TwoWayMotionMotor:1
Service Template Version 1.01

For UPnP Version 1.0
Status: Standardized DCP
Date: July 27th, 2007

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.

THE UPNP™ FORUM TAKES NO POSITION AS TO WHETHER ANY INTELLECTUAL PROPERTY RIGHTS EXIST IN THE STANDARDIZED DCPS. THE STANDARDIZED DCPS ARE PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP™ FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE STANDARDIZED DCPS, INCLUDING BUT NOT LIMITED TO ALL IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPOSE, OF REASONABLE CARE OR WORKMANLIKE EFFORT, OR RESULTS OR OF LACK OF NEGLIGENCE.

© 2008 Contributing Members of UPnP Forum. All Rights Reserved

Authors	Company
Dr Serge NEUMAN	Somfy
Hans-Joachim LANGELS	Siemens AG

[1] Note: UPnP Forum in no way guarantees the accuracy or completeness of this author list and in no way implies any rights for or support from those members listed. This list is not the specifications' contributor list that is kept on the UPnP Forum's website.

Contents

1. OVERVIEW AND SCOPE 4

2. SERVICE MODELING DEFINITIONS 5

2.1. SERVICE TYPE 5
2.2. STATE VARIABLES 5
2.2.1. OperationMode 7
2.2.2. ServiceLocked 7
2.2.3. Position 7
2.2.4. PositionArgType 8
2.2.5. Relationships between State Variables 8
2.3. EVENTING AND MODERATION 9
2.4. ACTIONS 10
2.4.1. (Void) Open() 11
2.4.2. (Void) Close() 13
2.4.3. (Void) Stop() 15
2.4.4. (String) GetOperationMode(RetVal) 16
2.4.5. (Void) SetOperationMode(String) 17
2.4.6. (Void) IsLocked(Boolean) 18
2.4.7. (Void) Lock() 19
2.4.8. (Void) UnLock() 20
2.4.9. (Float) GetPosition(RetVal) 21
2.4.10. (Void) SetPosition(I1) 22
2.4.11. (String) GetPositionArgType(RetVal) 24
2.4.12. Non-Standard Actions Implemented by a UPnP Vendor 25
2.4.13. Relationships between Actions 25
2.4.14. Common Error Codes 25
2.5. THEORY OF OPERATION 26

3. XML SERVICE DESCRIPTION 27

4. TEST 30

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

List of Tables

Table 1: State Variables 5

Table *1.1:* allowedValueList for *OperationMode* 6

Table *1.2:* DefaultValue for *OperationMode* 6

Table *1.5:* allowedValueRange for *Position* 6

Table *1.6:* DefaultValue for *Position* 6

Table *1.7:* allowedValueList for *PositionArgType* 7

Table *1.8:* DefaultValue for *PositionArgType* 7

Table 2: Event Moderation 9

Table 3: Actions 10

Table 4.1: Arguments for *Open* 11

Table 4.2: Arguments for *Close* 13

Table 4.3: Arguments for *Stop* 15

Table 4.4: Arguments for *GetOperationMode* 16

Table 4.5: Arguments for *SetOperationMode* 17

Table 4.6: Arguments for *IsLocked* 18

Table 4.7: Arguments for *Lock* 19

Table 4.8: Arguments for *UnLock* 20

Table 4.9: Arguments for *GetPosition* 21

Table 4.10: Arguments for *SetPosition* 22

Table 4.11: Arguments for *GetPositionArgType* 24

Table 5: Common Error Codes 25

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

1. Overview and Scope

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

This service-type enables the following functions:

- Driving a motor between two fixed end limits.
- Reaching any position written in percentage of the full run (optional).
- Returning the actual blind position written in percentage of the full run (optional).

This service template does not address:

- Any motorization whose run is not bounded.
- Any speed control.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2. Service Modeling Definitions

2.1. Service Type

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

urn:schemas-UPnP-org:service**:***TwoWayMotionMotor:1*.

