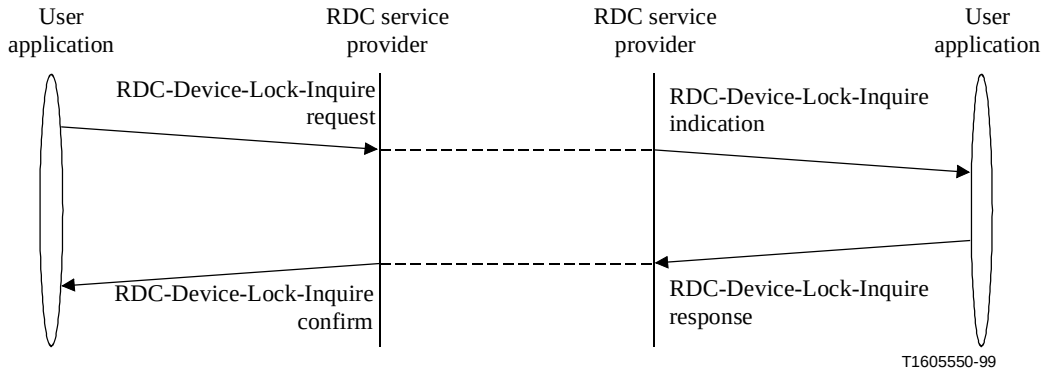

Figure 4/H.282 – RDC-Device-Lock-Inquire – Sequence of primitives

Table 6/H.282 – RDC-Device-Lock-Inquire

Content	Request	Indication	Response	Confirm
Request Handle	M	M(=)	M(=)	M(=)
Requesting Node		M	M(=)	
Device Owner	M			
Device Class	M	M(=)		
Device Identifier	M	M(=)		
Result			M	M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Node ID of the requesting node.

Device Owner: Unique node ID of the node at which the device is located.

Device Class: This parameter identifies the device type of the locked device. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class at the source owner node. The Device Identifier is extracted from the appropriate Device Profile.

Result: Indicates whether the device lock on a particular device is still required. It contains one of the following results: device lock may be released, device lock still required, unknown device.

8.2.3 RDC-Device-Lock-Terminated

A User Application always has ultimate authority over the locks placed on any device owned by it. It may choose to terminate a device lock placed by a remote terminal. If a device lock is terminated without first interrogating the remote node using the RDC-Device-Lock-Inquire request primitive

then the RDC-Device-Lock-Terminated request primitive should be used to inform the remote node that it no longer has exclusive access.

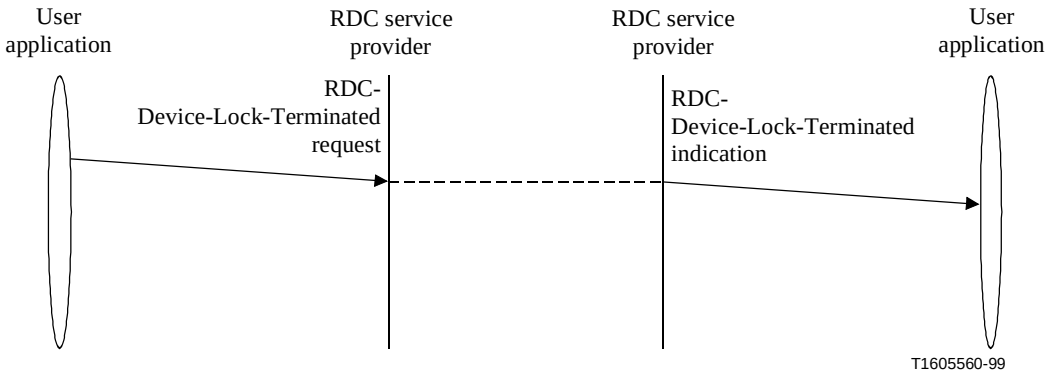

Figure 5/H.282 – RDC-Device-Lock-Terminated – Sequence of primitives

Table 7/H.282 – RDC-Device-Lock-Terminated

Content	Request	Indication
Host Node		M
Lock Owner	M	
Device Class	M	M(=)
Device Identifier	M	M(=)

Host Node: Node ID of the node that is hosting the device on which the lock has been terminated.

Lock Owner: Unique node ID of the node which owned the lock being terminated.

Device Class: This parameter identifies the type of device on which the lock was terminated. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class. The Device Identifier is extracted from the appropriate Device Profile.

8.2.4 RDC-Device-Control

A User Application can control a remote device by setting the value of attributes belonging to the device using the RDC-Device-Control request primitive. The RDC-Device-Control request primitive is issued by the User Application to control a particular remote device that has been advertised by a Device Profile. The request is directed to the node at which the device is located. The control indication will be ignored by the receiving node if it is currently locked by another RDC node. A control indication may also be ignored because the node is currently servicing an indication received from another node although this is a local matter outside of the scope of this Recommendation. If exclusive access is required, the device must be locked prior to performing device control.

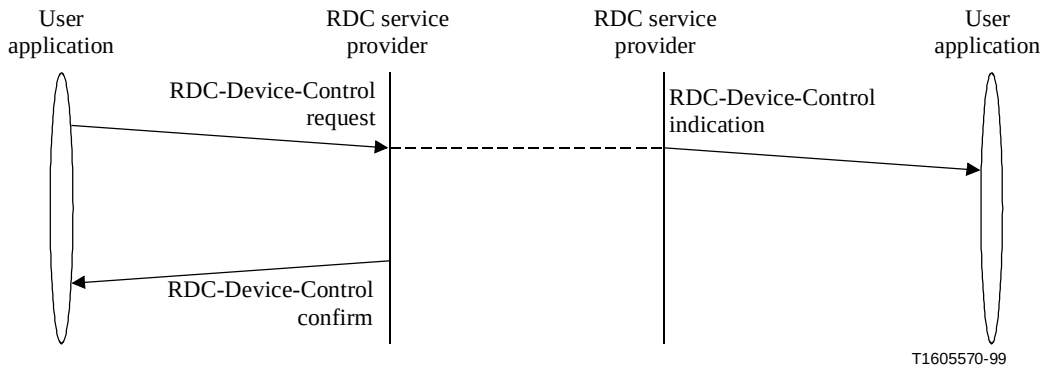

Figure 6/H.282 – RDC-Device-Control – Sequence of primitives

Table 8/H.282 – RDC-Device-Control

Content	Request	Indication	Confirm
Request Handle	M		M(=)
Requesting Node	M	M(=)	
Device Owner	M		
Low Latency Data Channel	O		
Device Class	M	M(=)	
Device Identifier	M	M(=)	
List of device control attributes	M	M(=)	
Result			M

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Node ID of the requesting node.

Device Owner: Unique node ID of the node at which the device is located.

Low Latency Data Channel: This optional parameter allows a User Application to request that the RDC Service Provider sends this request using a low latency data channel if this channel is available. If a low latency data channel is not available then the RDC Service Provider will ignore this parameter. It should be noted however that, even if an RDC Service Provider supports a low latency data channel, such a data path may not exist to all nodes. Hence, data sent initially via a low latency data channel may be transferred to a normal data channel before being delivered to its final destination.

Device Class: This parameter identifies the type of device on which the lock was terminated. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class. The Device Identifier is extracted from the appropriate Device Profile.

List of device control attributes: This is a list of device control attributes. The format of these attributes is defined in Annex A. This list should not contain attributes which would result in opposing actions being executed by the device.

Result: Indicates whether the request was successful. It contains one of the following results: successful, unknown device, device attribute error.

8.3 Source Selection

RDC provides mechanisms to allow a node to advertise a list of sources which it is making available to the conference for selection. Examples of sources include physical devices such as cameras. The Source Selection Service allows a node to request that a particular source be connected to a specific output stream.

8.3.1 RDC-Source-Select

The RDC-Source-Select request primitive is issued by the RDC application to request that a particular source be connected to a specific output stream. The request is directed to the node at which the source is located.

The request will be denied if the device cannot be connected to the stream because of local node restrictions. The request will also be rejected if the device that is currently the source of a stream is locked. A device that is locked cannot be disconnected from a stream.

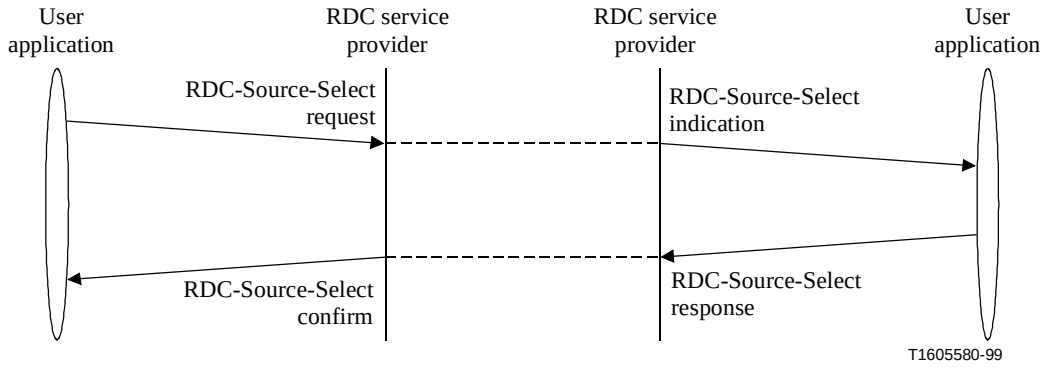

Figure 7/H.282 – RDC-Source-Select – Sequence of primitives

Table 9/H.282 – RDC-Source-Select

Content	Request	Indication	Response	Confirm
Request Handle	M	M(=)	M(=)	M(=)
Requesting Node		M	M(=)	
Source Owner	M			
Device Class	M	M(=)		
Device Identifier	M	M(=)		
Stream Identifier	M	M(=)		
Result			M	M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Node ID of the requesting node.

Source Owner: Node ID of the node at which the source to be selected is located.

Device Class: This parameter identifies the type of device required to act as a source. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class at the source owner node. The Device Identifier is extracted from the appropriate Device Profile.

Stream Identifier: Locally unique number, allowing discrimination between multiple output streams at the source owner node. The Stream Identifier is extracted from the appropriate Stream Profile.

Result: Indicates whether the selected device was connected to the stream. It contains one of the following results: successful, request denied, device not available, invalid stream identifier, current device source is locked, device incompatible with stream type.

8.3.2 RDC-Request-Source-Events

A User Application can request that it is notified of source selection changes occurring on a particular stream using the RDC-Request-Source-Events primitive. A node can declare support for source selection notification in the stream profile. This primitive is issued to the node which is the source of the stream. If a node wishes to cancel notifications it has previously requested then it should send this primitive with the request flag set to false.

The response primitive contains the device class and identifier of the device that is currently the source of the stream. Hence, this primitive may be used to determine the current source device without requesting source selection notifications by setting the request flag to false.

The number of requests for notification that a hosting node can support is a local implementation issue outside the scope of this Recommendation. Therefore, a request for source selection notification by a User Application may be refused because the hosting node has insufficient resources.

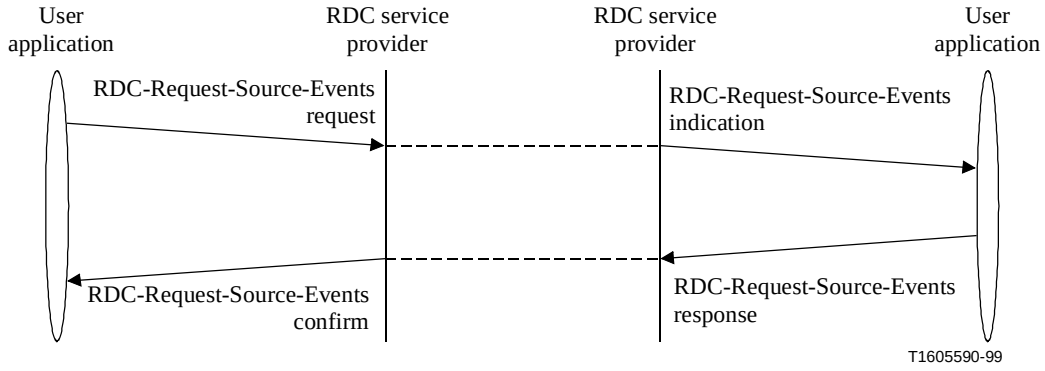

Figure 8/H.282 – RDC-Request-Source-Events – Sequence of primitives

Table 10/H.282 – RDC-Request-Source-Events

Content	Request	Indication	Response	Confirm
Request Handle	M	M(=)	M(=)	M(=)
Requesting Node		M	M(=)	
Source Owner	M			
Stream Identifier	M	M(=)		
Source Selection Notification Request Flag	M	M(=)		
Device Class			M	M(=)
Device Identifier			M	M(=)
Result			M	M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Node ID of the requesting node.

Source Owner: Node ID of the node at which the source is located.

Stream Identifier: Locally unique number, allowing discrimination between different output streams. The Stream Identifier is extracted from the appropriate Stream Profile.

Source Selection Notification Request Flag: This flag indicates if source selection notifications are required. If the flag is set to TRUE if notifications are required otherwise a value of FALSE cancels any previous request.

Device Class: This parameter identifies the type of device which is currently the source of the stream. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class. The Device Identifier is extracted from the appropriate Device Profile.

Result: Indicates whether the notification request was successful. It contains one of the following results: successful, event notifications not supported, invalid stream identifier.

