DigitalSecurityCameraSettings:1 Service Template Version 1.01

For UPnP™ Version 1.0
Status: Standardized DCP
Date: March 24, 2005

This Standardized DCP has been adopted as a Standardized DCP by the Steering Committee of the UPnP™ Forum, pursuant to Section 2.1(c)(ii) of the UPnP™ Forum Membership Agreement. UPnP™ Forum Members have rights and licenses defined by Section 3 of the UPnP™ Forum Membership Agreement to use and reproduce the Standardized DCP in UPnP™ Compliant Devices. All such use is subject to all of the provisions of the UPnP™ Forum Membership Agreement.

Authors	Company
Joacim Tullberg	Axis Communications AB
Ted Hartzell	Axis Communications AB
Göran Haraldsson	Axis Communications AB

Contents

List of Tables

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version *1.0*.

This service provides *control* of the basic setting of the actual image generating part of the security camera. Security Camera Settings are global to the device and will affect the Security Camera Still and Motion Image services contained within the Security Camera device.

2. Service Modeling Definitions

2.1. ServiceType

A service that is compliant with this template is identified with the following service type: **urn:schemas-upnp-org:service:** *DigitalSecurityCameraSettings:1*.

2.2. State Variables

Table 1: State Variables

Variable Name	Req. or Opt. [1]	Data Type	Allowed Value	Default Value	Eng. Units
AutomaticWhiteBalance	*O*	*boolean*	*1,0*	*1*	
FixedWhiteBalance	*R*	*ui4*	*[0..]*	*3000*	*K*
AvailableRotations	*O*	*string*			
DefaultRotation	*O*	*string*			
Brightness	*R*	*ui1*	*0..100*	*50*	*%*
ColorSaturation	*R*	*ui1*	*0..100*	*50*	*%*
Non-standard state variables implemented by an UPnP vendor go here.	X	*TBD*	*TBD*	*TBD*	*TBD*

[1] R = Required, O = Optional, X = Non-standard.

2.2.1. AutomaticWhiteBalance

White Balance describes what the eye perceives as white. This differs with ambient light. For instance an object does not have the same color if it is placed in a room lit by light bulbs which gives a rather red light as it would if the object was placed outdoors in bright sunlight (rather blue). The human eye automatically compensates for this difference in perceived color. Cameras must in a similar manner compensate for ambient light to avoid objects being reproduced with incorrect colors.

If automatic is ON white balance should be handled automatically by the camera.

If no automatic white balance is available in the device fixed white balance value shall be used.

2.2.2. FixedWhiteBalance

Fixed White balance is used in environments of known or constant light environments. The unit used is the temperature of the light in Kelvin.

2.2.3. AvailableRotations

The rotation in degrees of the image calculated from what is considered normal upright positioning (0) of the camera.

The value of this string should be a comma delimited list of all supported rotations.

The list order should be the lowest compression level first.

Example: 0, 90, 180, 270

This is a read only property.

2.2.4. DefaultRotation

The value of this string must be one of the supported rotations specified in AvailableRotations.

2.2.5. Brightness

Brightness is the attribute of a visual sensation according to which an area appears to emit more or less light. It is a relative value and the definition of the normal value is left to the manufacturer.

The Value 50 is the device specific normal value. This single value must be supported.

The value 100 shall give maximum brightness.

The value 0 shall give maximum darkness.

If the full range 0..100 is unsupported, the legal values shall be a sequence of numbers appropriately mapped onto the 1..100 scale. For instance: 2, 34,45,50,74,87.

The Actions up and down shall make a complete step to the next supported legal value.

The response to an attempt to set an unsupported but legal value shall result in the nearest *supported* legal value.

2.2.6. ColorSaturation

Color Saturation describes the colorfulness of an area judged in proportion to its brightness.

At least the value 0 or the value 50 must be supported.

0 means Black and white and is the only value that must be supported by black and white cameras.

50 means device specific normal color, if no color control is available in a color camera this value is the only supported legal value.

The value 100 means very colorful images

The value 0 means black and white images only.