2.2. State Variables

Table 1: State Variables

Variable Name	Req. or Opt.[1]	Data Type	Allowed Value [2]	Default Value [2]	Eng. Units
OperationMode	*R*	*String*	*"Manual Unprotected", "Manual Protected", "Automatic"*	*Vendor-defined*	
ServiceLocked	*O*	*Boolean*	*0,1*	*1*	
Position	*O*	*I1*	*0 to 100*	*Vendor-defined*	
PositionArgType	*O*	*String*	*"End Limits", "Continuous"*	*Vendor-defined*	
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 this column 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.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

Table *1.1:* allowedValueList for *OperationMode*

Value	Req. or Opt.
Manual Unprotected	*O*[1]
Manual Protected	*O*[1,2]
Automatic	*O*[2]
Vendor-defined	*O*[2]

[1] *At least one out of "Manual Unprotected" and "Manual Protected" must be implemented.*
[2] *If one of those optional values is implemented "ServiceLocked" must be implemented.*

Table *1.2:* DefaultValue for *OperationMode*

Value	Req. or Opt.
Vendor-defined	*R*

Table *1.5:* allowedValueRange for *Position*

	Value	Req. or Opt.
Minimum	*0*	*R*
Maximum	*100*	*R*
Step	*Vendor-defined*	*O*

Table *1.6:* DefaultValue for *Position*

Value	Req. or Opt.
Vendor-defined[1]	*R*

[1]*This value must reflect the real position of the device.*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

Table *1.7:* allowedValueList for *PositionArgType*

Value	Req. or Opt.
End Limits	*R*
Continuous	*R*

Table *1.8:* DefaultValue for *PositionArgType*

Value	Req. or Opt.
Vendor-defined[1]	*R*

[1]*This value cannot be changed afterward.*

2.2.1. OperationMode

This variable describes with "ServiceLocked" the actual operation mode of the device:

- "Manual Unprotected" allows the user to control the device and disables all the protections available. "Manual Unprotected" is required.
- "Manual Protected" allows the user to control the device and enable all the protections available. "Manual Protected" is optional,
- "Automatic" disables any manual order. "Automatic" is optional.

2.2.2. ServiceLocked

This variable defines if the device is locked or not:

- 1: actions that get information from the service and those that change the operation mode (i.e. "OperationMode" + "ServiceLocked") are enabled. All other actions are disabled. "ServiceLocked" overcomes automation and protections.
- 0: all actions are enabled.

"ServiceLocked" is optional, but it must be implemented if automation or protections can be turned on.

2.2.3. Position

This variable stores the actual position of the device. It's written in percentage of the full run:

- 0 is at one end of the full run. For a human being looking at the device, it should be synonymous with closed, down, far right or switch limit of a clockwise move.
- 100 is at the other end of the full run. For a human being looking at the device, it should be synonymous with open, up, far left or switch limit of a counter-clockwise move.
- 50 is the half run. It's also the value of "Position" if an accurate number cannot be provided which is memorized by "PositionArgType".

"Position" can be implemented if and only if the service can informs when a limit switch is reached and which one is it. This variable is optional, but if it's implemented, "PositionArgType" must also be implemented.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.2.4. PositionArgType

This variable describes how to interpret “Position”:

- “End Limits”: the service can only determine if the device has reached a limit switch and if so, which one.
- “Continuous”: the service can always give an accurate value.

“PositionArgType” is implemented if and only if “Position” is implemented. This state variable must be defined by the vendor and cannot be changes afterward.

2.2.5. Relationships between State Variables

“ServiceLocked” must be implemented if “OperationMode” can value Manual Protected” or “Automatic”.

“PositionArgType” and “Position” cannot be implemented one without the other.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

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]
OperationMode	*Yes*	*No*			
ServiceLocked	*Yes*	*No*			
Position	*Yes*	*Yes*			*5*
PositionArgType	*No*	*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).

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4. Actions

The service provides three kinds of action:

1. to control the device (“Open”, “Close”; “Stop”; “SetPosition”). “SetPosition” is implemented if and only if the device can perform it,
2. to know and to set the operation mode (“GetOperationMode”, “SetOperationMode”, “IsLocked”, “Lock” and “UnLock”),
3. to have information about the device position (“GetPosition”, “GetPositionArgType”). These functions are implemented if and only if the variables “Position” and “PositionArgType” are implemented.

