

# ControlValve:1 Service Template

For UPnP™ Device Architecture V 1.0

**Status: Standardized DCP**

**Date: May 13<sup>th</sup>, 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 Membership Agreement. UPnP Forum Members have rights and licenses defined by Section 3 of the UPnP 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 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.

© 2001-2003 Contributing Members of the UPnP™ Forum. All Rights Reserved

Authors	Company
Andrew Fiddian-Green	Siemens Building Technologies Ltd.
Larry Stickler	Honeywell

## Contents

<b>1. OVERVIEW AND SCOPE .....</b>	<b>3</b>
1.1. CHANGE LOG .....	3
<b>2. SERVICE MODELING DEFINITIONS .....</b>	<b>3</b>
2.1. SERVICE TYPE .....	3
2.2. STATE VARIABLES .....	4
2.2.1. <i>ControlMode</i> .....	4
2.2.2. <i>PositionTarget</i> .....	4
2.2.3. <i>PositionStatus</i> .....	4
2.2.4. <i>MinPosition</i> .....	5
2.2.5. <i>MaxPosition</i> .....	5
2.2.6. <i>Relationships Between State Variables</i> .....	5
2.3. EVENTING AND MODERATION .....	5
2.4. ACTIONS.....	6
2.4.1. <i>GetMode</i> .....	6
2.4.2. <i>SetMode</i> .....	7
2.4.3. <i>GetPosition</i> .....	7
2.4.4. <i>GetPositionTarget</i> .....	8
2.4.5. <i>SetPosition</i> .....	9
2.4.6. <i>GetMinMax</i> .....	9
2.4.7. <i>SetMinMax</i> .....	10
2.4.8. <i>Non-Standard Actions Implemented by an UPnP Vendor</i> .....	10
2.4.9. <i>Relationships Between Actions</i> .....	11
2.4.10. <i>Common Action Error Codes</i> .....	11
2.5. THEORY OF OPERATION .....	11
<b>3. XML SERVICE DESCRIPTION .....</b>	<b>12</b>
<b>4. TEST.....</b>	<b>15</b>

## List of Tables

Table 1: State Variables .....	4
Table 2: Relationship Between State Variables .....	5
Table 3: Event Moderation.....	5
Table 4: Actions .....	6
Table 5: Arguments for <i>GetMode</i> .....	6
Table 6: Arguments for <i>SetMode</i> .....	7
Table 7: Arguments for <i>GetPosition</i> .....	7
Table 8: Arguments for <i>GetPositionTarget</i> .....	8
Table 9: Arguments for <i>SetPosition</i> .....	9
Table 10: Arguments for <i>GetMinMax</i> .....	9
Table 11: Arguments for <i>SetMinMax</i> .....	10

## 1. Overview and Scope

This service definition is compliant with the UPnP Device Architecture version 1.0. It defines a service type referred to herein as ControlValve:1

ControlValve:1 provides programmatic control and status information for modulating water control valves or modulating air dampers used in Heating Ventilation and Air-Conditioning (HVAC) applications.

ControlValve:1 enables the following functions:

- Analog (modulating) control of Control Valves and Air Dampers

ControlValve:1 does not address:

- On/Off control valves or dampers
- Fire or Smoke dampers

### 1.1. Change Log

[04 Sep 2000]	v0.1	Document created
[18 Oct 2000]	v0.12	Updates based on Roseville review.
[12 Dec 2000]	v0.7	Clarification of the meaning of position, minor editorial corrections, and a new error return code for SetMinMax added. Layout adapted to Service Template v1.0 and Checklist v0.9
[14 Dec 2000]	v0.81	Moved to Template Design Complete.
[23 Jan 2001]	v0.82	Corrections based on checklist v0.9, structure adapted to template v1.01 and Argument names changed to be different from Related State Variable names.
[23 Feb 2001]	v0.83	GetPositionTarget added.
[19 Apr 2001]	v0.87	Version number increased to match remainder of the HVAC services set. No editorial changes...
[22 Aug 2001]	v0.88	XML for MaxPosition corrected.
[31 May 2002]	v0.9	Revision marks removed. Test chapter added.
[13 May 2003]	v1.0	Converted to Approved Standard

## 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:ControlValve:1**

The shorthand ControlValve:1 is used herein to refer to this service type.

