Dimming:1 Service Template Version 1.01

For UPnP™ Version 1.0
Status: Standardized DCP
Date: November 23, 2003

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
Christoph Sahm	EIBA s.c.
Scott Manchester	Panja
Hans Langels	Siemens
Scott Manchester	Microsoft

Contents

List of Tables

1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version *1.0* and Version 1.01 of the UPnP Standard Service Template.

This service-type enables the following functions:

1. Enables remote control of a dimmable lighting device,
2. Defines required basic actions for dimming to a value,
3. Defines optional actions for the dimming effect when the dimmable light is turned on,
4. Defines optional actions for stepped dimming,
5. Defines optional actions for ramped dimming (time base ramping or target value ramping).

2. Service Modeling Definitions

2.1. ServiceType

The following service type identifies a service that is compliant with this template:

urn:schemas-upnp-org:service:*Dimming:1*.

2.2. State Variables

Table 2-1: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [2]	Default Value [2]	Eng. Units
LoadLevelTarget	R	ui1	0-100	0	%
LoadLevelStatus	R	ui1	0-100	0	%
OnEffectLevel	O	ui1	0-100	100	%
OnEffect	O	string	OnEffectLevel, LastSetting, Default	Default	
StepDelta	O	ui1	1-100	See Table 1.1 below	%
RampRate	O	ui1	0-100	0	%/s
IsRamping	O	Boolean		0	
RampPaused	O	Boolean		0	
RampTime	O	ui4	0-4294967295	0	ms
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] Values listed in these columns are required. To specify standard optional values or to delegate assignment of values to the vendor, you must reference a specific instance of an appropriate table below.

Table 2-2: DefaultValue for StepDelta

Value	Req. or Opt.
Vendor-defined within range of 1-100	R

2.2.1. LoadLevelTarget

Value of the desired load level, LoadLevelStatus is actual value of dimmer output. Dimmer may take time to ramp or jump to the target level so LoadLevelTarget may not equal LoadLevelStatus until the dimmer reaches the desired state. Can be any value between 0 (off) and 100 (full on). Also, see SetLoadLevelTarget().

2.2.2. LoadLevelStatus

Current dimmer output level. Can be any value between 0 (off) and 100 (full on). This value may change multiple times during the transition from the current state to the LoadLevelTarget value. The dimmer should only send the evented update when LoadLevelStatus is equal to LoadLevelTarget, but a control point may query this value during the time the dimmer is transitioning to LoadLevelTarget.

2.2.3. OnEffectLevel

Defines the level to which LoadLevelTarget is set when the OnEffect variable is set to "OnEffectLevel" and the device is turned on, either logically (e.g. by a PowerSwitch service) or physically.If the OnEffect variable contains any other value than "OnEffectLevel", this value is not used at all. The unit is % of maximum output, like LoadLevelStatus.

2.2.4. OnEffect

Defines which value LoadLevelTarget is set to when power is provided to the device, either logically (e.g. by a PowerSwitch service) or physically. "OnEffectLevel" causes LoadLevelStatus beeing set to the value of OnEffectLevel; "LastSetting" sets LoadLevelTarget to the last value of LoadLevelStatus before a power-down occurred; "Default" defines a manufacturer specific setting for LoadLevelTarget (which includes leaving the value of LoadLevelTarget unchanged).

2.2.5. StepDelta

Incremental amount of change for StepUp and StepDown actions. The values are expressed in % and can range from 0 to 100. This value is used for incrementing or decrementing LoadLevelTarget when calling SetupUp() or StepDown(), respectively.

2.2.6. RampRate

Incremental level change per second. (5 = 5/100 of max level in 1 second) for RampUp and RampDown commands..

2.2.7. IsRamping

Set to 1 when ramping is currently in progress, also if it is paused.

2.2.8. RampPaused

Set to 1 when ramping is currently paused.

2.2.9. RampTime

Time to reach the LoadLevelTarget, applies only when a StartRampToLevel action has been invoked, otherwise it must be 0. The unit is milliseconds.