If the full range 0..100 is unsupported, the legal values shall be a sequence of numbers appropriately mapped onto the 1..100 scale. For instance: 2, 34,45,50,74,87.

The Actions up and down shall make a complete step to the next supported legal value.

The response to an attempt to set an unsupported but legal value shall result in the nearest *supported* legal value.

2.3. Eventing and Moderation

Table 2: Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate [1]	Logical Combination	Min Delta per Event [2]
AutomaticWhiteBalance	*Yes*	*Yes*	*1*		*None*
FixedWhiteBalance	*Yes*	*Yes*	*1*		*None*
AvailableRotations	*No*	*No*	*N/A*		*N/A*
DefaultRotation	*Yes*	*Yes*	*1*		*None*
Brightness	*Yes*	*Yes*	*1*		*None*
ColorSaturation	*Yes*	*Yes*	*1*		*None*
Non-standard state variables implemented by an UPnP vendor go here.	*TBD*	*TBD*	*TBD*	*TBD*	*TBD*

[1] Determined by N, where Rate = (Event)/(N secs).
[2] (N) * (allowedValueRange Step).

2.4. Actions

Immediately following this table is detailed information about these actions, including short descriptions of the actions, the effects of the actions on state variables, and error codes defined by the actions.

Table 3: Actions

Name	Req. or Opt. [1]
SetAutomaticWhiteBalance	*O*
GetAutomaticWhiteBalance	*O*
SetFixedWhiteBalance	*R*
GetFixedWhiteBalance	*R*
GetAvailableRotations	*O*
SetRotation	*O*
GetRotation	*O*
SetBrightness	*R*
GetBrightness	*R*
IncreaseBrightness	*R*
DecreaseBrightness	*R*
SetColorSaturation	*R*
GetColorSaturation	*R*
IncreaseColorSaturation	*R*
DecreaseColorSaturation	*R*
Non-standard actions implemented by an UPnP vendor go here.	X

[1] R = Required, O = Optional, X = Non-standard.

2.4.1. SetAutomaticWhiteBalance

Enable or disable Automatic White Balance algorithm. When disabled the current fixed white balance value will be used.

2.4.1.1. Arguments

Table 4: Arguments for *SetAutomaticWhiteBalance*

Argument	Direction	relatedStateVariable
NewAutomaticWhiteBalance	*IN*	*AutomaticWhiteBalance*

2.4.1.2. Effect on State

Changes the value of AutomaticWhiteBalance to NewAutomaticWhiteBalance.

2.4.2. GetAutomaticWhiteBalance

Get AutomaticWhiteBalance.

2.4.2.1. Arguments

Table 5: Arguments for *GetAutomaticWhiteBalance*

Argument	Direction	relatedStateVariable
RetAutomaticWhiteBalance	*OUT*	*AutomaticWhiteBalance*

2.4.3. SetFixedWhiteBalance

Set the value of the FixedWhiteBalance to NewFixedWhiteBalance.

2.4.3.1. Arguments

Table 6: Arguments for *SetFixedWhiteBalance*

Argument	Direction	relatedStateVariable
NewFixedWhiteBalance	*IN*	*FixedWhiteBalance*

2.4.3.2. Effect on State

Changes the value of FixedWhiteBalance to NewFixedWhiteBalance.
AutomaticWhiteBalance is set to false.

2.4.4. GetFixedWhiteBalance

Get FixedWhiteBalance.

2.4.4.1. Arguments

Table 7: Arguments for *GetFixedWhiteBalance*

Argument	Direction	relatedStateVariable
RetFixedWhiteBalance	*OUT*	*FixedWhiteBalance*

2.4.5. GetAvailableRotations

Get the list of supported rotations.

2.4.5.1. Arguments

Table 8: Arguments for *GetAvailableRotations*

Argument	Direction	relatedStateVariable
RetAvailableRotations	*OUT*	*AvailableRotations*

2.4.6. SetDefaultRotation

Set the default rotation of the image.