8.3.3 RDC-Source-Change-Event

A stream may have received a request for source change event notification from a remote RDC node using the RDC-Request-Source-Events primitive. When the device sourcing its output to the stream changes, a User Application can inform an interested remote node using the RDC-Source-Change-Event request primitive. The request is directed to the remote node which requested notification. If more than one node has requested notification then the User Application shall issue this primitive for each of these nodes in turn.

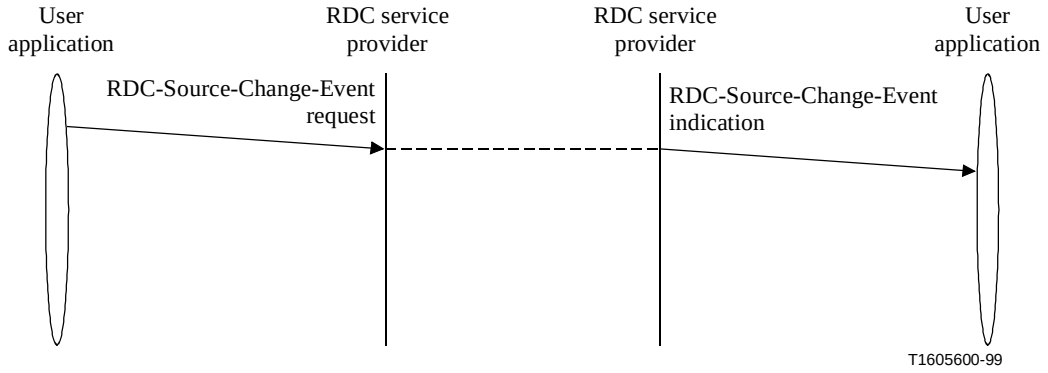

Figure 9/H.282 – RDC-Source-Change-Event – Sequence of primitives

Table 11/H.282 – RDC-Source-Change-Event

Content	Request	Indication
Receiving Node	M	
Reporting Node		M
Stream Identifier	M	M(=)
Device Class	M	M(=)
Device Identifier	M	M(=)

Receiving Node: The node ID of the node that has successfully registered to receive source change events.

Reporting Node: The node ID of the node that is reporting the source change event.

Stream Identifier: Locally unique number, allowing discrimination between different output streams. The Stream Identifier is extracted from the appropriate Stream Profile.

Device Class: This parameter identifies the type of device which is the new source of the stream. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class. The Device Identifier is extracted from the appropriate Device Profile.

8.4 Device Status

8.4.1 RDC-Device-Status

A User Application can interrogate a remote device for its status by reading the value of attributes belonging to the device using the RDC-Device-Status request primitive. The RDC-Device-Status request primitive is issued by the User Application to inquire of the status of a particular remote device that has been advertised in an RDC Device Profile. This primitive contains a list of control attributes for which the status is required. The request is directed to the node at which the device is located. Assuming a successful response, the RDC-Device-Status confirm primitive contains a list of these same control attributes with accompanying parameters indicating their status value.

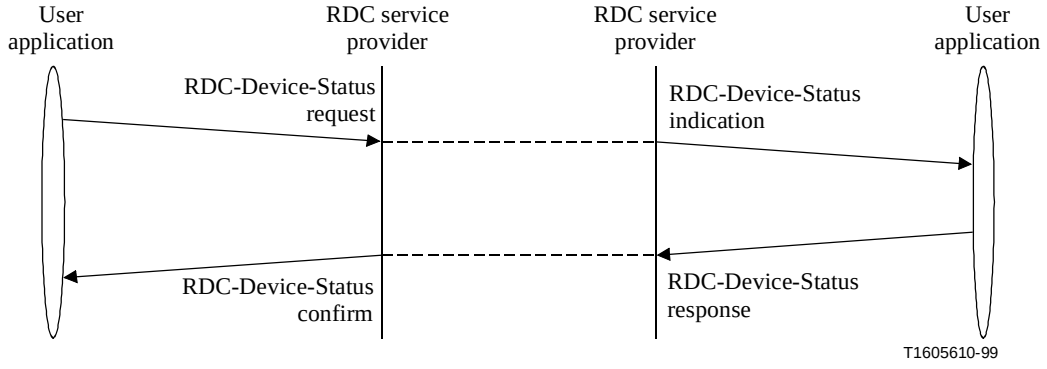

Figure 10/H.282 – RDC-Device-Status – Sequence of primitives

Table 12/H.282 – RDC-Device-Status

Content	Request	Indication	Response	Confirm
Request Handle	M	M(=)	M(=)	M(=)
Requesting Node		M	M(=)	
Device Owner	M			
Device Class	M	M(=)		
Device Identifier	M	M(=)		
List of device control attributes	M	M(=)		
List of device control attributes with associated status values			M	M(=)
Result			M	M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Node ID of the requesting node.

Device Owner: Node ID of the node at which the device is located.

Device Class: This parameter identifies the type of device. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class at the device owner node. The Device Identifier is extracted from the appropriate Device Profile.

List of device control attributes: This is a list of device control attributes for which the status is required. The format of these attributes is defined in Annex A.

List of device control attributes with associated status values: This is the list of device control attributes with associated status values. The format of these attributes is defined in Annex A.

Result: Indicates whether the status collection was successful. It contains one of the following results: successful, request denied, unknown device, device not available, device attribute error.

8.5 Device Events

8.5.1 RDC-Configure-Device-Events

A User Application can request that it is notified of event changes occurring on a remote device. A remote device can declare support for particular event attributes in the attribute list contained in the RDC-Device-Attributes confirm primitive. The User Application can configure a remote device with a list of event attributes for which notification is required using the RDC-Configure-Device-Events primitive.

The RDC-Configure-Device-Events request primitive is issued by the User Application to configure an event list for a particular remote device that has been advertised in an RDC Device Profile. The request is directed to the node at which the device is located. If a node wishes to cancel an event list which it has previously configured then it should send this primitive with an empty list of events.

The number of event lists that a hosting node can support is a local implementation issue outside the scope of this Recommendation. Therefore a request to configure an event list by a User Application may be refused because the hosting node has insufficient resources to store a further list.

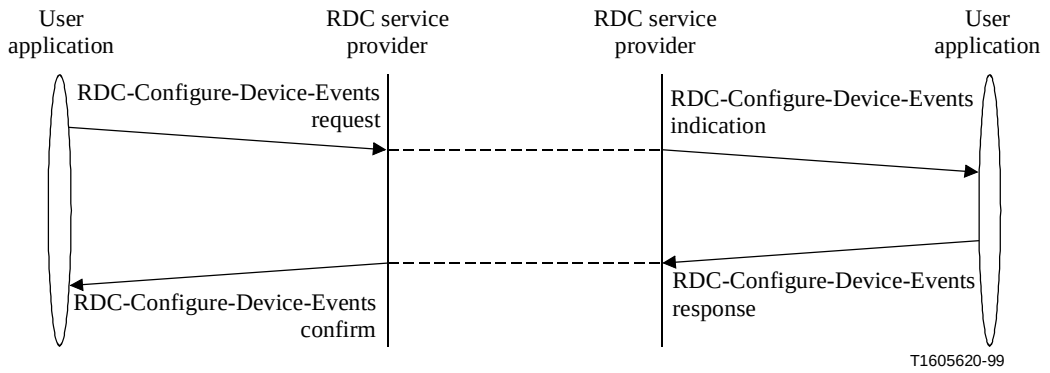

Figure 11/H.282 – RDC-Configure-Device-Events – Sequence of primitives

Table 13/H.282 – RDC-Configure-Device-Events

Content	Request	Indication	Response	Confirm
Request Handle	M	M(=)	M(=)	M(=)
Requesting Node		M	M(=)	
Device Owner	M			
Device Class	M	M(=)		
Device Identifier	M	M(=)		
List of device event attributes	M	M(=)		
Result			M	M(=)

Request Handle: Locally unique number to allow the requesting node to match responses.

Requesting Node: Node ID of the requesting node.

Device Owner: Node ID of the node where the device is located.

Device Class: This parameter identifies the type of device. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class at the device owner node. The Device Identifier is extracted from the appropriate Device Profile.

List of device event attributes: This is a list of device event attributes for which event notification is requested. The format of these attributes is defined in Annex A. This event list replaces any event list that has been previously configured by the sending node.

Result: Indicates whether the request was successful. It contains one of the following results: successful, request denied, unknown device, device not available, device attribute error.

8.5.2 RDC-Notify-Device-Event

A device may have an event list configured by a remote RDC node using the RDC-Configure-Device-Events primitive. When a listed event occurs, a User Application can inform an interested remote node using the RDC-Notify-Device-Event request primitive. The request is directed to the remote node which configured the device event list. If more than one node has requested notification then the User Application shall issue this primitive for each of these nodes in turn.

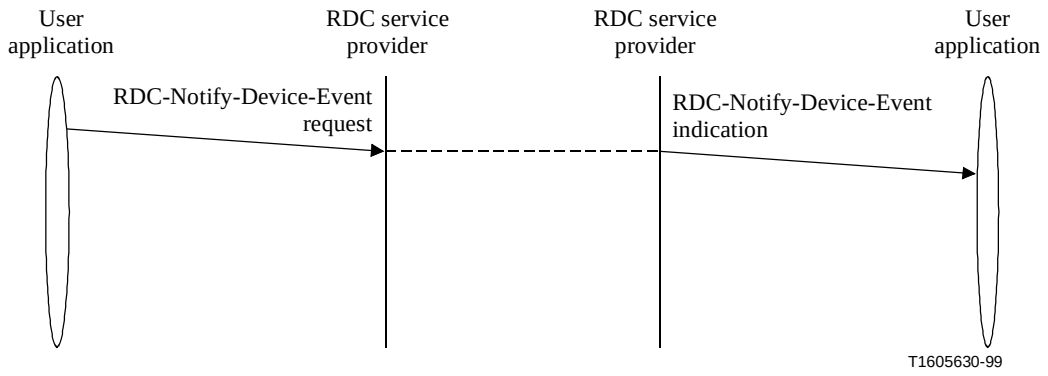

Figure 12/H.282 – RDC-Notify-Device-Event – Sequence of primitives

Table 14/H.282 – RDC-Notify-Device-Event

Content	Request	Indication
Receiving Node	M	
Reporting Node		M
Device Class	M	M(=)
Device Identifier	M	M(=)
List of event attributes	M	M(=)

Receiving Node: The node ID of the node that has successfully registered to receive device events.

Reporting Node: The node ID of the node that is reporting the device event.

Device Class: This parameter identifies the type of device which is sending the event notification. Standardized device classes are defined in Annex A.

Device Identifier: Locally unique number, allowing discrimination between multiple devices of the same class. The Device Identifier is extracted from the appropriate Device Profile.

List of event attributes: This is a list of device event attributes indicating which events are being notified. The format of these attributes is defined in Annex A.

9 RDC Service Management Protocol Definition

9.1 Remote Device Control

9.1.1 Requesting Device Attributes

On receipt of an RDC-Device-Attributes request primitive, an RDC service Provider shall send a DeviceAttributeRequest PDU to the device owner using a medium priority data channel.

Table 15/H.282 – DeviceAttributeRequest PDU

Content	Source	Sink
Request Handle	Request	Indication
Device Class	Request	Indication
Device Identifier	Request	Indication

On receipt of a DeviceAttributeRequest PDU, an RDC Service Provider shall determine whether the device exists at the local node. If this criteria is not met, it shall send a DeviceAttributeResponse PDU to the node that originated the request, indicating the reason for rejection. Otherwise, the RDC Service Provider shall generate an RDC-Device-Attributes indication and issue it to the local User Application.

The User Application is responsible for processing the Device Attribute request and for issuing an RDC-Device-Attributes response containing the result and the requested device attributes. On receipt of an RDC-Device-Attributes response primitive, the RDC Service Provider shall generate a DeviceAttributeResponse PDU and send it to the node that originated the request using a medium priority data channel. A successful response contains the requested device attributes.

Table 16/H.282 – DeviceAttributeResponse PDU

Content	Source	Sink
Request Handle	Response	Confirm
Instance number	Response	Confirm
Device Attribute List	Response	Confirm
Result	Response	Confirm

The format of each attribute is as follows:

Parameter	Description
Device Attribute	This is an attribute for this device class. (See Annex A for more details.)

On receiving the DeviceAttributeResponse PDU from the responding node, the RDC Service Provider should make a copy of the received device attribute list and issue an RDC-Device-Attributes confirm primitive to the User Application. It shall then use this attribute list to verify the contents of an attribute list contained in other primitive requests. The RDC Service Provider shall delete this attribute list if the device profile is no longer advertised by the hosting node. It may also delete this list due to local circumstances (for example the available memory is low and the device has not been remotely accessed for a long period of time). In this case, further request primitives containing device attributes destined for that device will be rejected until another successful DeviceAttributeResponse PDU has been received.

9.1.2 Locking Devices

On receipt of an RDC-Device-Lock request primitive, an RDC Service Provider shall send a DeviceLockRequest PDU to the device owner using a medium priority data channel.

Table 17/H.282 – DeviceLockRequest PDU

Content	Source	Sink
Request Handle	Request	Indication
Device Class	Request	Indication
Device Identifier	Request	Indication
Lock/Unlock	Request	Indication

On receipt of a DeviceLockRequest PDU, an RDC Service Provider shall determine whether the device exists at the local node. If this criteria is not met, it shall send a DeviceLockResponse PDU to the node that originated the request, indicating the reason for rejection. Otherwise, the RDC Service Provider shall generate an RDC-Device-Lock indication and issue it to the local User Application.

The User Application is responsible for processing the Device Lock request and for issuing an RDC-Device-Lock response containing the result. On receipt of an RDC-Device-Lock response primitive, the RDC Service Provider shall generate a DeviceLockResponse PDU and send it to the node that originated the request using a medium priority data channel.