2.2.10. Relationships Between State Variables

2.2.10.1. OnEffectLevel and OnEffect

OnEffectLevel and OnEffect are both optional, however it is a requirement to implement either none or both together, as they are dependent.

2.2.10.2. RampRate, IsRamping and RampTime

RampRate, IsRamping and RampTime are all optional, however it is a requirement two implement either none or all, as they are dependent.

2.2.10.3. RampPaused

RampPaused is optional but can only be implemented when RampRate, IsRamping, and RampTime are implemented.

2.3. Eventing and Moderation

Table 2-3: Event Moderation

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
LoadLevelTarget	No	n/a	n/a		n/a
LoadLevelStatus	Yes	No	None		None
OnEffectLevel	No	n/a	n/a		n/a
OnEffect	No	n/a	n/a		n/a
StepDelta	Yes	No	None		None
RampRate	Yes	No	None		None
IsRamping	Yes	No	None		None
RampPaused	Yes	No	None		None
RampTime	No	n/a	n/a		n/a
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 2-4: Actions

Name	Req. or Opt. [1]
SetLoadLevelTarget	R
GetLoadLevelTarget	R
GetLoadLevelStatus	R
SetOnEffectLevel	O
SetOnEffect	O
GetOnEffectParameters	O
StepUp	O
StepDown	O
StartRampUp	O
StartRampDown	O
StopRamp	O
StartRampToLevel	O
SetStepDelta	O
GetStepDelta	O
SetRampRate	O
GetRampRate	O
PauseRamp	O
ResumeRamp	O
GetRampPaused	O
GetRampTime	O
GetIsRamping	O
Non-standard actions implemented by an UPnP vendor go here.	X

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

2.4.1. SetLoadLevelTarget

2.4.1.1. Arguments

Table 2-5: Arguments for SetLoadLevelTarget

Argument	Direction	relatedStateVariable
NewLoadLevelTarget	IN	LoadLevelTarget

2.4.1.2. Effect on State

Set LoadLevelTarget to newLoadLevelTarget. When LoadLevelTarget changes, the device implementation is supposed to set LoadLevelStatus as soon as possible to LoadLevelTarget, or, in other words, to physically execute the command.

A call to SetLoadLevelTarget() implicitely stops ramping if applicable ("last action wins"), resetting the state variables as defined by StopRamp().

2.4.1.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.2. GetLoadLevelTarget

2.4.2.1. Arguments

Table 2-6: Arguments for GetLoadLevelTarget

Argument	Direction	relatedStateVariable
retLoadLevelTarget	OUT	LoadLevelTarget

2.4.2.2. Effect on State

Returns LoadLevelTarget in retLoadLevelTarget.

2.4.2.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.

402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.3. GetLoadLevelStatus

2.4.3.1. Arguments

Table 2-7: Arguments for GetLoadLevelStatus

Argument	Direction	relatedStateVariable
retLoadLevelStatus	OUT	LoadLevelStatus

2.4.3.2. Effect on State

Returns LoadLevelStatus in retLoadLevelStatus.

2.4.3.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.4. SetOnEffectLevel

2.4.4.1. Arguments

Table 2-8: Arguments for SetOnEffectLevel

Argument	Direction	relatedStateVariable
NewOnEffectLevel	IN	OnEffectLevel

2.4.4.2. Effect on State

Set OnEffectLevel to newOnEffectLevel.

2.4.4.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.5. SetOnEffect

2.4.5.1. Arguments

Table 2-9: Arguments for SetOnEffect

Argument	Direction	relatedStateVariable
NewOnEffect	IN	OnEffect

2.4.5.2. Effect on State

Set OnEffect to newOnEffect.

2.4.5.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.6. GetOnEffectParameters

2.4.6.1. Arguments

Table 2-10: Arguments for GetOnEffectParameters

Argument	Direction	relatedStateVariable
RetOnEffect	OUT	OnEffect

Argument	Direction	relatedStateVariable
RetOnEffectLevel	OUT	OnEffectLevel

2.4.6.2. Effect on State

Returns the current values for OnEffect and OnEffectLevel. The function returns both at the same time to provide a consistent state as concurrent access may occur.