2.4.6.1. Arguments

Table 9: Arguments for *SetDefaultRotation*

Argument	Direction	relatedStateVariable
NewRotation	*IN*	*DefaultRotation*

2.4.6.2. Effect on State

Changes the value of DefaultRotation to NewRotation.

2.4.6.3. Errors

errorCode	errorDescription	Description
700	*NewRotation not supported*	NewRotation is not one of the supported rotations specified in AvailableRotations.

2.4.7. GetDefaultRotation

Get the default rotation.

2.4.7.1. Arguments

Table 10: Arguments for *GetDefaultRotation*

Argument	Direction	relatedStateVariable
RetRotation	*OUT*	*DefaultRotation*

2.4.8. SetBrightness

Set the value of the target brightness of the image. If set to a legal but by the device unsupported value the nearest supported value shall be set.

2.4.8.1. Arguments

Table 11: Arguments for *SetBrightness*

Argument	Direction	relatedStateVariable
NewBrightness	*IN*	*Brightness*

2.4.8.2. Effect on State

Changes the value of Brightness to NewBrightness.

2.4.8.3. Errors

errorCode	errorDescription	Description
701	*NewBrightness not supported*	NewBrightness is not a supported value.

2.4.9. GetBrightness

Get the brightness.

2.4.9.1. Arguments

Table 12: Arguments for *GetBrightness*

Argument	Direction	relatedStateVariable
RetBrightness	*OUT*	*Brightness*

2.4.10.IncreaseBrightness

Increase brightness of the image to the nearest higher *supported* legal value.

2.4.10.1.Arguments

(None.)

2.4.10.2.Effect on State

The value of Brightness is being increased.

2.4.10.3.Errors

errorCode	errorDescription	Description
701	NewBrightness not supported	NewBrightness is not a supported value.

2.4.11.DecreaseBrightness

Decrease brightness of the image to the nearest lower *supported* legal value.

2.4.11.1.Arguments

(None.)

2.4.11.2.Effect on State

The value of Brightness is being decreased.

2.4.11.3.Errors

errorCode	errorDescription	Description
701	NewBrightness not supported	NewBrightness is not a supported value.

2.4.12.SetColorSaturation

Set the value of the target Color Saturation of the image. If set to a legal but by the device unsupported value the nearest supported value shall be set.

2.4.12.1.Arguments

Table 13: Arguments for *SetColorSaturation*

Argument	Direction	relatedStateVariable
NewColorSaturation	IN	ColorSaturation

2.4.12.2.Effect on State

The value of ColorSaturation is changed to NewColorSaturation.

2.4.12.3.Errors

errorCode	errorDescription	Description
702	NewColorSaturation not supported	NewColorSaturation is not a supported value.

2.4.13.GetColorSaturation

Get the Color Saturation.

2.4.13.1.Arguments

Table 14: Arguments for *GetColorSaturation*

Argument	Direction	relatedStateVariable
RetColorSaturation	*OUT*	*ColorSaturation*

2.4.14.IncreaseColorSaturation

Increase ColorSaturation of the image to the nearest higher *supported* legal value.

2.4.14.1.Arguments

(None.)

2.4.14.2.Effect on State

The value of ColorSaturation is being increased.

2.4.14.3.Errors

errorCode	errorDescription	Description
702	*NewColorSaturation not supported*	NewColorSaturation is not a supported value.

2.4.15.DecreaseColorSaturation

Decrease brightness of the image to the nearest lower *supported* legal value.

2.4.15.1.Arguments

(None.)

2.4.15.2.Effect on State

The value of ColorSaturation is being decreased.

2.4.15.3.Errors

errorCode	errorDescription	Description
702	*NewColorSaturation not supported*	NewColorSaturation is not a supported value.

2.4.16.Non-Standard Actions Implemented by a UPnP Vendor

To facilitate certification, non-standard actions implemented by UPnP vendors should be included in this service template. The UPnP Device Architecture lists naming requirements for non-standard actions (see the section on Description).

2.4.17.Common Error Codes

The following table lists error codes common to actions for this service type. If an action results in multiple errors, the most specific error should be returned.