Table 18/H.282 – DeviceLockResponse PDU

Content	Source	Sink
Request Handle	Response	Confirm
Result	Response	Confirm

On receiving the DeviceLockResponse PDU from the responding node, the RDC Service Provider shall issue an RDC-Device-Lock confirm primitive to the local User Application.

9.1.3 Device Lock Inquire

On receipt of an RDC-Device-Lock-Inquire request primitive, an RDC Service Provider shall send a DeviceLockInquireRequest PDU to the device owner using a medium priority data channel.

Table 19/H.282 – DeviceLockInquireRequest PDU

Content	Source	Sink
Request Handle	Request	Indication
Device Class	Request	Indication
Device Identifier	Request	Indication

On receipt of a DeviceLockInquireRequest PDU, an RDC Service Provider shall determine whether the device exists at the local node. If this criteria is not met, it shall send a DeviceLockInquireResponse PDU to the node that originated the request, indicating the reason for rejection. Otherwise, the RDC Service Provider shall generate an RDC-Device-Lock-Inquire indication and issue it to the local User Application.

The User Application is responsible for processing the Device Lock Inquire request and for issuing an RDC-Device-Lock-Inquire response containing the result. On receipt of an RDC-Device-Lock-Inquire response primitive, the RDC Service Provider shall generate a DeviceLockInquireResponse PDU and send it to the node that originated the request using a medium priority data channel. A successful response indicates the node's requirement for the device lock to continue.

Table 20/H.282 – DeviceLockInquireResponse PDU

Content	Source	Sink
Request Handle	Response	Confirm
Result	Response	Confirm

On receipt of the DeviceLockInquireResponse PDU, the RDC Service Provider shall generate an RDC-Device-Lock-Inquire confirm primitive and issue it to the local User Application.

9.1.4 Device Lock Terminated Indication

On receipt of an RDC-Device-Lock-Terminated request primitive, an RDC Service Provider shall send a DeviceLockTerminatedIndication PDU to the node which had originally locked the device. The sending node shall issue the PDU using a high priority data channel.

Table 21/H.282 – DeviceLockTerminatedIndication PDU

Content	Source	Sink
Device Class	Request	Indication
Device Identifier	Request	Indication

On receipt of a DeviceLockTerminatedIndication PDU, an RDC Service Provider shall generate an RDC-Device-Lock-Terminated indication and issue it to the local User Application.

9.1.5 Device Control Request

On receipt of an RDC-Device-Control request primitive, an RDC service Provider shall verify the contents of the list of device control attributes against the attribute information previously received in the DeviceAttributeResponse PDU. If any attribute is incorrect then the request primitive must be rejected by returning an RDC-Device-Control confirm primitive with the appropriate result. If this is not the case it shall send a DeviceControlRequest PDU. If the User Application requested a low latency data channel, then the PDU shall be sent over a low latency data channel, if this is available, otherwise the PDU shall be issued using a high priority data channel.

Table 22/H.282 – DeviceControlRequest PDU

Content	Source	Sink
Device Class	Request	Indication
Device Identifier	Request	Indication
List of Device Control attributes	Request	Indication

On receipt of a DeviceControlRequest PDU, an RDC Service Provider shall determine whether the device exists at the local node and then verify that the list of control attributes are all supported by that device. If these criteria are met, the service provider will generate an RDC-Device-Control indication and issue it to the local User Application. The User Application should then command the specified device to perform the appropriate actions. All actions should be processed by the device in the same order as the reception of RDC-Device-Control indication primitives.

9.2 Remote Device Source Selection

9.2.1 Source Selection

On receipt of an RDC-Source-Select request primitive, an RDC Service Provider shall send a SourceSelectRequest PDU to the Source Owner using a high priority data channel.

On receipt of a SourceSelectRequest PDU, an RDC Service Provider shall determine whether the device selected exists at its local node. If this criteria is not met, it shall send a SourceSelectResponse PDU to the node that originated the request, indicating the reason for rejection using a high priority data channel. Otherwise, the RDC Service Provider shall generate an RDC-Source-Select indication and issue it to the local User Application.

The User Application is responsible for processing the source selection request and for issuing an RDC-Source-Select response to notify the originating node of the outcome. On receipt of an RDC-Source-Select response primitive, the RDC Service Provider shall generate a SourceSelectResponse PDU and send it to the node that originated the request.

Table 23/H.282 – SourceSelectRequest PDU

Content	Source	Sink
Request Handle	Request	Indication
Device Class	Request	Indication
Device Identifier	Request	Indication
Stream Identifier	Request	Indication

Table 24/H.282 – SourceSelectResponse PDU

Content	Source	Sink
Request Handle	Response	Confirm
Result	Response	Confirm

On receipt of a SourceSelectResponse PDU, an RDC Service Provider shall generate an RDC-Source-Select confirm primitive indicating the result included in the PDU and issue it to the local User Application.

9.2.2 Requesting Source Changes

An RDC application can register an interest in source change notifications for a specific stream using the RDC-Request-Source-Events primitive. On receipt of an RDC-Request-Source-Events request primitive, an RDC Service Provider shall send a SourceEventsRequest PDU to the source owner using a medium priority data channel.

Table 25/H.282 – SourceEventsRequest PDU

Content	Source	Sink
Request Handle	Request	Indication
Stream Identifier	Request	Indication
Source Event Notification Flag	Request	Indication

On receipt of a SourceEventsRequest PDU, an RDC Service Provider shall determine whether the selected stream exists at the local node. If this criteria is not met, it shall send a SourceEventsResponse PDU to the node that originated the request, indicating the reason for rejection. Otherwise, the RDC Service Provider shall generate an RDC-Request-Source-Events indication and issue it to the local User Application.

The User Application is responsible for processing the SourceEventsRequest and for issuing an RDC-Request-Source-Events response containing the result. On receipt of an RDC-Request-Source-Events response primitive, the RDC Service Provider shall generate a SourceEventsResponse PDU and send it to the node that originated the request using a medium priority data channel.

Table 26/H.282 – SourceEventsResponse PDU

Content	Source	Sink
Request Handle	Response	Confirm
Device Class	Request	Indication
Device Identifier	Request	Indication
Result	Response	Confirm

On receiving the SourceEventsResponse PDU from the responding node, the RDC Service Provider shall issue an RDC-Request-Source-Events confirm primitive to the local User Application.

9.2.3 Source Change Events

On receipt of an RDC-Source-Change-Event request primitive, an RDC Service Provider shall send a SourceChangeEventIndication PDU to the node which had registered an interest in receiving a notification of source changes using a high priority data channel.

Table 27/H.282 – SourceChangeEventIndication PDU

Content	Source	Sink
Device Class	Request	Indication
Device Identifier	Request	Indication

On receipt of a SourceChangeEventIndication PDU, an RDC Service Provider shall generate an RDC-Source-Change-Event indication and issue it to the local User Application.

9.3 Remote Device Status

9.3.1 Device Status Inquire

On receipt of an RDC-Device-Status-Inquire request primitive, an RDC Service Provider shall verify the contents of the list of device control attributes against the attribute information previously received in the DeviceAttributeResponse PDU. If any attribute is incorrect, the request primitive must be rejected by returning an RDC-Device-Status-Inquire confirm primitive with the appropriate result. If the attribute list is correct, the RDC Service Provider shall send a DeviceStatusInquire Request PDU to the device owner using a medium priority data channel.

Table 28/H.282 – DeviceStatusInquireRequest PDU

Content	Source	Sink
Request Handle	Request	Indication
Device Class	Request	Indication
Device Identifier	Request	Indication
List of device status attributes	Request	Indication

On receipt of a DeviceStatusInquireRequest PDU, an RDC Service Provider shall determine whether the selected device exists at the local node. If this criteria is not met, it shall send a DeviceStatusInquireResponse PDU to the node that originated the request, indicating the reason for rejection. Otherwise, the RDC Service Provider shall generate an RDC-Device-Status indication and issue it to the local User Application.

The User Application is responsible for processing the Device Status Inquire request and for issuing an RDC-Device-Status response containing the result and the requested device status. On receipt of an RDC-Device-Status response primitive, the RDC Service Provider shall generate a DeviceStatusInquireResponse PDU and send it to the node that originated the request using a medium priority data channel. A successful response contains a list of status attributes with accompanying status fields containing the current state of the device.

Table 29/H.282 – DeviceStatusInquireResponse PDU

Content	Source	Sink
Request Handle	Response	Confirm
List of device status attributes	Response	Confirm
Result	Response	Confirm

On receiving the DeviceStatusInquireResponse PDU from the responding node, the RDC Service Provider shall issue an RDC-Device-Status confirm primitive to the local User Application.

9.4 Remote Device Events

9.4.1 Configuring Device Events

An RDC application can register an interest in event notifications from a specific remote device using the RDC-Configure-Device-Events primitive. On receipt of an RDC-Configure-Device-Events request primitive, an RDC Service Provider shall verify the contents of the list of device event attributes against the attribute information previously received in the DeviceAttributeResponse PDU. If any attribute is incorrect, the request primitive must be rejected by returning an RDC-Configure-Device-Events confirm primitive with the appropriate result. If the attribute list is correct, the RDC Service Provider shall send a ConfigureDeviceEventsRequest PDU to the device owner using a medium priority data channel.

Table 30/H.282 – ConfigureDeviceEventsRequest PDU

Content	Source	Sink
Request Handle	Request	Indication
Device Class	Request	Indication
Device Identifier	Request	Indication
List of device event attributes	Request	Indication

On receipt of a ConfigureDeviceEventsRequest PDU, an RDC Service Provider shall determine whether the selected device exists at the local node. If this criteria is not met, it shall send a ConfigureDeviceEventsResponse PDU to the node that originated the request, indicating the reason for rejection. Otherwise, the RDC Service Provider shall generate an RDC-Configure-Device-Events indication and issue it to the local User Application.

The User Application is responsible for processing the Configure Device Events request and for issuing an RDC-Configure-Device-Events response containing the result. On receipt of an RDC-Configure-Device-Events response primitive, the RDC Service Provider shall generate a ConfigureDeviceEventsResponse PDU and send it to the node that originated the request using a medium priority data channel.

Table 31/H.282 – ConfigureDeviceEventsResponse PDU

Content	Source	Sink
Request Handle	Response	Confirm
Result	Response	Confirm

On receiving the ConfigureDeviceEventsResponse PDU from the responding node, the RDC Service Provider shall issue an RDC-Configure-Device-Events confirm primitive to the local User Application.

9.4.2 Notify Device Event

On receipt of an RDC-Notify-Device-Event request primitive, an RDC Service Provider shall send a NotifyDeviceEventIndication PDU to the node which had registered an interest in receiving a notification of that device event using a high priority data channel.

Table 32/H.282 – NotifyDeviceEventIndication PDU

Content	Source	Sink
Device Class	Request	Indication
Device Identifier	Request	Indication
List of event attributes	Request	Indication

On receipt of a NotifyDeviceEventIndication PDU, an RDC Service Provider shall generate an RDC-Notify-Device-Event indication and issue it to the local User Application.