2.4.6.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.7. StepUp

2.4.7.1. Arguments

No Arguments.

2.4.7.2. Effect on State

Increments the LoadLevelTarget by StepDelta. If LoadLevelTarget would exceed the maximum of 100%, it is set to 100%.

A call to StepUp() implicitely stops ramping if applicable ("last action wins"), resetting the state variables as defined by StopRamp().

2.4.7.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.8. StepDown

2.4.8.1. Arguments

No Arguments.

2.4.8.2. Effect on State

Decrements the LoadLevelTarget by StepDelta. If LoadLevelTarget would fall below the minimum of 0%, it is set to 0%.

A call to StepDown() implicitely stops ramping if applicable ("last action wins"), resetting the state variables as defined by StopRamp().

2.4.8.3. Errors

errorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.9. StartRampUp

2.4.9.1. Arguments

No Arguments.

2.4.9.2. Effect on State

LoadLevelTarget is incremented by the value defined by the RampRate. The update of LoadLevelTarget must occur at least every second (the manufacturer may support faster update, e.g. every half a second). The function stops ramping if LoadLevelTarget reaches the maximum of 100%, StopRamp() is called or another action like SetLoadLevelTarget() stops ramping implicitly ("last action wins").

A call to StartRampUp() implicitly stops current ramping if applicable ("last action wins") before starting the new ramping function.

Set IsRamping to 1.

Set RampPaused to 0.

Start incrementing LoadLevelTarget.

2.4.9.3. Errors

errorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.10. StartRampDown

2.4.10.1. Arguments

No Arguments.

2.4.10.2. Effect on State

LoadLevelTarget is decremented by the value defined by the RampRate. The update of LoadLevelTarget must occur at least every second (the manufacturer may support faster update, e.g. every half a second). The function stops ramping if LoadLevelTarget reaches the minimum of 0%, StopRamp() is called or another action like SetLoadLevelTarget() stops ramping implicitly ("last action wins").

A call to StartRampDown() implicitly stops current ramping if applicable ("last action wins") before starting the new ramping function.

Sets IsRamping to 1.

Sets RampPaused to 0.

Starts decrementing LoadLevelTarget.

2.4.10.3. Errors

errorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.11. StopRamp

2.4.11.1. Arguments

No Arguments.

2.4.11.2. Effect on State

Stops ramping started either by StartRampUp(), StartRampDown() or StartRampToLevel() and resets ramping-related state table variables.

To ease device handling by a user (e.g. "stop all dynamic actions"), a call to StopRamp() issued when no ramping is currently in progress returns with success although no action is performed.

Set IsRamping to 0.

Set RampPaused to 0.

Set RampTime to 0.

2.4.11.3. Errors

errorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.12. StartRampToLevel

2.4.12.1. Arguments

Table 2-11: Arguments for StartRampToLevel

Argument	Direction	relatedStateVariable
NewLoadLevelTarget	IN	LoadLevelTarget
NewRampTime	IN	RampTime

2.4.12.2. Effect on State

Starts ramping from the current value of LoadLevelTarget to the value defined by newLoadLevelTarget over the time provided by newRampTime. Furthermore, RampTime is initally set to newRampTime and decreased until Ramptime reaches 0. The update of LoadLevelTarget and RampTime must occur at least every second (the manufacturer may support faster update, e.g. every half a second).

A call to StartRampToLevel() implicitely stops current ramping if applicable ("last action wins") before starting the new ramping function.

Set IsRamping to 1.

Set RampPaused to 0.

Start LoadLevelTarget approaching the value of newLoadLevelTarget.

Set RampTime to newRampTime.