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]
Open	*R*
Close	*R*
Stop	*R*
GetOperationMode	*R*
SetOperationMode	*R*
IsLocked	*O*
Lock	*O*
UnLock	*O*
GetPosition	*O*
SetPosition	*O*
GetPositionArgType	*O*
Non-standard actions implemented by an UPnP vendor go here.	X

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

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.1. (Void) Open()

2.4.1.1. Arguments

Table 4.1: Arguments for *Open*

Argument	Direction	relatedStateVariable

2.4.1.2. Dependency on State

This action is disabled if "ServiceLocked" is True (see description 2.4.1.4)

"OperationMode" changes how this action is executed (see description 2.4.1.4).

2.4.1.3. Effect on State

If the action is done:

If "Position" is implemented then it's incremented according to the sensor (expressed in percentage of the full run).

If the protection must stop the move then "ServiceLocked" switches True.

2.4.1.4. Description

If the device is locked then an error 700 ("Forbidden") is retuned.

Otherwise, depending on the operation mode, the system acts differently:

- "OperationMode" = "Manual Unprotected"

  Drives the motor in the way that increments "Position" (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise) until one of the following conditions is verified:

  "Stop", "Close" or "SetPosition" with a position requested smaller than the actual one, is received,

  The device is fully opened.

- "OperationMode" = "Manual Protected"

  If the protection doesn't allow the device to open, an error 701 ("Not Allowed") is returned. Then the protection decides if the device must be locked or not.

  If the protection allows the device to open, the system drives the motor in the way that increments "Position" (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise) until one of the following conditions is verified:

  "Stop", "Close" or "SetPosition" with a position requested smaller than the actual one, is received,

  The device is fully opened.

  The protection stops the opening. If it happens, "ServiceLocked" must immediately switch True. Note: the protection can still drive the motor to put the device in a safe configuration.

- "OperationMode" = "Automatic"

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

An error 700 (“Forbidden”) is returned.

2.4.1.5. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
700	*Forbidden*	*The device is locked or in the automatic mode*
701	*Not Allowed*	*The protection doesn’t allow the action.*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.2. (Void) Close()

2.4.2.1. Arguments

Table 4.2: Arguments for *Close*

Argument	Direction	relatedStateVariable

2.4.2.2. Dependency on State

This action is disabled if "ServiceLocked" is True (see description 2.4.2.4)

"OperationMode" changes how this action is executed (see description 2.4.2.4).

2.4.2.3. Effect on State

If the action is done:

If "Position" is implemented then it's decremented according to the sensor (expressed in percentage of the full run),

If the protection must stop the move then "ServiceLocked" switches True.

2.4.2.4. Description

If the device is locked then an error 700 ("Forbidden") is retuned.

Otherwise, depending on the operation mode, the system acts differently:

- "OperationMode" = "Manual Unprotected"

  Drives the motor in the way that decrements "Position" (i.e. the device is closing, moving down, moving right or the motor is turning clockwise) until one of the following conditions is verified:

  "Stop", "Open" or "SetPosition" with a position requested greater than the actual one, is received,

  the device is fully closed.

- "OperationMode" = "Manual Protected"

  If the protection doesn't allow the device to close, an error 701 ("Not Allowed") is returned. Then the protection decides if the device must be locked or not.

  If the protection allows the device to close, the system drives the motor in the way that decrements "Position" (i.e. the device is closing, moving down, moving right or the motor is turning clockwise) until one of the following conditions is verified:

  "Stop", "Open" or "SetPosition" with a position requested greater than the actual one, is received,

  The device is fully closed.

  The protection stops the move. If it happens, "ServiceLocked" must immediately switch True. Note: the protection can still drive the motor to put the device in a safe configuration.

- "OperationMode" = "Automatic"

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

An error 700 (“Forbidden”) is returned.