10 RDC PDU Definitions

The structure of RDC PDUs is specified as follows using the notation ASN.1 of Recommendation X.680. All RDC PDUs shall be encoded for transmission by applying the Packed Encoding Rules of Recommendation X.691 using the Basic Aligned variant. The bit string generated by the ASN.1 encoding is placed in the OCTET STRING used by the lower layer protocol in the order such that for each octet, the leading bit is placed in the most significant bit position and the trailing bit is placed in the least significant bit position.

```
RDC-PROTOCOL { itu-t(0) recommendation(0) h 282 version(0) 1}


DEFINITIONS AUTOMATIC TAGS::=
BEGIN


-- Export all symbols
-- ================================================================================
-- Part 1:  Message Components
-- ================================================================================

H221NonStandardIdentifier ::= OCTET STRING (SIZE (4..255))
                                                -- First four octets shall be country code and
                                                -- Manufacturer code, assigned as specified in
                                                -- H.221 Annex A for NS-cap and NS-comm

Key ::= CHOICE                                  -- Identifier of a standard or non-standard object
{
        object                  OBJECT IDENTIFIER,
        h221NonStandard         H221NonStandardIdentifier
}

NonStandardParameter                    ::= SEQUENCE
{
        key             Key,
        data            OCTET STRING
}

Handle                                  ::= INTEGER (0..4294967295)

NonStandardIdentifier                   ::= CHOICE
{
   objectOBJECT IDENTIFIER,
   h221nonStandard                      H221NonStandardIdentifier
}

TextString ::= BMPString (SIZE (0..255))        -- Basic Multilingual Plane of ISO/IEC 10646-1 (Unicode)


DeviceClass                             ::= CHOICE
{
        camera                          NULL,
        microphone                      NULL,
        streamPlayerRecorder            NULL,
        slideProjector                  NULL,
        lightSource                     NULL,
        sourceCombiner                  NULL,
        nonStandardDevice               NonStandardIdentifier
}

DeviceID                                ::= INTEGER (0..127)

StreamID                                ::= INTEGER (0..65535)

DeviceProfile                           ::= SEQUENCE
{
        deviceID                        DeviceID,
        audioSourceFlag                 BOOLEAN,
        audioSinkFlag                   BOOLEAN,
        videoSourceFlag                 BOOLEAN,
        videoSinkFlag                   BOOLEAN,
```

```
        remoteControlFlag               BOOLEAN,
        instanceNumber                  INTEGER (0..255),
        deviceName                      TextString OPTIONAL,
        ...
}

StreamProfile                           ::= SEQUENCE
{
        streamID                        StreamID,
        videoStreamFlag                 BOOLEAN,
        sourceChangeFlag                BOOLEAN,
        streamName                      TextString OPTIONAL,
        ...
}

CapabilityID                    ::= CHOICE
{
        standard                INTEGER (0..65535),     -- Assigned by this specification
        nonStandard             Key
}

NonCollapsingCapabilities       ::= SET OF SEQUENCE
{
        capabilityID            CapabilityID,
        applicationData         CHOICE
        {
                deviceList      SET SIZE(0..127) OF DeviceProfile,
                streamList      SET SIZE(0..127) OF StreamProfile
        }
}

--
-- Attribute parameter types
--
Day                             ::= INTEGER(1..31)
Month                           ::= INTEGER(1..12)
Year                            ::= INTEGER(1980..2999)
Hour                            ::= INTEGER(0..23)
Minute                          ::= INTEGER(0..59)
DeviceText                      ::= OCTET STRING (SIZE (0..32))
PanPosition                     ::= INTEGER(-18000..18000)      -- 100ths of a degree
TiltPosition                    ::= INTEGER(-18000..18000)      -- 100ths of a degree
ZoomPosition                    ::= INTEGER(-1023..1023)
IrisPosition                    ::= INTEGER(-127..127)
FocusPosition                   ::= INTEGER(-127..127)
CameraPanSpeed                  ::= INTEGER(1..18000)           -- 100ths of a degree/sec
CameraTiltSpeed                 ::= INTEGER(1..18000)           -- 100ths of a degree/sec
BackLight                       ::= INTEGER(0..255)
WhiteBalance                    ::= INTEGER(0..255)
PresetNumber                    ::= INTEGER(1..255)

StreamPlayerState               ::= CHOICE
{
        playing                 NULL,
        recording               NULL,
        pausedOnRecord          NULL,
        pausedOnPlay            NULL,
        rewinding               NULL,
        fastForwarding          NULL,
        searchingForwards       NULL,
        searchingBackwards      NULL,
```

```
        stopped                 NULL,
        programUnavailable      NULL
}


DevicePresetCapability          ::= SEQUENCE
{
        maxNumber               PresetNumber,
        presetCapability        SET SIZE(0..255) OF SEQUENCE
        {
                presetNumber            PresetNumber,
                storeModeSupported      BOOLEAN,
                presetTextLabel         DeviceText
        }OPTIONAL
}

CameraFilterCapability          ::= SEQUENCE
{
        maxNumberOfFilters      INTEGER(2..255),
        filterTextLabel         SET SIZE(0..255) OF SEQUENCE
        {
                filterNumber            INTEGER(1..255),
                filterTextLabel         DeviceText
        }OPTIONAL
}

CameraLensCapability            ::= SEQUENCE
{
        maxNumberOfLens         INTEGER(2..255),
        accessoryTextLabel      SET SIZE(0..255) OF SEQUENCE
        {
                lensNumber              INTEGER(1..255),
                lensTextLabel           DeviceText
        }OPTIONAL
}

ExternalCameraLightCapability   ::= SEQUENCE
{
        maxNumber               INTEGER(1..10),
        lightTextLabel          SET SIZE(0..10) OF SEQUENCE
        {
                lightNumber             INTEGER(1..10),
                lightLabel              DeviceText
        }OPTIONAL
}

CameraPanSpeedCapability        ::= SEQUENCE
{
        maxSpeed                CameraPanSpeed,
        minSpeed                CameraPanSpeed,
        speedStepSize           CameraPanSpeed
}

CameraTiltSpeedCapability       ::= SEQUENCE
{
        maxSpeed                CameraTiltSpeed,
        minSpeed                CameraTiltSpeed,
        speedStepSize           CameraTiltSpeed
}

MaxBacklight                    ::= INTEGER(1..255)
MaxWhiteBalance                 ::= INTEGER(1..255)
```

```
MinZoomPositionSetSize          ::= INTEGER(1..1023)
MinFocusPositionStepSize              ::= INTEGER(1..127)
MinIrisPositionStepSize         ::= INTEGER(1..127)

PanPositionCapability           ::= SEQUENCE
{
        maxLeft                 INTEGER(-18000..0),     -- Max pan left
        maxRight                INTEGER(0..18000),      -- Max pan right
        minStepSize             INTEGER(1..18000)
}

TiltPositionCapability          ::= SEQUENCE
{
        maxDown                 INTEGER(-18000..0),     -- Max tilt down
        maxUp                   INTEGER(0..18000),      -- Max tilt up
        minStepSize             INTEGER(1..18000)
}

MinZoomMagnificationStepSize ::= INTEGER(1..1000)
MaxNumberOfSlides               ::= INTEGER(1..1024)
MaxSlideDisplayTime             ::= INTEGER(1..255)     -- Max time in seconds
MaxNumberOfPrograms             ::= INTEGER(1..1023)

PlayBackSpeedCapability         ::= SEQUENCE
{
        multiplierFactors       SET SIZE (1..64) OF INTEGER(10..1000),
        divisorFactors          SET SIZE (1..64) OF INTEGER(10..1000)
}

VideoInputsCapability                   ::= SEQUENCE
{
        numberOfDeviceInputs            INTEGER(2..64),
        numberOfDeviceRows              INTEGER(1..64),

        -- The Optional Device list is only included if the inputs are configurable
        availableDevices                SET SIZE(2..64) OF SEQUENCE
        {
                deviceClass             DeviceClass,
                deviceIdentifier        DeviceID
        }OPTIONAL
}

AudioInputsCapability                   ::= SEQUENCE
{
        numberOfDeviceInputs            INTEGER(2..64),

        -- The Optional Device list is only included if the inputs are configurable
        availableDevices                SET SIZE(2..64) OF SEQUENCE
        {
                deviceClass             DeviceClass,
                deviceIdentifier        DeviceID
        }OPTIONAL
}

DeviceAttribute                                 ::= CHOICE
{
        deviceStateSupported                    NULL,
        deviceDateSupported                     NULL,
        deviceTimeSupported                     NULL,
        devicePresetSupported                   DevicePresetCapability,
        irisModeSupported                       NULL,
        focusModeSupported                      NULL,
```

pointingModeSupported **NULL,**
cameraLensSupported **CameraLensCapability,**
cameraFilterSupported **CameraFilterCapability,**
homePositionSupported **NULL,**
externalCameraLightSupported **ExternalCameraLightCapability,**
clearCameraLensSupported **NULL,**
cameraPanSpeedSupported **CameraPanSpeedCapability,**
cameraTiltSpeedSupported **CameraTiltSpeedCapability,**
backLightModeSupported **NULL,**
backLightSettingSupported **MaxBacklight,**
whiteBalanceSettingSupported **MaxWhiteBalance,**
whiteBalanceModeSupported **NULL,**
calibrateWhiteBalanceSupported **NULL,**
focusImageSupported **NULL,**
captureImageSupported **NULL,**
panContinuousSupported **NULL,**
tiltContinuousSupported **NULL,**
zoomContinuousSupported **NULL,**
focusContinuousSupported **NULL,**
irisContinuousSupported **NULL,**
zoomPositionSupported **MinZoomPositionSetSize,**
focusPositionSupported **MinFocusPositionStepSize,**
irisPositionSupported **MinIrisPositionStepSize,**
panPositionSupported **PanPositionCapability,**
tiltPositionSupported **TiltPositionCapability,**
zoomMagnificationSupported **MinZoomMagnificationStepSize,**
panViewSupported **NULL,**
tiltViewSupported **NULL,**
selectSlideSupported **MaxNumberOfSlides,**
selectNextSlideSupported **NULL,**
slideShowModeSupported **NULL,**
playSlideShowSupported **NULL,**
setSlideDisplayTimeSupported **MaxSlideDisplayTime,**
continuousRewindSupported **NULL,**
continuousFastForwardSupported **NULL,**
searchBackwardsSupported **NULL,**
searchForwardsSupported **NULL,**
pauseSupported **NULL,**
selectProgramSupported **MaxNumberOfPrograms,**
nextProgramSupported **NULL,**
gotoNormalPlayTimePointSupported **NULL,**
readStreamPlayerStateSupported **NULL,**
readProgramDurationSupported **NULL,**
continuousPlayBackModeSupported **NULL,**
playbackSpeedSupported **PlayBackSpeedCapability,**
playSupported **NULL,**
setAudioOutputStateSupported **NULL,**
playToNormalPlayTimePointSupported **NULL,**
recordSupported **NULL,**
recordForDurationSupported **NULL,**
configurableVideoInputsSupported **VideoInputsCapability,**
videoInputsSupported **VideoInputsCapability,**
configurableAudioInputsSupported **AudioInputsCapability,**
audioInputsSupported **AudioInputsCapability,**