2.4.12.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.13. SetStepDelta

2.4.13.1. Arguments

Table 2-12: Arguments for SetStepDelta

Argument	Direction	relatedStateVariable
NewStepDelta	IN	StepDelta

2.4.13.2. Effect on State

Sets StepDelta to newStepDelta. Variable is used by the StepUp() and SetupDown() actions

2.4.13.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.14. GetStepDelta

Table 2-13: Arguments for GetStepDelta

Argument	Direction	relatedStateVariable
RetStepDelta	OUT	StepDelta

2.4.14.1. Effect on State

Returns StepDelta which is used by the StepUp() and SetupDown() actions.

2.4.14.2. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.15. SetRampRate

2.4.15.1. Arguments

Table 2-14: Arguments for SetRampRate

Argument	Direction	relatedStateVariable
NewRampRate	IN	RampRate

2.4.15.2. Effect on State

Sets RampRate to newRampRate. Variable is used by the StartRampUp() and StartRampDown() actions

2.4.15.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.16. GetRampRate

2.4.16.1. Arguments

Table 2-15: Arguments for GetRampRate

Argument	Direction	relatedStateVariable
RetRampRate	OUT	RampRate

2.4.16.2. Effect on State

Returns RampRate which is used by the StartRampUp() and StartRampDown() actions.

2.4.16.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.

402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.17. PauseRamp

2.4.17.1. Arguments

No Arguments.

2.4.17.2. Effect on State

If ramping is in progress, RampPaused is set to 1 and the ramping function is paused until a call to ResumeRamp() occurs.

Set RampPaused to 1.

2.4.17.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.
700	No ramping in progress	The PauseRamp() action has been invoked while no ramping was in progress.

2.4.18. ResumeRamp

2.4.18.1. Arguments

No Arguments.

2.4.18.2. Effect on State

If ramping is in progress but has been paused by a call to PauseRamp(), RampPaused is set to 0 and ramping is continued.

Set RampPaused to 0.

2.4.18.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.
700	No ramping in pause mode	The ResumeRamp() action has been invoked while no ramping was in progress or ramping was not paused.

2.4.19. GetRampPaused

2.4.19.1. Arguments

Table 2-16: Arguments for GetRampPaused

Argument	Direction	relatedStateVariable
RetRampPaused	OUT	RampPaused

2.4.19.2. Effect on State

Returns RampPaused.

2.4.19.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.20. GetRampTime

2.4.20.1. Arguments

Table 2-17: Arguments for GetRampTime

Argument	Direction	relatedStateVariable
RetRampTime	OUT	RampTime

2.4.20.2. Effect on State

Returns RampTime.

2.4.20.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.21. GetIsRamping

2.4.21.1. Arguments

Table 2-18: Arguments for GetIsRamping

Argument	Direction	relatedStateVariable
RetIsRamping	OUT	IsRamping

2.4.21.2. Effect on State

Returns IsRamping.

2.4.21.3. Errors

ErrorCode	ErrorDescription	Description
401	Invalid Action	See UPnP Device Architecture Section on Control.
402	Invalid Args	See UPnP Device Architecture Section on Control.
403	Out of Synch	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 the UPnP Forum Technical Committee.

2.4.22.

2.4.23. 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).

2.4.24. Relationships Between Actions

In general, the principle is "last action wins". Therefore, the background operations initiated StartRampUp() and StartRampDown() and StartRampToLevel() are canceled by any call to SetLoadLevelTarget(), StepUp() or StepDown(), and by initiating another ramping function. For more details, see "Theory of operation".

The function StopRamp() is used to cancel the background operations initiated by StartRampUp() and StartRampDown() and StartRampToLevel(). Furthermore, the background operation can be paused by using PauseRamp() and continued by ResumeRamp() (when implemented).

Following are the list of implementation packages. If a particular package is chosen all actions in that package have to be implemented.