2.4.2.5. Errors

ErrorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
700	*Forbidden*	*The device is locked or in the automatic mode.*
701	*Not Allowed*	*The protection doesn’t allow the action.*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.3. (Void) Stop()

2.4.3.1. Arguments

Table 4.3: Arguments for *Stop*

Argument	Direction	relatedStateVariable

2.4.3.2. Dependency on State

This action is disabled if "ServiceLocked" is True (see description 2.4.3.4)

"OperationMode" changes how this action is executed (see description 2.4.3.4).

2.4.3.3. Effect on State

If the action tries to overcome the protection or the automation then "ServiceLocked" switches True.

2.4.3.4. Description

If the device is locked then an error 700 ("Forbidden") is returned.

Otherwise, depending on the operation mode, the system acts differently:

- "OperationMode" = "Manual Unprotected"

  Stops the device at the current position until a control action ("Open", "Close",or "SetPosition") is received.

- "OperationMode" = "Manual Protected"

  If the protection doesn't allow the device to stop then "ServiceLocked" switches True. Note: the protection can still drive the motor to put the device in a safe configuration.

  If the protection allows the device to stop, the system is stop at the current "Position". It remains motionless until one of the following conditions is verified:

  A control action ("Open", "Close", or "SetPosition" with a position requested not equal to the actual one) is received,

  The protection decides to move the device.

- "OperationMode" = "Automatic"

  If the device is moving then "ServiceLocked" switches True. Note: the automation can still drive the motor to put the device in a safe configuration.

2.4.3.5. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
700	*Forbidden*	*The device is locked or in the automatic mode*
701	*Not Allowed*	*The protection doesn't allow the action.*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.4. (String) GetOperationMode(RetVal)

2.4.4.1. Arguments

Table 4.4: Arguments for *GetOperationMode*

Argument	Direction	relatedStateVariable
RetOperationMode	*Out – RetVal*	*OperationMode*

2.4.4.2. Description

Returns the current value of “OperationMode” in “RetOperationMode”.

2.4.4.3. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.5. (Void) SetOperationMode(String)

2.4.5.1. Arguments

Table 4.5: Arguments for *SetOperationMode*

Argument	Direction	relatedStateVariable
NewOperationMode	*In*	*OperationMode*

2.4.5.2. Dependency on State

Each not implemented value of “OperationMode” is an invalid argument for “SetOperationMode”.

2.4.5.3. Effect on State

Sets the new value of “OperationMode”.

2.4.5.4. Description

If “NewOperationMode” is an operation mode implemented then “OperationMode” gets this new value.

Otherwise an error 702 (“Disabled”) is returned.

2.4.5.5. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
702	*Disabled*	*The operation mode request is disabled.*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.6. (Void) IsLocked(Boolean)

2.4.6.1. Arguments

Table 4.6: Arguments for *IsLocked*

Argument	Direction	relatedStateVariable
RetLocking	*Out – RetVal*	*ServiceLocked*

2.4.6.2. Dependency on State

This action is implemented if and only if “ServiceLocked” is implemented.

2.4.6.3. Description

Returns the current value of “ServiceLocked” in “RetLocking”.

2.4.6.4. Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.7. (Void) Lock()

2.4.7.1. Arguments

Table 4.7: Arguments for *Lock*

Argument	Direction	relatedStateVariable

2.4.7.2. Dependency on State

This action is implemented if and only if "ServiceLocked" is implemented.

2.4.7.3. Effect on State

If "ServiceLocked" is implemented then switches it True.

2.4.7.4. Description

Switches "ServiceLocked" True and stops immediately any move.

2.4.7.5. Errors

ErrorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.8. (Void) UnLock()

2.4.8.1. Arguments

Table 4.8: Arguments for *UnLock*

Argument	Direction	relatedStateVariable

2.4.8.2. Dependency on State

This action is implemented if and only if “ServiceLocked” is implemented.

2.4.8.3. Effect on State (if any)

If “ServiceLocked” is implemented then it switches False

2.4.8.4. Description

If the device is performing a safe move then an error 701 (“Not Allowed”) is returned. Otherwise, “ServiceLocked” switches False and any move is immediately stopped.