## 2.2. State Variables

Defines the state variables for the operating mode of the control valve / air damper, its target position, and its actual position. Additionally defines optional state variables for “soft” minimum and maximum positions.

NOTE: (Explanation of the meaning of positions): Table 1 below describes Allowed Value ranges of 0 to 100, which signify a control valve / air damper position in the range of 0% to 100%. In all such cases, a value of 0% corresponds to the FULLY CLOSED control valve position (minimum mechanical limit of the actuator), and a value of 100% corresponds to the FULLY OPEN control valve position (maximum mechanical limit of the actuator).

**Table 1: State Variables**

Variable Name	Req. or Opt. <sup>1</sup>	Data Type	Allowed Value <sup>2</sup>	Default Value <sup>2</sup>	Eng. Units
ControlMode	R	string	OPEN, CLOSED, AUTO	CLOSE D	n/a
PositionTarget	R	ui1	>= 0, <= 100, += 1	0	Percent
PositionStatus	R	ui1	>= 0, <= 100, += 1	0	Percent
MinPosition	O	ui1	>= 0, <= 100, += 1	0	Percent
MaxPosition	O	ui1	>= 0, <= 100, += 1	100	Percent
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	<i>X</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

<sup>1</sup> R = Required, O = Optional, X = Non-standard.

<sup>2</sup> Values listed in this column are (all) required.

### 2.2.1. ControlMode

Determines the control mode. When ControlMode is OPEN, the device should motor to the 100% (fully open) position; when ControlMode is CLOSED, it should motor to the 0% (fully closed) position; and when ControlMode is AUTO the device position will be dependent on the value of the PositionTarget and the MinPosition and MaxPosition variables – described below.

### 2.2.2. PositionTarget

Determines the desired position of the device when ControlMode is AUTO. (See above Note “Explanation of the meaning of positions”)

### 2.2.3. PositionStatus

Measures the actual physical position of the device. (See above Note “Explanation of the meaning of positions”)

### 2.2.4. MinPosition

Determines the minimum position that the device may take when ControlMode is AUTO. (See above Note “Explanation of the meaning of positions”). MinPosition is the “soft” minimum position, which is typically  $> 0\%$  - where  $0\%$  is the “hard” minimum position.

MinPosition is an optional state variable; if MinPosition is not implemented in the device, then the device must behave as if the value of MinPosition were to be  $0\%$

### 2.2.5. MaxPosition

Determines the maximum position of that the device may take when ControlMode is AUTO. (See above Note “Explanation of the meaning of positions”). MaxPosition is the “soft” maximum position, which is typically  $< 100\%$  - where  $100\%$  is the “hard” maximum position.

MaxPosition is an optional state variable; if MaxPosition is not implemented in the device, then the device must behave as if the value of MaxPosition were to be  $100\%$

### 2.2.6. Relationships Between State Variables

The actual position (PositionStatus) of the device must follow the table below:

**Table 2: Relationship Between State Variables**

ControlMode	PositionTarget	PositionStatus
CLOSED	any	$0\%$ (i.e. “hard” minimum)
OPEN	any	$100\%$ (i.e. “hard” maximum)
AUTO	$\text{PositionTarget} < \text{MinPosition}$	MinPosition (i.e. “soft” minimum)
AUTO	$\text{MinPosition} < \text{PositionTarget} < \text{MaxPosition}$	PositionTarget
AUTO	$\text{PositionTarget} > \text{MaxPosition}$	MaxPosition (i.e. “soft” maximum)

If the values of ControlMode, PositionTarget, MinPosition or MaxPosition change, then the device should start to change its physical position towards the new value determined by the above table. Due to the physical running time of the motor /actuator, this process will take a certain period of time that depends on the vendor’s implementation. The value of the PositionStatus state variable should correspond to the actual physical device position.

**NOTE:** Vendors that implement control point strategies should bear in mind that due to friction, inertia, hysteresis and numerical rounding it is quite possible that the PositionStatus variable will take an indeterminate time to reach the value of the corresponding PositionTarget variable. Indeed it is quite likely that PositionStatus variable might *never* achieve exactly the same value as the PositionTarget variable.