-- Attributes for Device Events

deviceLockStateChangedSupported **NULL,**
deviceAvailabilityChangedSupported **NULL,**
cameraPannedToLimitSupported **NULL,**
cameraTiltedToLimitSupported **NULL,**
cameraZoomedToLimitSupported **NULL,**
cameraFocusedToLimitSupported **NULL,**

```
        autoSlideShowFinishedSupported          NULL,
        streamPlayerStateChangeSupported        NULL,
        streamPlayerProgramChangeSupported      NULL,
        nonStandardAttributeSupported           NonStandardParameter,
        ...
}

DeviceState                 ::=CHOICE
{
        active              NULL,
        inactive            NULL
}

DeviceDate                  ::= SEQUENCE
{
        day                 Day,
        month               Month,
        year                Year
}

DeviceTime                  ::= SEQUENCE
{
        hour                Hour,
        minute              Minute
}

DevicePreset                ::= SEQUENCE
{
        presetNumber        PresetNumber,
        mode                CHOICE
        {
                store       NULL,
                activate    NULL
        }
}

Mode                ::= CHOICE
{
        manual      NULL,
        auto        NULL
}

PointingToggle      ::= CHOICE
{
        manual      NULL,
        auto        NULL,
        toggle      NULL
}


SelectExternalLight ::= CHOICE
{
        lightNumber INTEGER(1..10),
        none        NULL
}

PanContinuous       ::= SEQUENCE
{
        panDirectionCHOICE
        {
                left    NULL,
                right   NULL,
```

```
                stop            NULL,
                continue        NULL
        },
        timeOut                 INTEGER(50..1000)       -- Milliseconds
}

TiltContinuous                  ::= SEQUENCE
{
        tiltDirection  CHOICE
        {
                up              NULL,
                down            NULL,
                stop            NULL,
                continue        NULL
        },
        timeOut                 INTEGER(50..1000)       -- Milliseconds
}

ZoomContinuous                  ::= SEQUENCE
{
        zoomDirection           CHOICE
        {
                telescopic      NULL,
                wide            NULL,
                stop            NULL,
                continue        NULL
        },
        timeOut                 INTEGER(50..1000)       -- Milliseconds
}


FocusContinuous                 ::= SEQUENCE
{
        focusDirection          CHOICE
        {
                near            NULL,
                far             NULL,
                stop            NULL,
                continue        NULL
        },
        timeOut                 INTEGER(50..1000)       -- Milliseconds
}

IrisContinuous                  ::= SEQUENCE
{
        irisDirection CHOICE
        {
                darker          NULL,
                lighter         NULL,
                stop            NULL,
                continue        NULL
        },
        timeOut                 INTEGER(50..1000)       -- Milliseconds
}

PositioningMode                 ::= CHOICE
{
        relative                NULL,
        absolute                NULL
}
```

```
CameraLensNumber        ::= INTEGER(1..255)
CameraFilterNumber      ::= INTEGER(1..255)

SetZoomPosition         ::= SEQUENCE
{
        zoomPosition        ZoomPosition,
        positioningMode     PositioningMode
}

SetFocusPosition        ::= SEQUENCE
{
        focusPosition       FocusPosition,
        positioningMode     PositioningMode
}

SetIrisPosition         ::= SEQUENCE
{
        irisPosition        IrisPosition,
        positioningMode     PositioningMode
}

SetPanPosition          ::= SEQUENCE
{
        panPosition         PanPosition,
        positioningMode     PositioningMode
}

SetTiltPosition         ::= SEQUENCE
{
        tiltPosition        TiltPosition,
        positioningMode     PositioningMode
}

ZoomMagnification::= INTEGER(10..1000)
PanView                 ::= INTEGER(-1000..1000)
TiltView                ::= INTEGER(-1000..1000)
SlideNumber             ::= INTEGER(0..1023)

SelectDirection         ::= CHOICE
{
        next                NULL,
        previous            NULL
}

AutoSlideShowControl    ::= CHOICE
{
        start               NULL,
        stop                NULL,
        pause               NULL
}

AutoSlideDisplayTime        ::= INTEGER(1..255)     -- Automatic slide display time in seconds
ProgramNumber               ::= INTEGER(1..1023)

ProgramDuration             ::= SEQUENCE
{
        hours               INTEGER(0..24),
        minutes             INTEGER(0..59),
        seconds             INTEGER(0..59),
        microseconds        INTEGER(0..99999)
}
```

```
PlaybackSpeed               ::= SEQUENCE
{
        scaleFactor         INTEGER(10..1000),
        multiplyFactor      BOOLEAN         -- TRUE to multiply, FALSE to divide
}

RecordForDuration           ::= SEQUENCE
{
        hours               INTEGER(0..24),
        minutes             INTEGER(0..59),
        seconds             INTEGER(0..59)
}

DeviceInputs                ::= SEQUENCE
{
        -- When used with the configurableVideoInputs type this device list must only contain
        -- devices that source a video stream.
        -- When used with the configurableAudioInputs type this device list must only contain
        -- devices that source an audio stream.
        inputDevices                SET SIZE(2..64) OF SEQUENCE
        {
                deviceClass         DeviceClass,
                deviceIdentifier    DeviceID
        }
}

ControlAttribute            ::= CHOICE
{
        setDeviceState              DeviceState,
        setDeviceDate               DeviceDate,
        setDeviceTime               DeviceTime,
        setDevicePreset             DevicePreset,
        setIrisMode                 Mode,
        setFocusMode                Mode,
        setBackLightMode            Mode,
        setPointingMode             PointingToggle,
        selectCameraLens            CameraLensNumber,
        selectCameraFilter          CameraFilterNumber,
        gotoHomePosition            NULL,
        selectExternalLight         SelectExternalLight,
        clearCameraLens             NULL,
        setCameraPanSpeed           CameraPanSpeed,
        setCameraTiltSpeed          CameraTiltSpeed,

        setBackLight                BackLight,
        setWhiteBalance             WhiteBalance,
        setWhiteBalanceMode         Mode,
        calibrateWhiteBalance       NULL,
        focusImage                  NULL,
        captureImage                NULL,
        panContinuous               PanContinuous,
        tiltContinuous              TiltContinuous,
        zoomContinuous              ZoomContinuous,
        focusContinuous             FocusContinuous,
        setZoomPosition             SetZoomPosition,
        setFocusPosition            SetFocusPosition,
        setIrisPosition             SetIrisPosition,
        setPanPosition              SetPanPosition,
        setTiltPosition             SetTiltPosition,
        setZoomMagnification        ZoomMagnification,
        setPanView                  PanView,
```

```
        setTiltView                     TiltView,
        selectSlide                     SlideNumber,
        selectNextSlide                 SelectDirection,
        playAutoSlideShow               AutoSlideShowControl,
        setAutoSlideDisplayTime         AutoSlideDisplayTime,
        continuousRewindControl         BOOLEAN,  -- TRUE to start, FALSE to Stop
        continuousFastForwardControl    BOOLEAN,  -- TRUE to start, FALSE to Stop
        searchBackwardsControl          BOOLEAN,  -- TRUE to start, FALSE to Stop
        searchForwardsControl           BOOLEAN,  -- TRUE to start, FALSE to Stop
        pause                           BOOLEAN,  -- TRUE to pause, FALSE to release
        selectProgram                   ProgramNumber,
        nextProgramSelect               SelectDirection,
        gotoNormalPlayTimePoint         ProgramDuration,
        continuousPlayBackMode          BOOLEAN,  -- TRUE to set continuous mode
        setPlaybackSpeed                PlaybackSpeed,
        play                            BOOLEAN,  -- TRUE to pause, FALSE to release
        setAudioOutputMute              BOOLEAN,  -- TRUE to mute, FALSE to unmute
        playToNormalPlayTimePoint       ProgramDuration,
        record                          BOOLEAN,  -- TRUE to start, FALSE to Stop
        recordForDuration               RecordForDuration,
        configureVideoInputs            DeviceInputs, -- Device list contains Video devices
        configureAudioInputs            DeviceInputs, -- Device list contains Audio Devices
        nonStandardControl              NonStandardParameter,
        ...
}

StatusAttributeIdentifier               ::= CHOICE
{
        getDeviceState                  NULL,
        getDeviceDate                   NULL,
        getDeviceTime                   NULL,
        getdevicePreset                 NULL,
        getIrisMode                     NULL,
        getFocusMode                    NULL,
        getBacklightMode                NULL,
        getPointingMode                 NULL,
        getCameraLens                   NULL,
        getCameraFilter                 NULL,
        getExternalLight                NULL,
        getCameraPanSpeed               NULL,
        getCameraTiltSpeed              NULL,
        getBackLightMode                NULL,
        getBackLight                    NULL,
        getWhiteBalance                 NULL,
        getWhiteBalanceMode             NULL,
        getZoomPosition                 NULL,
        getFocusPosition                NULL,
        getIrisPosition                 NULL,
        getPanPosition                  NULL,
        getTiltPosition                 NULL,
        getSelectedSlide                NULL,
        getAutoSlideDisplayTime         NULL,
        getSelectedProgram              NULL,
        getStreamPlayerState            NULL,
        getCurrentProgramDuration       NULL,
        getPlaybackSpeed                NULL,
        getAudioOutputState             NULL,
        getConfigurableVideoInputs      NULL,
        getVideoInputs                  NULL,
        getConfigurableAudioInputs      NULL,
        getAudioInputs                  NULL,
```

```
        getNonStandardStatus        NonStandardIdentifier,
        ...
}

CurrentDeviceState      ::= CHOICE
{
        deviceState         DeviceState,
        unknown             NULL
}

CurrentDeviceDate       ::= SEQUENCE
{
        currentDay          CHOICE
        {
                day         Day,
                unknown     NULL
        },

        currentMonth        CHOICE
        {
                month       Month,
                unknown     NULL
        },

        currentYear         CHOICE
        {
                year        Year,
                unknown     NULL
        }
}

CurrentDeviceTime       ::= SEQUENCE
{
        currentHour         CHOICE
        {
                hour        Hour,
                unknown     NULL
        },

        currentMinute       CHOICE
        {
                minute      Minute,
                unknown     NULL
        }
}

CurrentDevicePreset     ::= CHOICE
{
        preset              PresetNumber,
        unknown             NULL
}

CurrentMode             ::= CHOICE
{
        mode                Mode,
        unknown             NULL
}

CurrentPointingMode     ::= CHOICE
{
```

```
        automatic           NULL,
        manual              NULL,
        unknown             NULL
}

CurrentCameraLensNumber     ::= CHOICE
{
        lensNumber          CameraLensNumber,
        unknown             NULL
}

CurrentCameraFilterNumber   ::= CHOICE
{
        lensNumber          CameraFilterNumber,
        unknown             NULL
}

CurrentExternalLight        ::= CHOICE
{
        lightNumber         INTEGER(1..10),
        none                NULL,
        unknown             NULL
}

CurrentCameraPanSpeed       ::= CHOICE
{
        speed               CameraPanSpeed,
        unknown             NULL
}

CurrentCameraTiltSpeed      ::= CHOICE
{
        speed               CameraTiltSpeed,
        unknown             NULL
}

CurrentBackLight            ::= CHOICE
{
        backLight           BackLight,
        unknown             NULL
}

CurrentWhiteBalance         ::= CHOICE
{
        whiteBalance        WhiteBalance,
        unknown             NULL
}

CurrentZoomPosition         ::= CHOICE
{
        zoomPosition        ZoomPosition,
        unknown             NULL
}

CurrentFocusPosition        ::= CHOICE
{
        focusPosition       FocusPosition,
        unknown             NULL
}

CurrentIrisPosition         ::= CHOICE
{
```

```
        irisPosition            IrisPosition,
        unknown                 NULL
}

CurrentPanPosition              ::= CHOICE
{
        panPosition             PanPosition,
        unknown                 NULL
}

CurrentTiltPosition             ::= CHOICE
{
        tiltPosition            TiltPosition,
        unknown                 NULL
}

CurrentSlide                    ::= CHOICE
{
        slide                   SlideNumber,
        unknown                 NULL
}

CurrentAutoSlideDisplayTime     ::= CHOICE
{
        time                    AutoSlideDisplayTime,
        unknown                 NULL
}

CurrentSelectedProgram          ::= CHOICE
{
        program                 ProgramNumber,
        unknown                 NULL
}

CurrentStreamPlayerState        ::= CHOICE
{
        state                   StreamPlayerState,
        unknown                 NULL
}

CurrentPlaybackSpeed            ::= CHOICE
{
        speed                   PlaybackSpeed,
        unknown                 NULL
}

CurrentAudioOutputMute          ::= CHOICE
{
        mute                    BOOLEAN,  -- TRUE if muted
        unknown                 NULL
}

StatusAttribute                         ::= CHOICE
{
        currentdeviceState              CurrentDeviceState,
        currentDeviceDate               CurrentDeviceDate,
        currentDeviceTime               CurrentDeviceTime,
        currentDevicePreset             CurrentDevicePreset,
        currentIrisMode                 CurrentMode,
        currentFocusMode                CurrentMode,
        currentBackLightMode            CurrentMode,
```

```
        currentPointingMode             CurrentPointingMode,
        currentCameraLens               CurrentCameraLensNumber,
        currentCameraFilter             CurrentCameraFilterNumber,
        currentExternalLight            CurrentExternalLight,
        currentCameraPanSpeed           CurrentCameraPanSpeed,
        currentCameraTiltSpeed          CurrentCameraTiltSpeed,
        currentBackLight                CurrentBackLight,
        currentWhiteBalance             CurrentWhiteBalance,
        currentWhiteBalanceMode         CurrentMode,
        currentZoomPosition             CurrentZoomPosition,
        currentFocusPosition            CurrentFocusPosition,
        currentIrisPosition             CurrentIrisPosition,
        currentPanPosition              CurrentPanPosition,
        currentTiltPosition             CurrentTiltPosition,
        currentSlide                    CurrentSlide,
        currentAutoSlideDisplayTime     CurrentAutoSlideDisplayTime,
        currentSelectedProgram          CurrentSelectedProgram,
        currentstreamPlayerState        CurrentStreamPlayerState,
        currentProgramDuration          ProgramDuration,
        currentPlaybackSpeed            CurrentPlaybackSpeed,
        currentAudioOutputMute          CurrentAudioOutputMute,
        configurableVideoInputs         DeviceInputs,
        videoInputs                     DeviceInputs,
        configurableAudioInputs         DeviceInputs,
        audioInputs                     DeviceInputs,
        nonStandardStatus               NonStandardParameter,
        ...
}

DeviceEventIdentifier                   ::= CHOICE
{
        requestDeviceLockChanged        NULL,
        requestDeviceAvailabilityChanged NULL,
        requestCameraPannedToLimit      NULL,
        requestCameraTiltedToLimit      NULL,
        requestCameraZoomedToLimit      NULL,
        requestCameraFocusedToLimit     NULL,
        requestAutoSlideShowFinished    NULL,
        requestStreamPlayerStateChange  NULL,
        requestStreamPlayerProgramChange NULL,
        requestNonStandardEvent         NonStandardIdentifier,
        ...
}

CameraPannedToLimit             ::= CHOICE
{
        left                    NULL,
        right                   NULL
}

CameraTiltedToLimit             ::= CHOICE
{
        up                      NULL,
        down                    NULL
}

CameraZoomedToLimit             ::= CHOICE
{
        telescopic              NULL,
        wide                    NULL
}
```

```
CameraFocusedToLimit        ::= CHOICE
{
        near                NULL,
        far                 NULL
}

DeviceEvent                         ::= CHOICE
{
        deviceLockChanged           BOOLEAN,  -- TRUE if now locked FALSE if now unlocked
        deviceAvailabilityChanged   BOOLEAN,  -- TRUE if available FALSE if now unavailable
        cameraPannedToLimit         CameraPannedToLimit,
        cameraTiltedToLimit         CameraTiltedToLimit,
        cameraZoomedToLimit         CameraZoomedToLimit,
        cameraFocusedToLimit        CameraFocusedToLimit,
        autoSlideShowFinished       NULL,
        streamPlayerStateChange     StreamPlayerState,
        streamPlayerProgramChange   ProgramNumber,
        nonStandardEvent            NonStandardParameter,
        ...
}

-- ========================================================================
-- Part 2:  PDU Messages
-- ========================================================================

SourceSelectRequest                 ::= SEQUENCE
{
        requestHandle               Handle,
        deviceClass                 DeviceClass,
        deviceID                    DeviceID,
        streamIdentifier            StreamID,
        ...
}

SourceSelectResponse                ::= SEQUENCE
{
        requestHandle               Handle,
        result          CHOICE
        {
                successful          NULL,
                requestDenied       NULL,
                deviceUnavailable   NULL,
                invalidStreamID     NULL,
                currentDeviceIsLocked   NULL,
                deviceIncompatible  NULL,
                ...
        },
        ...
}

SourceEventsRequest                 ::= SEQUENCE
{
        requestHandle               Handle,
        streamIdentifier            StreamID,
        sourceEventNotify           BOOLEAN,  -- TRUE to request source events
        ...
}
```

```
SourceEventsResponse                    ::= SEQUENCE
{
        requestHandle                   Handle,
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        result          CHOICE
        {
                successful              NULL,
                eventsNotSupported      NULL,
                invalidStreamID         NULL,
                ...
        },
        ...
}

SourceChangeEventIndication             ::= SEQUENCE
{
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        ...
}

DeviceAttributeRequest                  ::= SEQUENCE
{
        requestHandle                   Handle,
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        ...
}

DeviceAttributeResponse                 ::= SEQUENCE
{
        requestHandle                   Handle,
        deviceAttributeList             SET  OF DeviceAttribute OPTIONAL,
        result          CHOICE
        {
                successful              NULL,
                requestDenied           NULL,
                unknownDevice           NULL,
                ...
        },
        ...
}

DeviceLockRequest                       ::= SEQUENCE
{
        requestHandle                   Handle,
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        lockFlag                        BOOLEAN,  -- TRUE to lock
        ...
}

DeviceLockResponse                      ::= SEQUENCE
{
        requestHandle                   Handle,
        result                          CHOICE
        {
                successful              NULL,
                requestDenied           NULL,
                unknownDevice           NULL,
```

```
                lockingNotSupported             NULL,
                deviceAlreadyLocked             NULL,
                ...
        },
        ...
}

DeviceLockEnquireRequest                ::= SEQUENCE
{
        requestHandle                   Handle,
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        ...
}

DeviceLockEnquireResponse               ::= SEQUENCE
{
        requestHandle                   Handle,
        result          CHOICE
        {
                lockRequired            NULL,
                lockNotRequired         NULL,
                unknownDevice           NULL,
                ...
        },
        ...
}

DeviceLockTerminatedIndication          ::= SEQUENCE
{
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        ...
}

DeviceControlRequest                    ::= SEQUENCE
{
        requestHandle                   Handle,
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        controlAttributeList            SET SIZE (1..8) OF ControlAttribute,
        ...
}

DeviceStatusEnquireRequest              ::= SEQUENCE
{
        requestHandle                   Handle,
        deviceClass                     DeviceClass,
        deviceID                        DeviceID,
        statusAttributeIdentifierList   SET SIZE (1..16) OF StatusAttributeIdentifier,
        ...
}

DeviceStatusEnquireResponse             ::= SEQUENCE
{
        requestHandle                   Handle,
        statusAttributeList             SET SIZE (1..16) OF StatusAttribute OPTIONAL,
        result          CHOICE
        {
                successful              NULL,
                requestDenied           NULL,
                unknownDevice           NULL,
```

```
                deviceUnavailable           NULL,
                deviceAttributeError        NULL,
                ...
        },
        ...
}

ConfigureDeviceEventsRequest        ::= SEQUENCE
{
        requestHandle               Handle,
        deviceClass                 DeviceClass,
        deviceID                    DeviceID,
        deviceEventIdentifierList   SET OF DeviceEventIdentifier,
        ...
}

ConfigureDeviceEventsResponse       ::= SEQUENCE
{
        requestHandle               Handle,
        result                      CHOICE
        {
                successful          NULL,
                requestDenied       NULL,
                unknownDevice       NULL,
                deviceUnavailable   NULL,
                deviceAttributeError NULL,
                ...
        },
        ...
}

DeviceEventNotifyIndication         ::= SEQUENCE
{
        deviceClass                 DeviceClass,
        deviceID                    DeviceID,
        deviceEventList             SET SIZE (1..8) OF DeviceEvent,
        ...
}

NonStandardPDU                      ::= SEQUENCE
{
        nonStandardData             NonStandardParameter,
        ...
}

-- ========================================================================
-- Part 3:  Messages sent using  lower layer protocol
-- ========================================================================
RDCPDU                              ::= CHOICE
{
        request                     RequestPDU,
        response                    ResponsePDU,
        indication                  IndicationPDU
}

RequestPDU                          ::= CHOICE
{
        sourceSelectRequest         SourceSelectRequest,
        sourceEventsRequest         SourceEventsRequest,
        deviceAttributeRequest      DeviceAttributeRequest,
        deviceLockRequest           DeviceLockRequest,
        deviceLockEnquireRequest    DeviceLockEnquireRequest,
```

```
        deviceControlRequest                DeviceControlRequest,
        deviceStatusEnquireRequest          DeviceStatusEnquireRequest,
        configureDeviceEventsRequest        ConfigureDeviceEventsRequest,
        nonStandardRequest                  NonStandardPDU,
        ...
}

ResponsePDU                                 ::= CHOICE
{
        sourceSelectResponse                SourceSelectResponse,
        sourceEventsResponse                SourceEventsResponse,
        deviceAttributeResponse             DeviceAttributeResponse,
        deviceLockResponse                  DeviceLockResponse,
        deviceLockEnquireResponse           DeviceLockEnquireResponse,
        deviceStatusEnquireResponse         DeviceStatusEnquireResponse,
        configureDeviceEventsResponse       ConfigureDeviceEventsResponse,
        nonStandardResponse                 NonStandardPDU,
        ...
}

IndicationPDU                               ::= CHOICE
{
        sourceChangeEventIndication         SourceChangeEventIndication,
        deviceLockTerminatedIndication      DeviceLockTerminatedIndication,
        deviceEventNotifyIndication         DeviceEventNotifyIndication,
        nonStandardIndication               NonStandardPDU,
        ...
}

END
```