2.4.24.1. Mandatory actions

SetLoadLevelTarget(newLoadLevelTarget)

GetLoadLevelTarget(RetLoadLevelTarget)

GetLoadLevelStatus(retLoadLevelStatus)

2.4.24.2. OnEffect actions (optional)

SetOnEffectLevel (newOnEffectLevel)

SetOnEffect(newOnEffect)

GetOnEffectParameters(retOnEffect, retOnEffectLevel)

2.4.24.3. Stepping actions (optional)

StepUp( )

StepDown( )

SetStepDelta(newStepDelta)

GetStepDelta(retStepDelta)

2.4.24.4. Simple ramping actions (optional)

StartRampUp( )

StartRampDown( )

StopRamp( )

StartRampToLevel(newLoadLevelTarget, newRampTime)

SetRampRate(newRampRate)

GetRampRate(retRampRate)

GetIsRamping(retIsRamping)

GetRampPaused(retRampPaused)

GetRampTime(retRampTime)

2.4.24.5. Additional ramping actions (optional but require Simple ramping implementations)

PauseRamp( )

ResumeRamp( )

2.4.25. 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 must be returned.

Table 2-19: 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.
701-799		Common action errors defined by the UPnP Forum working committees.
800-899	*TBD*	*(Specified by UPnP vendor.)*

2.5. Theory of Operation

Instances of Dimming Services are embedded into devices to provide a standard means of programmatic control over dimmable lighting devices.

This service model provides for situations where requested state changes may not cause immediate output state changes, reflected via the *Status* variable, for any number of reasons. For example if there are time delays involved or maybe the requested state can't be achieved because of a hardware failure.

In the simplest of cases the output state (*LoadLevelStatus*) will always follow the requested state changes submitted via *LoadLevelTarget*.

There is also the situation where the *LoadLevelStatus* variable could change state without any programmatic action against this model at all. For example, this could happen if there was a front-panel power control that was changed by a user.

In addition to the explicit setting of *LoadLevelTarget*, several typical but optional functions are defined. The value of *LoadLevelTarget* can therefore be modified by either:

1. Explicit setting of *LoadLevelTarget* by calling SetLoadLevelTarget(),
2. Setting *LoadLevelTarget* by the OnEffect (when power is provided to the device, either logically or physically) (optional)
3. Stepping up or down by calling StepUp() or StepDown() (optional), or
4. Ramping up or down or to a specific value by calling StartRampUp(), StartRampDown() or RampToLevel() (optional)

While support of SetLoadLevelTarget() is mandatory, any combination of OnEffect, stepping, or ramping implementation is valid.

The actions invoking ramping (StartRampUp(), StartRampDown(), StartRampToLevel()) return immediately but start a background operation which modifies the state over time. These operations stop when:

1. LoadLevelTarget reaches the maximum (100%) after a call to StartRampUp()
2. LoadLevelTarget reaches the minimum (0%) after a call to StartRampDown()
3. LoadLevelTarget reaches the targeted level after a call to StartRampToLevel()
4. Ramping is terminated explicitly by invoking StopRamp()
5. Ramping is terminated implicitly by invoking another action affecting the LoadLevelTarget, e.g. SetLoadLevelTarget()

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>SetLoadLevelTarget</name>
      <argumentList>
        <argument>
          <name>newLoadlevelTarget</name>
          <direction>in</direction>
          <relatedStateVariable>LoadLevelTarget</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetLoadLevelTarget</name>
      <argumentList>
        <argument>
          <name>retLoadlevelTarget</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>LoadLevelTarget</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetLoadLevelStatus</name>
      <argumentList>
        <argument>
          <name>retLoadlevelStatus</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>LoadLevelStatus</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetOnEffectLevel</name>
      <argumentList>
        <argument>
          <name>newOnEffectLevel</name>
          <direction>in</direction>
          <relatedStateVariable>OnEffectLevel</relatedStateVariable>
        </argument>
      </argumentList>
    </action>