*Relationships between standard state variable(s) defined herein and any non-standard state variable(s) is TBD.*

## 2.3. Eventing and Moderation

**Table 3: Event Moderation**

Variable Name	Ev
---------------	----

### 2.4.1.3. Effect on State

None.

### 2.4.1.4. Errors

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
800-899	TBD	(Specified by UPnP vendor.)

## 2.4.2. SetMode

Sets the new value of ControlMode.

### 2.4.2.1. Arguments

**Table 6: Arguments for SetMode**

Argument	Direction	relatedStateVariable
NewControlMode	IN	ControlMode

### 2.4.2.2. Dependency on State

None.

### 2.4.2.3. Effect on State

Sets the new value of ControlMode. This may influence PositionStatus according to Table 2.

### 2.4.2.4. Errors

errorCode	ErrorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
800-899	TBD	(Specified by UPnP vendor.)

## 2.4.3. GetPosition

Returns the current value of value of PositionStatus

### 2.4.3.1. Arguments

**Table 7: Arguments for GetPosition**

Argument	Direction	relatedStateVariable
CurrentPositionStatus	OUT <sup>R</sup>	PositionStatus

<sup>R</sup> = Return Value (RETVAl)

#### ***2.4.3.2. Dependency on State***

Returns the current value of PositionStatus.

#### ***2.4.3.3. Effect on State***

None.

#### ***2.4.3.4. Errors***

errorCode	ErrorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.



### 2.4.5. SetPosition

Sets the new value of PositionTarget.

#### 2.4.5.1. Arguments

**Table 9: Arguments for SetPosition**

Argument	Direction	relatedStateVariable
NewPositionTarget	IN	PositionTarget

#### 2.4.5.2. Dependency on State

None.

#### 2.4.5.3. Effect on State

Sets the new value of PositionTarget. After a certain period of time, which depends on the physical speed of the vendor's actuator, the value of PositionStatus should (normally) become the same as PositionTarget according to Table 2.

#### 2.4.5.4. Errors

errorCode	ErrorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
800-899	TBD	(Specified by UPnP vendor.)

### 2.4.6. GetMinMax

Gets the current values of MinPosition and MaxPosition.

#### 2.4.6.1. Arguments

**Table 10: Arguments for GetMinMax**

Argument	Direction	relatedStateVariable
CurrentMinPosition	OUT	MinPosition
CurrentMaxPosition	OUT	MaxPosition

#### 2.4.6.2. Dependency on State

Returns the current values of MinPosition and MaxPosition.

#### 2.4.6.3. Effect on State

None.

#### 2.4.6.4. Errors

errorCode	ErrorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
800-899	TBD	(Specified by UPnP vendor.)

### 2.4.7. SetMinMax

Sets new values of MinPosition and MaxPosition.

#### 2.4.7.1. Arguments

**Table 11: Arguments for SetMinMax**

Argument	Direction	relatedStateVariable
NewMinPosition	IN	MinPosition
NewMaxPosition	IN	MaxPosition

#### 2.4.7.2. Dependency on State

None.

#### 2.4.7.3. Effect on State

Sets new values of MinPosition and MaxPosition. This may influence PositionStatus according to Table 2.

#### 2.4.7.4. Errors

In addition to the individual range constraints on newMinPosition and newMaxPosition (valid range 0 to 100), there is a further combination range constraint: The value of newMinPosition must be less than the value of newMaxPosition. In case that SetMinMax is called with newMinPosition  $\geq$  newMaxPosition, the action must fail with an error code 701

errorCode	errorDescription	Description
402	Invalid Args	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
701	Min Exceeds Max	Requested value for MinPosition is greater than (or equal to) the requested value for MaxPosition
800-899	TBD	(Specified by UPnP vendor.)

### 2.4.8. Non-Standard Actions Implemented by an UPnP Vendor

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

### 2.4.9. Relationships Between Actions

The actions defined herein may be called in any order.