ANNEX A

Standard Device Classes and Device Control Specifications

A.1 Standard Device Classes

This following table lists the standard device classes.

Device Class Table

Device Class Name
Camera
Microphone
Stream player recorder
Slide Projector
Light Source
Source Combiner

A.1.1 Device Identifier

A Device Identifier is a locally unique number in the range 0 to 127, allowing discrimination between multiple devices of the same class. The Device Identifier is listed as part of the Device Profile, advertised using the capability mechanism of the lower layer protocol. Each node may therefore support up to 128 devices that use the same device class.

A.1.1.1 Interoperability with H.281 far end camera control

A terminal that supports H.282 may interoperate with a terminal that only supports H.281 via a gateway that performs the relevant transcoding between the two protocols. In order to allow the H.281 terminal to perform remote camera source selection the manufacturer of the H.282 terminal should consider assigning the device identifiers in the same way as described by Recommendation H.281.

Device	Device ID
Main Camera	1
Auxiliary Camera	2
Document Camera	3
Auxiliary Document Camera	4

A.1.2 Device Attributes

A number of standard device classes are defined. A library of standard device attributes is defined for use by the device classes. For each standard class a set of attributes, which may be used by that class, is listed. Where appropriate, it is permissible to use other standard attributes which are not listed as part of a device class. Where *a priori* information is available, a node may use non-standard attributes as part of a standard device class. A node may also declare a non-standard device class using standard or non-standard attributes.

Each attribute is described individually with a table showing all the available attribute fields. Each field is assigned a particular type depending upon how it is used. The following table shows Attribute Field Types and the primitives in which the attribute fields types are used.

Attribute Field Type	Description	Primitives
Attribute Range	The attribute field conveys a value that is used as the allowable limit for setting a device attribute.	RDC-Device-Attributes response RDC-Device-Attributes confirm
Set Only	The attribute field is used to set a new device attribute value.	RDC-Device-Control request RDC-Device-Control indication
Read Only	The attribute field is used to return the current device attribute value.	RDC-Device-Status response RDC-Device-Status confirm
Set or Read (see Note)	The attribute field is used to either set a new device attribute value or return the current device attribute value depending upon the primitive in which it is used.	RDC-Device-Control request RDC-Device-Control indication RDC-Device-Status response RDC-Device-Status confirm
Event Notification	The attribute field conveys information about the nature of the event being notified.	RDC-Device-Event-Notify indication
NOTE – When a Set or Read field is used to read an attribute value using the RDC-Device-Status primitive, the value Unknown may be returned to indicate that the current status for that attribute field could not be determined (for example the Host node lost communication with the device at the point that it was trying to obtain the status). It is illegal to use this value to set an attribute field when using the RDC-Device-Control primitive.		

A.1.3 Camera Device Class

In a video conferencing environment, cameras are normally the principle source of video. The camera device class therefore provides a comprehensive remote control capability allowing control over the camera position and viewing angle as well as the ability to manipulate parameters relating to the image.

Control Attributes
Device State
Device Date
Device Time
Device Pre-set
Iris Mode
Focus Mode
Pointing Mode
Camera Lens
Camera Filter
Go To Camera Home Position
External Camera Light
Clear Camera Lens
Camera Pan Speed
Camera Tilt Speed
Camera Back Light Mode
Camera Back light Setting
Camera White Balance Alignment
Camera White Balance Alignment Mode
Calibrate White Balance Alignment
Focus Image
Capture Image
Pan Continuous
Tilt Continuous
Zoom Continuous
Focus Continuous
Iris Continuous
Zoom Position
Focus Position
Iris Position
Pan Position
Tilt Position
Zoom Magnification
Pan View
Tilt View

Event Attributes
Device Lock State Changed
Device Availability Changed
Camera Panned to Limit
Camera Tilted to Limit
Camera Zoomed to Limit
Camera Focused to Limit

A.1.4 Slide Presenter Device Class

The slide presenter device class provides a means for controlling, configuring and collecting the status of a remote slide presenter. The device class may also contain camera control attributes where the slide presenter contains a camera.

Control Attributes
Device State
Select Slide
Select Next Slide
Slide Show Mode
Play Automatic Slide Show
Automatic Side Display Time

Event Attributes
Device Lock State Changed
Device Availability Changed

A.1.5 Microphone Device Class

In a conferencing environment, a microphone is normally the principle source of audio. The microphone device class provides a means for controlling and collecting the status of a remote microphone.

Control Attributes
Device State

Event Attributes
Device Lock State Changed
Device Availability Changed

A.1.6 Light Source Device Class

The light source device class provides a means for controlling and collecting the status of a remote light source that can be considered a device in its own right and is not part of another device. An example of when this device class may be used is when a light source is part of an external security installation. The light source will need to be remotely activated when the natural light fades and removed when the natural light returns.

Control Attributes
Device State

Event Attributes
Device Lock State Changed
Device Availability Changed

A.1.7 Stream Player Recorder Device Class

The Stream Player Recorder device class provides a means for controlling, configuring and collecting the status of a remote Record/Playback device. This device may be connected to both a video and audio stream or only a video stream or only an audio stream. The Device Profile should be consulted to determine this. This device class may be used to model such devices as a Video Cassette Recorder or a CD Player.

This device class provides a means for a remote RDC node to select a program to be loaded into the stream player recorder. In the case of a VCR a program may be the cassette tape, or for a CD player it may be a disk. The stream positioning attributes then operate within the bounds of the selected program. Some stream positioning attributes make use of the concept of Normal Play Time which is a measurement of the amount of play time when played at normal speed. The Normal Play Time is measured from the start of the current program. This device class may use device pre-sets to represent tracks on a CD or index marks on a VCR tape. This class may additionally provide a means for recording an incoming stream.

Control Attributes
Device State
Device Pre-set
Device Date
Device Time
Continuous Rewind
Continuous Fast Forward
Search Backwards
Search Forwards
Pause
Select Program
Next Program
Go To Normal Play Time Point
Read Stream Player Recorder State
Read Current Program Duration
Continuous Play Back Mode
Play
Audio Output State
Play to Normal Play Time Point
Record
Record for Duration

Event Attributes
Device Lock State Changed
Device Availability Changed
Stream Player Recorder State Change
Stream Player Recorder Program Change

A.1.8 Source Combiner Device Class

The source selection device class provides a means for selecting the source output of two or more devices to be combined together to form one output stream.

This class may make use of the device pre-set attribute. The device pre-sets may act upon attached devices. For example, a source combiner may have two cameras attached with a device pre-set that sets both cameras to predefined positions when selected. This device pre-set would be separate to any device pre-sets that the individual cameras may advertise.

Control Attributes
Device State
Device Pre-set
Configurable Video Device Inputs
Video Device Inputs
Configurable Audio Device Inputs
Audio Device Inputs

Event Attributes
Device Lock State Changed
Device Availability Changed

A.2 Device Control Attributes

These control attributes are general attributes for use by any device class as appropriate.

Device State

This attribute allows a remote application to set a device to an active or inactive state or read the current device state.

Attribute Fields	Values	Field Type
Device state	Active/Inactive/Unknown	Set or Read

Device Date

This attribute allows the current device date to be set or read.

Attribute Fields	Values	Field Type
Day	1 .. 31/Unknown	Set or Read
Month	1 .. 12/Unknown	Set or Read
Year	1980 .. 2999/Unknown	Set or Read

Device Time

This attribute allows the current device time to be set or read.

Attribute Fields	Values	Field Type
Hours	0 .. 23/Unknown	Set or Read
Minutes	0 .. 59/Unknown	Set or Read

Device Pre-set

This attribute allows the selection or storing of a particular device pre-set.

Each pre-set has an optional Unicode text label associated with it which may be used to provide descriptive text to a remote user. Each text label is limited to a maximum size of 32 characters.

A remote RDC application can read this attribute to determine the current active device pre-set. If the device is currently in a state that is defined by a pre-set, then this attribute will return the number of that pre-set otherwise the value unknown will be returned.

Attribute Fields	Values	Field Type
Maximum number available	1 .. 255	Attribute Range
Number of text label records	0 .. 255	Attribute Range
List of text label records	List of up to 255 label records	Attribute Range
Device pre-set	1 .. 255/Unknown	Set or Read
Mode	Store, Activate	Set Only

Each text label record has the following format:

Pre-set number	1 .. 255
Store Mode supported	TRUE, FALSE
Associated text label	Text string

The Mode field indicates whether the device should Store the current coordinates or state of the device as the new value of this pre-set, or if it should just Activate the preset. If an application attempts to store a pre-set but Store Mode is not supported, the pre-set store request will be ignored.

A.2.1 Camera Control Attributes

The attributes in this group may be used for controlling a camera. Camera control movement parameters may be specified relative to the current position, or as an absolute value. The terms left and right used in this subclause are used to describe movement as viewed from the camera's perspective.

Iris Mode

This attribute allows the camera iris mode to be either set or read.

Attribute Fields	Values	Field Type
Mode	Manual/Auto/Unknown	Set or Read

Focus Mode

This attribute allows the camera focusing mode to be either set or read.

Attribute Fields	Values	Field Type
Mode	Manual/Auto/Unknown	Set or Read

Pointing Mode

This attribute allows the camera pointing mode to be either set or read. When the camera is in auto pointing mode, some or all of the following manual commands will have no effect depending upon which axes of movement the automatic pointing mode supports:

– Pan Continuous;

– Tilt Continuous;

– Zoom Continuous;

– Pan Position;

– Tilt Position;

– Zoom Position.