    <action>
      <name>SetOnEffect</name>
      <argumentList>
```

```
      <argument>
        <name>newOnEffect</name>
        <direction>in</direction>
        <relatedStateVariable>OnEffect</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetOnEffectParameters</name>
    <argumentList>
      <argument>
        <name>retOnEffect</name>
        <direction>out</direction>
        <relatedStateVariable>OnEffect</relatedStateVariable>
      </argument>
      <argument>
        <name>retOnEffectLevel</name>
        <direction>out</direction>
        <relatedStateVariable>OnEffectLevel</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>StepUp</name>
  </action>
  <action>
    <name>StepDown</name>
  </action>
  <action>
    <name>StartRampUp</name>
  </action>
  <action>
    <name>StartRampDown</name>
  </action>
  <action>
    <name>StopRamp</name>
  </action>
  <action>
    <name>StartRampToLevel</name>
    <argumentList>
      <argument>
        <name>newLoadLevelTarget</name>
        <direction>in</direction>
        <relatedStateVariable>LoadLevelTarget</relatedStateVariable>
      </argument>
      <argument>
        <name>newRampTime</name>
        <direction>in</direction>
        <relatedStateVariable>RampTime</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>SetStepDelta</name>
    <argumentList>
      <argument>
```

```
        <name>newStepDelta</name>
        <direction>in</direction>
        <relatedStateVariable>StepDelta</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetStepDelta</name>
    <argumentList>
      <argument>
        <name>retStepDelta</name>
        <direction>out</direction>
        <retval />
        <relatedStateVariable>StepDelta</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>SetRampRate</name>
    <argumentList>
      <argument>
        <name>newRampRate</name>
        <direction>in</direction>
        <relatedStateVariable>RampRate</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>GetRampRate</name>
    <argumentList>
      <argument>
        <name>retRampRate</name>
        <direction>out</direction>
        <retval />
        <relatedStateVariable>RampRate</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>PauseRamp</name>
  </action>
  <action>
    <name>ResumeRamp</name>
  </action>
  <action>
    <name>GetIsRamping</name>
    <argumentList>
      <argument>
        <name>retIsRamping</name>
        <direction>out</direction>
        <retval />
        <relatedStateVariable>IsRamping</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
```

```
      <name>GetRampPaused</name>
      <argumentList>
        <argument>
          <name>retRampPaused</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>RampPaused</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetRampTime</name>
      <argumentList>
        <argument>
          <name>retRampTime</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>RampTime</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
      Declarations for other actions added by UPnP vendor (if any) go
here
  </actionList>

  <serviceStateTable>
    <stateVariable sendEvents="no">
      <name>LoadLevelTarget</name>
      <dataType>ui1</dataType>
      <defaultValue>0</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>100</maximum>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>LoadLevelStatus</name>
      <dataType>ui1</dataType>
      <defaultValue>0</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>100</maximum>
      </allowedValueRange>
    </stateVariable>

    <stateVariable sendEvents="no">
      <name>OnEffectLevel</name>
      <dataType>ui1</dataType>
      <defaultValue>100</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>100</maximum>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>OnEffect</name>
```

```
      <dataType>string</dataType>
      <defaultValue>Default</defaultValue>
      <allowedValueList>
        <allowedValue>OnEffectLevel</allowedValue>
        <allowedValue>LastSetting</allowedValue>
        <allowedValue>Default</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>StepDelta</name>
      <dataType>ui1</dataType>
      <defaultValue> Manufacturer defined default value </defaultValue>
      <allowedValueRange>
        <minimum>1</minimum>
        <maximum>100</maximum>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>RampRate</name>
      <dataType>ui1</dataType>
      <defaultValue>0</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>100</maximum>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>RampTime</name>
      <dataType>ui4</dataType>
      <defaultValue>0</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>4294967295</maximum>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>IsRamping</name>
      <dataType>boolean</dataType>
      <defaultValue>0</defaultValue>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>RampPaused</name>
      <dataType>boolean</dataType>
      <defaultValue>0</defaultValue>
    </stateVariable>
  </serviceStateTable>
      Declarations for other state variables added by UPnP vendor (if
any)go here
</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: ***Dimming1.xml***
- The UPnP Test Tool service template test file: ***Dimming1.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.