2.4.8.5. Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
701	*Not Allowed*	*The protection doesn’t allow the action.*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.9. (Float) GetPosition(RetVal)

2.4.9.1. Arguments

Table 4.9: Arguments for *GetPosition*

Argument	Direction	relatedStateVariable
RetPosition	*Out – RetVal*	*Position*

2.4.9.2. Dependency on State

This action is implemented if and only if "Position" is implemented.

2.4.9.3. Description

Returns the current value of "Position" in "RetPosition".

2.4.9.4. Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.10.(Void) SetPosition(I1)

2.4.10.1. Arguments

Table 4.10: Arguments for *SetPosition*

Argument	Direction	relatedStateVariable
NewPosition	*In*	*Position*

2.4.10.2.Dependency on State

This action is implemented if and only if "PositionArgType" equals "Continuous".

This action is disabled if "ServiceLocked" is True (see description 2.4.10.4)

"OperationMode" changes how this action is executed (see description 2.4.10.4).

2.4.10.3.Effect on State

If the action is performed, "Position" will increase or decrease until it reaches the value of "NewPosition".

If the protection stops the move, "ServiceLocked" switches True.

2.4.10.4.Description

If "NewPosition" is not between 0 and 100 (included) then an error 601 ("Out of Range") is returned.

If the device is locked then an error 700 ("Forbidden") is returned.

Otherwise, depending on the operation mode, the system acts differently:

- "OperationMode" = "Manual Unprotected"

  If "NewPosition" equals "Position" then the device is stopped.

  If "NewPosition" is greater than "Position" then the motor is driven in the way that increments "Position" (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise).

  If "NewPosition" is smaller than "Position" then the motor is driven in the way that decrements "Position" (i.e. the device is closing, moving down, moving right or the motor is turning clockwise).

  In these two last cases, the motion is performed until one of the following conditions is verified:

  A control action ("Stop", "Open", "Close" or "SetPosition") is received,

  "Position" equals "NewPosition".

- "OperationMode" = "Manual Protected"

  If "NewPosition" equals "Position", the protection must allow the stop. Otherwise, an error 701 ("Not Allowed") is returned and "ServiceLocked" switches True. Note: in that case, the protection can still drive the motor to put the device in a safe configuration.

  If "NewPosition" is greater than "Position", the protection must allow the motor to be driven in the way that increments "Position" (i.e. the device is opening, moving up, moving left or the motor is turning counter-clockwise). If the protection doesn't allow it, an error 701 ("Not Allowed") is returned and "ServiceLocked" switches True.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

If “NewPosition” is smaller than “Position”, the protection must allow the motor to be driven in the way that decrements “Position” (i.e. the device is closing, moving down, moving right or the motor is turning clockwise). If the protection doesn’t allow it, an error 701 (“Not Allowed”) is returned and “ServiceLocked” switches True.

In these two last cases, any move is performed until one of the following conditions is verified:

A control action (“Stop”, “Open”, “Close” or “SetPosition”) is received,

“Position” equals “NewPosition”.

The protection stops the move. If it happens, “ServiceLocked” must immediately switch True. Note: the protection can still drive the motor to put the device in a safe configuration.

- “OperationMode” = “Automatic”

  An error 700 (“Forbidden”) is returned.

2.4.10.5.Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
601	Out of Range	The argument value is not between 0 and 100 included.
700	*Forbidden*	*The device is locked or in the automatic mode.*
701	*Not Allowed*	*The protection doesn’t allow the action.*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.11.(String) GetPositionArgType(RetVal)

2.4.11.1. Arguments

Table 4.11: Arguments for *GetPositionArgType*

Argument	Direction	relatedStateVariable
RetArgType	*Out – RetVal*	*PositionArgType*

2.4.11.2.Dependency on State

This action is implemented if and only if "PositionArgType" is implemented.

2.4.11.3.Description

Returns the current value of "PositionArgType" in "RetArgType".

2.4.11.4.Errors

errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.4.12.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).

2.4.13.Relationships between Actions

"SetPosition" with a position requested greater than the actual position acts as "Open". If the position requested is smaller than the actual position then "SetPosition" acts as "Close".

"Lock" and "UnLock" are opposite actions.