*Relationships between standard action(s) defined herein and any non-standard action(s) is TBD.*

### 2.4.10. Common Action 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.

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.
800-899	<i>TBD</i>	<i>(Specified by UPnP vendor.)</i>

## 2.5. Theory of Operation

A Control Valve or Air Damper is used to control the amount of heating or cooling energy being provided into a building space.

Under normal operation, ControlMode will be AUTO. A control point, (e.g. a temperature controller), will modulate the position of the Control Valve or Air Damper by using SetPosition to change the value of PositionTarget. Under normal operation the value of PositionStatus will track PositionTarget.

A control point can call GetPosition to read the current position of the Control Valve or Air Damper.

Under special override conditions, a control point may use SetMode to change the value of ControlMode, for example as follows:

- i) ControlMode can be set to CLOSED when the air-conditioning or heating plant is shut down – this forces the device to a “hard” closed position.
- ii) ControlMode can be set to OPEN to provide a forced override to a “hard” open position – e.g. for emergency (smoke) ventilation.

A control point can call GetMode to read the current value of ControlMode.

The MinPosition and MaxPosition are optional parameters, provided to set “soft” limits on the range (stroke) of PositionStatus. A control point can call SetMinMax to set MinPosition and MaxPosition to new values. For example in the case of Air Dampers:

- i) MinPosition may be set to provide minimum fresh air requirements, and
- ii) MaxPosition may be set to prevent drafts.

It is not likely that SetMinMax would be called very frequently – e.g. in the above example, the minimum fresh air setting might be adjusted on a day /night basis.

A control point can call GetMinMax to read the current values of MinPosition and MaxPosition.

### 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>GetMode</name>
      <argumentList>
        <argument>
          <name>CurrentControlMode</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>ControlMode</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetMode</name>
      <argumentList>
        <argument>
          <name>NewControlMode</name>
          <direction>in</direction>
          <relatedStateVariable>ControlMode</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetPosition</name>
      <argumentList>
        <argument>
          <name>CurrentPositionStatus</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>PositionStatus</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetPositionTarget</name>
      <argumentList>
        <argument>
          <name>CurrentPositionTarget</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>PositionTarget</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>

```

```

<name>SetPosition</name>
  <argumentList>
    <argument>
      <name>NewPositionTarget</name>
      <direction>in</direction>
      <relatedStateVariable>PositionTarget</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>GetMinMax</name>
  <argumentList>
    <argument>
      <name>CurrentMinPosition</name>
      <direction>out</direction>
      <relatedStateVariable>MinPosition</relatedStateVariable>
    </argument>
    <argument>
      <name>CurrentMaxPosition</name>
      <direction>out</direction>
      <relatedStateVariable>MaxPosition</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>SetMinMax</name>
  <argumentList>
    <argument>
      <name>NewMinPosition</name>
      <direction>in</direction>
      <relatedStateVariable>MinPosition</relatedStateVariable>
    </argument>
    <argument>
      <name>NewMaxPosition</name>
      <direction>in</direction>
      <relatedStateVariable>MaxPosition</relatedStateVariable>
    </argument>
  </argumentList>
</action>
Declarations for other actions added by UPnP vendor (if any) go here
</actionList>
<serviceStateTable>
  <stateVariable sendEvents="yes">
    <name>ControlMode</name>
    <dataType>string</dataType>
    <defaultValue>CLOSED</defaultValue>
    <allowedValueList>
      <allowedValue>CLOSED</allowedValue>
      <allowedValue>OPEN</allowedValue>
      <allowedValue>AUTO</allowedValue>
    </allowedValueList>
  </stateVariable>
  <stateVariable sendEvents="yes">
    <name>PositionStatus</name>

```

```

    <dataType>uil</dataType>
    <defaultValue>0</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>100</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>PositionTarget</name>
    <dataType>uil</dataType>
    <defaultValue>0</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>100</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>MinPosition</name>
    <dataType>uil</dataType>
    <defaultValue>0</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>100</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>MaxPosition</name>
    <dataType>uil</dataType>
    <defaultValue>100</defaultValue>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>100</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  Declarations for other state variables added by UPnP vendor (if any)
  go here
</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: *ControlValve1.xml*
- The UPnP Test Tool service template test file: *ControlValve1.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.