Table 15: Common Error Codes

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
404	Invalid Var	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
600-699	TBD	Common action errors. Defined by UPnP Forum Technical Committee.
700	*NewRotation not supported*	NewRotation is not one of the supported rotations specified in AvailableRotations.
701	*NewBrightness not supported*	NewBrightness is not a supported value.
702	*NewColorSaturation not supported*	NewColorSaturation is not a supported value.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.5. Theory of Operation

An instance of Digital Security Camera Settings Services may be embedded into a Digital Security Camera Device or other devices requiring this service.

This service provides control of basic settings of the actual image generating part of for instance a security camera. Settings are global to the device and will affect images generated by the device regardless of retrieval method. For instance if a device is equipped with an instance of the *Digital Security Camera Motion Image Service* and an instance of the *Digital Security Camera Still Image Service*, images "retrieved" using either of these services will all be affected by changes in the here described *Settings Service*

The algorithm and functionality of *Au*

3. XML Service Description

```
<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action>
    <name>SetAutomaticWhiteBalance</name>
      <argumentList>
        <argument>
          <name>NewAutomaticWhiteBalance</name>
          <relatedStateVariable>AutomaticWhiteBalance
          </relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetAutomaticWhiteBalance</name>
      <argumentList>
        <argument>
          <name>RetAutomaticWhiteBalance</name>
          <relatedStateVariable>AutomaticWhiteBalance
          </relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetFixedWhiteBalance</name>
      <argumentList>
        <argument>
          <name>NewFixedWhiteBalance</name>
          <relatedStateVariable>FixedWhiteBalance
          </relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetFixedWhiteBalance</name>
      <argumentList>
        <argument>
          <name>RetFixedWhiteBalance</name>
          <relatedStateVariable>FixedWhiteBalance
          </relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetAvailableRotations</name>
```

```
      <argumentList>
        <argument>
          <name>RetAvailableRotations</name>
         <relatedStateVariable>AvailableRotations</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
   <action>
    <name>SetDefaultRotation</name
```

```
      <argumentList>
        <argument>
          <name>NewColorSaturation</name>
         <relatedStateVariable>ColorSaturation</relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetColorSaturation</name>
      <argumentList>
        <argument>
          <name>RetColorSaturation</name>
         <relatedStateVariable>ColorSaturation</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>IncreaseColorSaturation</name>
    </action>
    <action>
    <name>DecreaseColorSaturation</name>
    </action>
  </actionList>
  <serviceStateTable>
    <stateVariable sendEvents="yes">
      <name>AutomaticWhiteBalance</name>
      <dataType>boolean</dataType>
      <defaultValue>1</defaultValue>
      <allowedValueList>
        <allowedValue>1</allowedValue>
        <allowedValue>0</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>FixedWhiteBalance</name>
      <dataType>ui4</dataType>
      <defaultValue>3000</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>maximum</maximum>
        <step>1</step>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>AvailableRotations</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>DefaultRotation</name>
      <dataType>string</dataType>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>Brightness</name>
      <dataType>ui1</dataType>
```

```
      <defaultValue>50</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>100</maximum>
        <step>1</step>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>ColorSaturation</name>
      <dataType>ui1</dataType>
      <defaultValue>50</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>100</maximum>
        <step>1</step>
      </allowedValueRange>
    </stateVariable>
  </serviceStateTable>
</scpd>
```

4. Test

Testing of the UPnP functions Addressing, Discovery, Description, Control (Syntax) and Eventing are performed by the UPnP Test Tool v1.1 based on the following documents:

- UPnP Device Architecture v1.0
- The Service Definitions in chapter 2 of this document
- The XML Service Description in chapter 3 of this document
- The UPnP Test Tool service template test file: ***DigitalSecurityCameraSettings1.xml***
- The UPnP Test Tool service template test file: ***DigitalSecurityCameraSettings1.SyntaxTests.xml***

The test suite does not include tests for Control Semantics, since it is felt that such tests would not provide a higher level of interoperability.