2.4.14.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 5: Common Error Codes

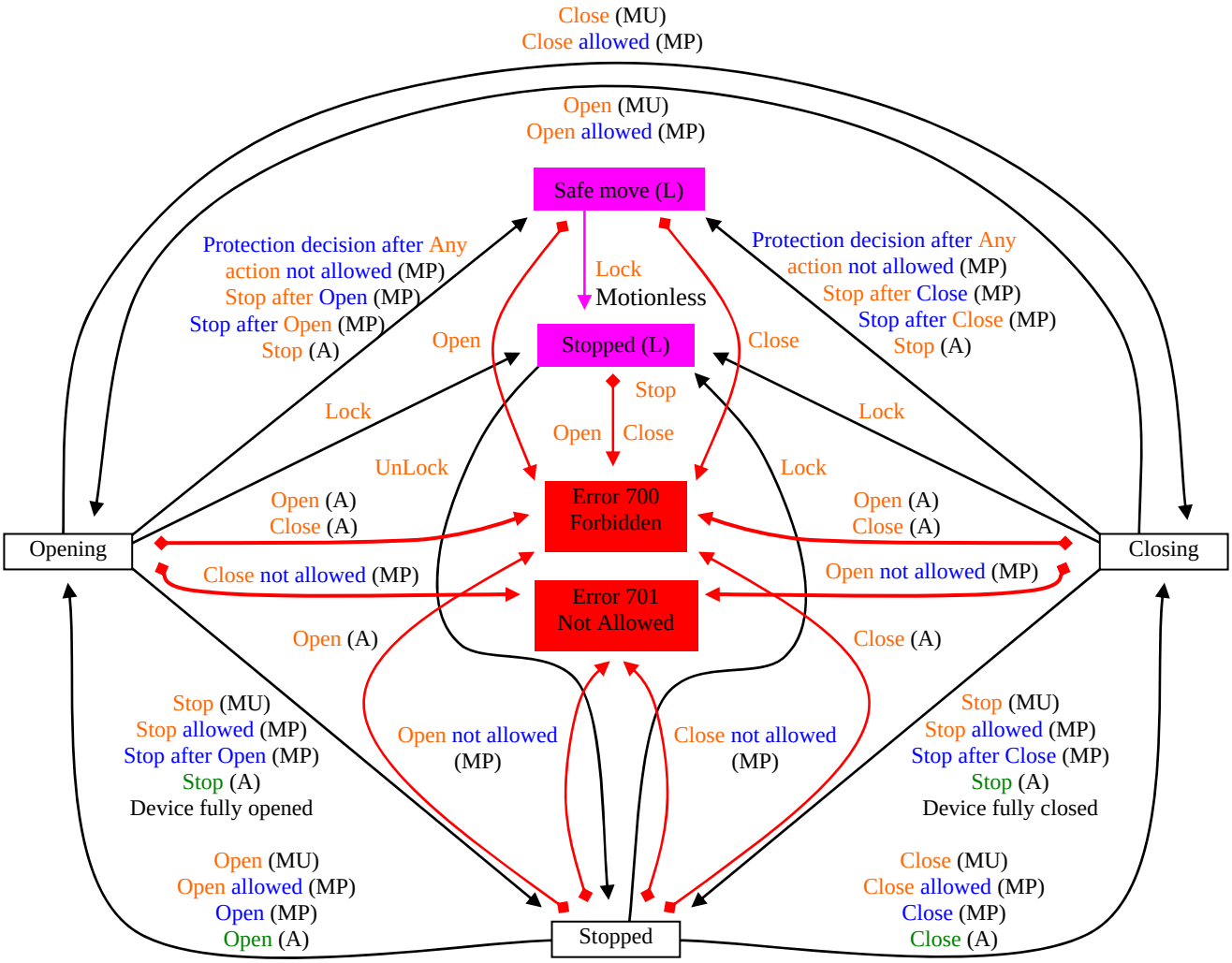
errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
404	Invalid Var	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
601	Out of Range	The argument value is not between 0 and 100 included.
700	*Forbidden*	*The device is locked or in the automatic mode.*
701	*Not Allowed*	*The protection doesn't allow the action.*
702	*Disabled*	*The operation mode request is disabled*
704-799		Common action errors defined by the UPnP Forum working committees.
800-899	*TBD*	*(Specified by UPnP vendor.)*