Attribute Fields	Values	Field Type
Mode	Manual/Auto/Toggle/Unknown	Set or Read

Camera Lens

This attribute allows the selection of a particular camera lens where the camera has more than one lens available. Each camera lens has an optional Unicode text label associated with it which may be used to provide descriptive text to a remote user. Each text label is limited to a maximum size of 32 characters.

A remote RDC application can read this attribute to determine the current selected camera lens.

Attribute Fields	Values	Field Type
Maximum number available	2 .. 255	Attribute Range
Number of text label records	0 .. 255	Attribute Range
List of text label records	List of up to 255 records	Attribute Range
Lens number	1 .. 255/Unknown	Set or Read

Each text label record has the following format:

Camera lens number	1 .. 255
Associated text label	Text string

Camera Filter

This attribute allows the selection of a particular camera filter where the camera has more than one filter available. Each camera filter has an optional Unicode text label associated with it which may be used to provide descriptive text to a remote user. Each text label is limited to a maximum size of 32 characters.

An RDC application can read this attribute to determine the current selected camera filter.

Attribute Fields	Values	Field Type
Maximum number available	2 .. 255	Attribute Range
Number of text label records	0 .. 255	Attribute Range
List of text label records	List of up to 255 records	Attribute Range
Filter number	1 .. 255/Unknown	Set or Read

Each text label record has the following format:

Filter number	1 .. 255
Associated text label	Text string

Go To Camera Home Position

This attribute, when used as part of the RDC-Device-Control request primitive, commands the camera to move to its default home position. It has no attribute fields.

External Camera Light

This attribute is used to control an external light source that is associated with a camera. A light source may be switched on in order to illuminate the camera's field of view. Up to 10 lights may be available. The Select field determines which light is illuminated. Only one light source may be illuminated at one time so any previously illuminated light is switched off. A value of None in the Select field will turn off all lights. Each light has an optional Unicode text label associated with it which may be used to provide descriptive text to a remote user. Each text label is limited to a maximum size of 32 characters.

An RDC application can read this attribute to determine the current selected external light.

Attribute Fields	Values	Field Type
Maximum number available	1 .. 10	Attribute Range
Number of text label records	0 .. 10	Attribute Range
List of text label records	List of up to 10 records	Attribute Range
Select	None/1 .. 10/Unknown	Set or Read

Each text label record has the following format:

Light number	1 .. 10
Associated text label	Text string

Clear Camera Lens

This attribute, when used as part of the RDC-Device-Control request primitive, commands the camera to clear the lens. For example, this attribute may be used by a camera situated in an outside location which has an external wiper attached to clear rain and dirt from the lens. It has no attribute fields.

Camera Pan Speed

This attribute configures the speed of camera movement used when panning. The speed is expressed as the number of degrees of movement per second. A range from 1 to +18000 is used to represent a speed from 0.01° to 180° degrees per second in steps of 0.01° (= 0° 0' 36"). A minimum and maximum value is declared to indicate the actual range supported. A step size is declared which indicates the increments that are applied to the supported range starting from the minimum range value to calculate the actual speed values supported by this device.

This attribute can be read to determine the current panning speed.

Attribute Fields	Values	Field Type
Maximum pan speed	1 .. 18000	Attribute Range
Minimum pan speed	1 .. 18000	Attribute Range
Pan speed step size	1 .. 18000	Attribute Range
Pan speed	1 .. 18000	Set or Read

Camera Tilt Speed

This attribute configures the speed of camera movement used when tilting. The speed is expressed as the number of degrees of movement per second. A range from 1 to +18000 is used to represent a speed from 0.01° to 180° degrees per second in steps of 0.01° (= 0° 0' 36"). A step size is declared which indicates the increments that are applied to the range starting from 1 to calculate the values supported by this device.

This attribute can be read to determine the current tilting speed.

Attribute Fields	Values	Field Type
Maximum Tilt speed	1 .. 18000	Attribute Range
Minimum Tilt speed	1 .. 18000	Attribute Range
Tilt speed step size	1 .. 18000	Attribute Range
Tilt speed	1 .. 18000	Set or Read

Camera Back Light Mode

This attribute allows a remote RDC application to set or read the camera back light mode.

Attribute Fields	Values	Field Type
Mode	Manual/Auto/Unknown	Set or Read

Camera Back Light Setting

This attribute allows a remote RDC application to set or read the camera back light setting.

Attribute Fields	Values	Field Type
Maximum back light setting	1 .. 255	Attribute Range
Back light setting	0 .. 255/Unknown	Set or Read

Camera White Balance Alignment

This attribute allows a remote RDC application to set or read the camera white balance alignment.

Attribute Fields	Values	Field Type
Maximum manual white balance setting	1 .. 255	Attribute Range
White Balance setting	0 .. 255/Unknown	Set or Read

Camera White Balance Alignment Mode

This attribute allows the camera white balance alignment mode to be set or read.

Attribute Fields	Values	Field Type
Mode	Manual/Auto/Unknown	Set or Read

Calibrate White Balance Alignment

This attribute, when used as part of the RDC-Device-Control request primitive, instructs the camera to calibrate the white balance alignment. Once the camera has aligned the white balance it will maintain that alignment regardless of changes occurring in the current image. It has no attribute fields.

Focus Image

This attribute, when used as part of the RDC-Device-Control request primitive, instructs the camera to bring the current image into focus. Once the camera has focused it will maintain that focus position regardless of changes occurring in the current image. It has no attribute fields.

Capture Image

This attribute, when used as part of the RDC-Device-Control request primitive, instructs the camera to capture the current image and transmit it. Once the captured image has been transmitted the camera will revert to sending motion video. This attribute has no attribute fields.

A.2.2 Camera Control Attributes Used For Start Stop Movement

These attributes start a continuous movement and have a maximum time-out as part of the attribute. The continuous movement is started in the selected direction until the time-out expires. If another control service primitive is received before the time-out expires, the movement will be stopped immediately if the primitive contains the same attribute with a stop parameter. If the primitive contains the same attribute with a direction parameter (for example left, up or telescopic) or a continue parameter with a new time-out, the timer will immediately be restarted with the new time-out. A range from 50 to 1000 in steps of 50 is used to represent the time-out in milliseconds. The requesting node should use time-out values greater than 250 ms when continuous movement is required to avoid the need to send a large number of requests. If the time-out expires, the receiving node shall act as though it received a primitive containing the attribute with a stop parameter. If the node receives any subsequent primitive containing the attribute with a continue or stop parameter, then this request shall be discarded until a primitive containing the attribute with a direction parameter is received.

A device of the camera class that supports pan, tilt, zoom, or focus control shall support it with the continuous movement control attributes. Optionally, such devices may support the fixed range or spatial positioning or relative field of view control attributes.

Specifically, if a device supports the control attribute Pan Position or Pan View, then it shall also support the control attribute Pan Continuous. If a device supports the control attribute Tilt Position or Tilt View, then it shall also support the control attribute Tilt Continuous. If a device supports the control attribute Zoom Position or Zoom Magnification, then it shall also support the control attribute Zoom Continuous. If a device supports the control attribute Focus Position, then it shall also support the control attribute Focus Continuous.

When any continuous movement control attribute is sent as part of a RDC-Device-Control request primitive the User Application may request that this primitive is sent via a low latency data channel if such a channel is available.

Pan Continuous

Starts or stops a continuous horizontal panning movement with a maximum time-out.

Attribute Fields	Values	Field Type
Pan direction	Left/Right/Stop/Continue	Set Only
Pan time-out	50 .. 1000 ms	Set Only

Tilt Continuous

Starts or stops a continuous vertical tilting movement with a maximum time-out.

Attribute Fields	Values	Field Type
Tilt direction	Up/Down/Stop/Continue	Set Only
Tilt time-out	50 .. 1000 ms	Set Only

Zoom Continuous

Starts or stops a continuous zoom movement with a maximum time-out.

Attribute Fields	Values	Field Type
Zoom direction	Telescopic/Wide/Stop/Continue	Set Only
Zoom time-out	50 .. 1000 ms	Set Only

Focus Continuous

Starts or stops a continuous focus movement with a maximum time-out.

Attribute Fields	Values	Field Type
Focus direction	Near/Far/Stop/Continue	Set Only
Focus time-out	50 .. 1000 ms	Set Only

Iris Continuous

Starts or stops a continuous iris movement with a maximum time-out.

Attribute Fields	Values	Field Type
Iris direction	Darker/Lighter/Stop/Continue	Set Only
Iris time-out	50 .. 1000 ms	Set Only

A.2.3 Camera Control Attributes Using Fixed Range Positioning

This group of attributes uses a position mechanism with fixed ranges. The receiving RDC application must map this fixed positioning to the actual positioning system used by the camera. For example, the zoom attributes have a fixed coordinate range of –1023 to +1023. Zoom position +1023 represents the maximum zoom position. Zoom position –1023 is the maximum unzoomed position and 0 represents the mid point. A minimum step size is declared which indicates the increments that are applied to the range in both the positive and negative direction starting from 0 to calculate the coordinate points supported by this device. This method is not used for panning and tilting.

A fixed range value maps to the following camera positions depending upon the attribute with which it is used.

Maximum negative value	0	Maximum positive value
Maximum unzoomed	Midpoint Zoom position	Maximum Zoom
Maximum Focus Near	Midpoint Focus position	Maximum Focus Far

Zoom Position

This attribute is used to set or read the camera zoom position.

Attribute Fields	Values	Field Type
Minimum step size	1 .. 1023	Attribute Range
Zoom position	–1023 .. +1023	Set or Read
Positioning mode	Relative/Absolute	Set Only

Focus Position

This attribute is used to set or read the camera focus position.

Attribute Fields	Values	Field Type
Minimum step size	1 .. 127	Attribute Range
Focus position	–127 .. +127	Set or Read
Positioning mode	Relative/Absolute	Set Only

Iris Position

This attribute is used to set or read the camera iris position.

Attribute Fields	Values	Field Type
Minimum step size	1 .. 127	Attribute Range
Iris position	–127 .. +127	Set or Read
Positioning mode	Relative/Absolute	Set Only

A.2.4 Camera Control Attributes Using Spatial Positioning

This group of attributes uses a positioning mechanism where the positioning system represents the absolute camera position in degrees. The receiving RDC application must map these absolute degree positions to the camera positioning system.

These parameter values express the angle in degrees to which the camera is to be positioned assuming that the mid point of the camera pan and tilt range is 0°.

An absolute position value maps to the following camera positions depending upon the attribute with which it is used.

Negative value	0°	Positive value
Number of degrees to pan to the left of the midpoint.	Midpoint Pan position	Number of degrees to pan to the right of the midpoint.
Number of degrees to Tilt downwards from the midpoint.	Midpoint Tilt position	Number of degrees to Tilt upwards from the midpoint.

A range from –18000 to +18000 is used to represent a position from –180° to +180° in steps of 0.01° (= 0° 0' 36"). A minimum step size is declared which indicates the increments that are applied to the range in both the positive and negative direction starting from 0 to calculate the values supported by this device. The camera should position itself to the nearest position possible.

A device of the camera class that supports the Zoom Position attribute shall also support the Zoom Magnification attribute. If a device supports the Pan Position attribute it shall also support the Pan View attribute. If a device supports the Tilt Position attribute it shall also support the Tilt View attribute.

Pan Position

This attribute is used to set or read the camera pan position.

Attribute Fields	Values	Field Type
Max. pan range Left	–18000 .. 0	Attribute Range
Max. pan range right	0 .. +18000	Attribute Range
Min camera step size	1 .. +18000	Attribute Range
Pan position	–18000 .. +18000	Set or Read
Positioning mode	Relative/Absolute	Set Only

Tilt Position

This attribute is used to set or read the camera tilt position.

Attribute Fields	Values	Field Type
Max. tilt range Down	–18000 .. 0	Attribute Range
Max. tilt range Up	0 .. +18000	Attribute Range
Min camera step size	1 .. +18000	Attribute Range
Tilt position	–18000 .. +18000	Set or Read
Positioning mode	Relative/Absolute	Set Only

A.2.5 Camera Control Attributes Using the Current Field of View

This group of attributes allow a camera to be moved relative to the current field of view.

A device of the camera class that supports the Zoom Magnification attribute shall also support the Zoom Position attribute. If a device supports the Pan View attribute it shall also support the Pan Position attribute. If a device supports the Tilt View attribute it shall also support the Tilt Position attribute.

Zoom Magnification

This attribute is used to set the camera zoom position relative to the current field of view.

Attribute Fields	Values	Field Type
Minimum step size	1 .. 1000	Attribute Range
Zoom Magnification	+10 .. +1000	Set Only

The Zoom Magnification value represents the percentage (linearly, not in area) of the current field of view that should become the new field of view. For example, the value 20 means to zoom in by a factor of 5, such that 20% of the current field of view becomes the new field of view. A value of 100 means do not change the zoom setting. A value of 200 means zoom out by a factor of 2, such that the new field of view is twice as large as the old field of view in both horizontal and vertical directions.

If a Zoom Magnification value attempts to change the position beyond the range of the device, then the device shall move to the limit.

Pan View

This attribute is used to set the camera pan position relative to the current field of view.

Attribute Fields	Values	Field Type
Pan view	–1000 .. +1000	Set

The pan view value represents the percentage of the current field of view the camera should move, in steps of tenth of a per cent. For example, the value +500 means move right 50% of the current field of view. A negative value means move left.

If the pan view value attempts to move the camera beyond the range of the device, the camera will move to the limit. It is a local issue how the camera will respond if the pan view value attempts to move the camera using a smaller percentage value than the camera can support for the current field of view.