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

2.5. Theory of Operation

The device cannot be moved if the service is *locked (L)*. When it's unlocked, it can be controlled *manually* in the manual unprotected operation mode (MU) or under the control of the *protection* in the manual protected mode (MP). The last solution is to let the *automation* controls the device in the automatic operation mode (A). The following state machine summarizes how to control the device. To simplify the notation, only "Open", "Close", "Stop", "Stopped", "Opening" and "Closing" are shown. For full description refer to the corresponding action.

The other actions provide the tools needed to handle the state variables (see Description of each action for full information).



© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

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>Open</name>
      <argumentList>
        <argument>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>Close</name>
      <argumentList>
        <argument>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>Stop</name>
      <argumentList>
        <argument>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>GetOperationMode</name>
      <argumentList>
        <argument>
          <name>RetOperationMode</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>OperationMode</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>SetOperationMode</name>
      <argumentList>
        <argument>
          <name>NewOperationMode</name>
          <direction>in</direction>
          <relatedStateVariable>OperationMode</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
    <name>IsLocked</name>
      <argumentList>
        <argument>
```

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

```
        <name>RetLocking</name>
        <direction>out</direction>
        <retval />
        <relatedStateVariable>ServiceLocked</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>Lock</name>
    <argumentList>
      <argument>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>UnLock</name>
    <argumentList>
      <argument>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>GetPosition</name>
    <argumentList>
      <argument>
        <name>RetPosition</name>
        <direction>out</direction>
        <retval />
        <relatedStateVariable>Position</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>SetPosition</name>
    <argumentList>
      <argument>
        <name>NewPosition</name>
        <direction>in</direction>
        <relatedStateVariable>Position</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  <action>
  <name>GetPositionArgType</name>
    <argumentList>
      <argument>
        <name>RetArgType</name>
        <direction>out</direction>
        <retval />
        <relatedStateVariable>PositionArgType</relatedStateVariable>
      </argument>
    </argumentList>
  </action>
  Declarations for other actions added by UPnP vendor (if any) go here
</actionList>
<serviceStateTable>
```

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

```
    <stateVariable sendEvents="yes">
      <name>OperationMode</name>
      <dataType>String</dataType>
      <defaultValue>Vendor defined</defaultValue>
      <allowedValueList>
        <allowedValue>Manual Unprotected</allowedValue>
        <allowedValue>Manual Protected</allowedValue>
        <allowedValue>Automatic</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>ServiceLocked</name>
      <dataType>Boolean</dataType>
      <defaultValue>1</defaultValue>
      <allowedValueList>
        <allowedValue>1</allowedValue>
        <allowedValue>0</allowedValue>
      </allowedValueList>
    </stateVariable>
    <stateVariable sendEvents="yes">
      <name>Position</name>
      <dataType>I1</dataType>
      <defaultValue>Vendor defined</defaultValue>
      <allowedValueRange>
        <minimum>0</minimum>
        <maximum>100</maximum>
        <step></step>
      </allowedValueRange>
    </stateVariable>
    <stateVariable sendEvents="no">
      <name>PositionArgType</name>
      <dataType>String</dataType>
      <defaultValue>Vendor defined</defaultValue>
      <allowedValueList>
        <allowedValue>End Limits</allowedValue>
        <allowedValue>Continuous</allowedValue>
      </allowedValueList>
    </stateVariable>
    Declarations for other state variables added by UPnP vendor (if any)
    go here
  </serviceStateTable>
</scpd>
```

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.

4. Test

Syntactical testing is performed by the UPnP test tool based on the XML description as provided in Section 3.

The working committee and the implementers have come to the conclusion that further test descriptions e.g. for semantical testing do not provide a higher level of interoperability.

Thus the XML description is deemed to be sufficient for testing of devices that implement this template and further test descriptions are are not provided by this template.

© 2008 Contributing Members of the UPnP Forum. All Rights Reserved.