Tilt View

This attribute is used to set the camera tilt position relative to the current field of view.

Attribute Fields	Values	Field Type
Tilt view	–1000 .. +1000	Set

The tilt view value represents the percentage of the current field of view the camera should move, in steps of tenth of a per cent. For example, the value +500 means move up 50% of the current field of view. A negative value means move down.

If the tilt view value attempts to move the camera beyond the range of the device, the camera will move to the limit. It is a local issue how the camera will respond if the tilt view value attempts to move the camera using a smaller percentage value than the camera can support for the current field of view.

A.2.6 Slide Presenter Control Attributes

Select Slide

This attribute allows the selection of a particular slide for viewing. This attribute may also be read to determine which slide is currently selected for viewing.

Attribute Fields	Values	Field Type
Maximum number available	1 .. 1024	Attribute Range
Slide number	0 .. 1023/Unknown	Set or Read

Select Next Slide

This attribute allows the selection of a slide for viewing adjacent to the currently selected slide.

Attribute Fields	Values	Field Type
Direction	Forwards/Backwards	Set Only

Slide Show Mode

This attribute allows the slide presenter mode to be either set or read.

Attribute Fields	Values	Field Type
Mode	Manual/Auto/Unknown	Set or Read

Play Automatic Slide Show

This attribute starts, stops or pauses an automatic slide show.

Attribute Fields	Values	Field Type
Mode	Start/Stop/Pause	Set Only

Set Automatic Slide Display Time

This attribute allows the time period in seconds for which a slide is displayed when an automatic slide show is in progress to be set or read.

Attribute Fields	Values	Field Type
Maximum slide time available	1 .. 255	Attribute Range
Slide display time in seconds	1 .. 255/Unknown	Set or Read

A.2.7 Stream Player Recorder Control Attributes for Stream Positioning

Continuous Rewind

This attribute is used to start or stop a rewinding operation. The output stream is not visible during rewinding.

Attribute Fields	Values	Field Type
Mode	Start/Stop	Set Only

Continuous Fast Forward

This attribute is used to start or stop a fast forwarding operation. The output stream is not visible during fast forwarding.

Attribute Fields	Values	Field Type
Mode	Start/Stop	Set Only

Search Backwards

This attribute is used to start or stop reversing at greater than normal playing speed through the program whilst still showing the output.

Attribute Fields	Values	Field Type
Mode	Start/Stop	Set Only

Search Forwards

This attribute is used to start or stop forwarding at greater than normal playing speed through the program whilst still showing the output.

Attribute Fields	Values	Field Type
Mode	Start/Stop	Set Only

Pause

This attribute sets or releases the pause function.

Attribute Fields	Values	Field Type
Mode	Pause/Release	Set Only

Select Program

This attribute selects the current program. It can also be read to determine the current program selection.

Attribute Fields	Values	Field Type
Maximum number of programs available	1 .. 1023	Attribute Range
Select	1 .. 1023/Unknown	Set or Read

Next Program

This attribute selects an adjacent program to the current selected program.

Attribute Fields	Values	Field Type
Select	next/previous	Set Only

Go To Normal Play Time Point

This instructs the player to move to this Normal Play Time point within the current program. The Normal Play Time is measured from the start of the program and is expressed in hours, minutes,

seconds and microseconds. The player should position itself to the nearest play time point position possible.

Attribute Fields	Values	Field Type
Normal play time point hours	0 .. 24	Set Only
Normal play time point minutes	0 .. 59	Set Only
Normal play time point seconds	0 .. 59	Set Only
Normal play time point microseconds	0 .. 99999	Set Only

Read Stream Player Recorder State

This attribute returns the current state of the stream player recorder.

Attribute Fields	Values	Field Type
State	Playing/Recording/ Paused on record/ Paused on play/ Rewinding/Fast Forwarding/ Searching Forwards/ Searching Backwards/ Stopped/ Program Unavailable/ Unknown	Read Only

Read Current Program Duration

This attribute returns the total duration of the currently active program in hours, minutes, seconds and microseconds. A value of 0 in all fields indicates that the duration could not be determined.

Attribute Fields	Values	Field Type
Total duration hours	0 .. 23	Read only
Total duration minutes	0 .. 59	Read only
Total duration seconds	0 .. 59	Read only
Total duration microseconds	0 .. 99999	Read only

A.2.8 Stream Player Recorder Attributes for Configuration

Continuous Playback Mode

This attribute configures the setting of the continuous play back mode. It may also be read to determine the current playback mode setting.

Attribute Fields	Values	Field Type
Mode	On/Off/Unknown	Set or Read

Playback Speed

This attribute allows the playback speed to be set or read. In order to set a new play speed, a remote RDC application will set the Playback speed scale factor field and set the Scale factor operator field to indicate whether the normal playback speed is multiplied or divided by this factor. A range of 10

to +1000 is used to represent scale factor values from 1.0 to 100.0 in steps of 0.1. A scale factor value of 1 results in normal playback speed. Two lists are also declared by this attribute. The multiplier factors list declares those factor values which may be used to multiply the playback speed. Similarly the divisor factors list declares those factors which may be used to divide the playback speed. Each list is limited to 64 factor values. The playback speed scale factor should only be set to one of these declared values.

Attribute Fields	Values	Field Type
Number of multiplier factors	1 .. 64	Attribute Range
List of multiplier factors	List of up to 64 factors	Attribute Range
Number of divisor factors	1 .. 64	Attribute Range
List of divisor factors	List of up to 64 factors	Attribute Range
Playback speed scale factor	10 .. 1000/Unknown	Set or Read
Scale factor operator	Multiply/Divide/Unknown	Set or Read

A.2.9 Stream Player Recorder Attributes for Playback

Play

This attribute starts or stops stream playback.

Attribute Fields	Values	Field Type
Mode	Start/Stop	Set Only

Set Audio Output State

This attribute activates or mutes the audio output from the player.

Attribute Fields	Values	Field Type
State	Active/Muted/Unknown	Set or Read

Play to Normal Play Time Point

This instructs the player to start playing until the Normal Play Time point within the current program is reached. The Normal Play Time is measured from the start of the program and is expressed in seconds and microseconds.

Attribute Fields	Values	Field Type
Normal play time point hours	0 .. 24	Set Only
Normal play time point minutes	0 .. 59	Set Only
Normal play time point seconds	0 .. 59	Set Only
Normal play time point microseconds	0 .. 99999	Set Only

A.2.10 Stream Player Recorder Control Attributes for Recording

These attributes may be used by a device that is capable of recording an incoming stream.

Record

This attribute starts or stops recording an incoming stream.

Attribute Fields	Values	Field Type
Mode	Start/Stop	Set Only

Record for Duration

This attribute starts or stops recording an incoming stream for a specified duration which is expressed in hours, minutes and seconds.

Attribute Fields	Values	Field Type
Hours	0 .. 24	Set Only
Minutes	0 .. 59	Set Only
Seconds	0 .. 59	Set Only

A.2.11 Source Combiner Control Attributes

The attributes in this group are designed to be used with the Source Combiner device class. This device class models a device that has 2 or more input streams which are combined to produce only 1 output stream. A fixed number of inputs are declared to which other devices are attached.

A source combiner may allow a remote RDC application to specify the attachment of a particular device to a particular input. In this case a list of the available devices for attachment will be declared.

Configurable Video Device Inputs

This attribute allows a remote RDC application to configure the list of devices that will source a video input into the source combiner device. The attribute declares the number of inputs available for this source combiner. It also declares a list of available devices which can be attached to an input. This list may exceed the actual number of available inputs. The device input list may be set by a remote RDC application to specify a particular device to be attached to a particular input. This list should not exceed the number of available inputs. If the number of entries in this list is less than the number of available inputs, then some inputs will be unconnected.

Attribute Fields	Values	Field Type
Number of device inputs	2 .. 64	Attribute Range
Number of device rows	1 .. 64	Attribute Range
Number of entries in available devices list	2 .. 64	Attribute Range
List of available devices	List of device classes and identifiers	Attribute Range
Number of entries in device input list	2 .. 64	Set or Read
Device input list	List of device classes and identifiers	Set or Read

The resultant output stream can be considered to be a matrix of combined devices. The "Number of device inputs" attribute field declares the number of cells in the matrix and the "Number of device rows" declares the number of rows. The matrix numbering starts at the top left-hand corner and moves towards the right along the row and then down the matrix starting at the left-hand end of the next row.

For example, a device that declares 9 devices with 3 device rows would produce a 3 × 3 output matrix. This matrix defines how the combined sources will be ordered in the video stream as shown in the diagram. The exact format used to display these devices at the remote end will depend upon the remote terminal.

Input 1	Input 2	Input 3
Input 4	Input 5	Input 6
Input 7	Input 8	Input 9

The "Device input list" is used to associate devices with inputs. The first entry in the list is connected to input 1, the second entry to input 2 and so on. This list may be set or read. Only those devices previously declared in the "List of available devices" attribute field can be used in this list. If there are less entries in the device list than number of inputs then the remaining inputs are undefined. The number of entries in this list should not exceed the number of inputs.

Video Device Inputs

This attribute is used when the source combiner supports video inputs which are combined to form 1 video output. This attribute is similar to the Configure Video Device Inputs attribute except that it does not allow a remote RDC application to configure the list of devices that are attached to inputs. The Device input list may only be read by a remote RDC application.

Attribute Fields	Values	Field Type
Number of device inputs	2 .. 64	Attribute Range
Number of device rows	1 .. 64	Attribute Range
Number of entries in device input list	2 .. 64	Read Only
Device input list	List of device classes and identifiers	Read Only

Configurable Audio Device Inputs

This attribute allows a remote RDC application to configure the list of devices that will source an audio input into the source combiner device. The attribute declares the number of device inputs available for this source combiner. It also declares a list of available devices which can be attached to an input. This list may exceed the actual number of available inputs. The device input list may be set by a remote RDC application to specify the list of devices that are to be attached. This list should not exceed the number of device inputs. If the number of entries in this list is less than the number of device inputs, some inputs will be unconnected.

Attribute Fields	Values	Field Type
Number of device inputs	2 .. 64	Attribute Range
Number of entries in available devices list	2 .. 64	Attribute Range
List of available devices	List of device classes and identifiers	Attribute Range
Number of entries in device input list	2 .. 64	Set or Read
Device input list	List of device classes and identifiers	Set or Read

Audio Device Inputs

This attribute is used when the source combiner supports audio inputs which are combined to form 1 audio output. This attribute is similar to the Configure Audio Device Inputs attribute except that it does not allow a remote RDC application to configure the list of devices that are attached to inputs. The Device input list may only be read by a remote RDC application.

Attribute Fields	Values	Field Type
Number of device inputs	2 .. 64	Attribute Range
Number of entries in device input list	2 .. 64	Read Only
Device input list	List of device classes and identifiers	Read Only

A.3 Device Event Attributes

These event attributes are general attributes for use by any device class as appropriate.

Device Lock State Changed

This event attribute signals a change in the device lock state. The new state field indicates the new lock state of the device.

Attribute Fields	Values	Field Type
New state	Locked/Unlocked	Event notification

Device Availability Changed

This event attribute signals a temporary change in the availability of the device. The new state field indicates the new availability of the device. This may be because the device has been made inactive or is no longer communicating with the host node. If it remains unavailable for a long period of time, or a device is removed permanently, the device profile should be removed.

Attribute Fields	Values	Field Type
New state	Unavailable/Available	Event notification

A.3.1 Camera Event Attributes

The event attributes in this group are designed for use with a camera.

Camera Panned to Limit

This event attribute signals that the camera reached the limit of the available pan range. The direction field indicates which pan limit was reached.

Attribute Fields	Values	Field Type
Direction	Left/Right	Event notification

Camera Tilted to Limit

This event attribute signals that the camera reached the limit of the available tilt range. The direction field indicates which tilt limit was reached.

Attribute Fields	Values	Field Type
Direction	Up/Down	Event notification

Camera Zoomed to Limit

This event attribute signals that the camera reached the limit of the available zoom range. The direction field indicates which zoom limit was reached.

Attribute Fields	Values	Field Type
Direction	Telescopic/wide	Event notification

Camera Focused to Limit

This event attribute signals that the camera reached the limit of the available focus range. The direction field indicates which focus limit was reached.

Attribute Fields	Values	Field Type
Direction	Near/far	Event notification

A.3.2 Slide Presenter Event Attributes

The event attributes in this group may be used with a slide presenter.

Automatic Slide Show Finished

This event attribute signals that the automatic slide show has reached the end of the slides available for viewing.

A.3.3 Stream Player Recorder Event Attributes

The event attributes in this group may be used with a stream player recorder.

Stream Player Recorder State Change

This event attribute signals that the player/recorder has changed state. For example, a state change will occur when a local user starts playing a VCR. The VCR device will then inform interested parties that the VCR is now playing.

Attribute Fields	Values	Field Type
New State	Playing/ Recording/ Paused on record/ Paused on play/ Rewinding/ Fast Forwarding/ Searching Forwards/ Searching Backwards/ Stopped/ Program Unavailable	Event notification

Stream Player Recorder Program Change

This event attribute signals that the player/recorder has changed the selected program. If a program is no longer selected then the New program field will indicate 0.

Attribute Fields	Values	Field Type
New program	0 .. 1023	Event notification

ITU-T RECOMMENDATIONS SERIES

Series A	Organization of the work of the ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	**Audiovisual and multimedia systems**
Series I	Integrated services digital network
Series J	Transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	Global information infrastructure and Internet protocol aspects
Series Z	Languages and general software aspects for telecommunication systems