HVAC_UserOperatingMode:1 Service Template

For UPnP™ Device Architecture V 1.0

Status: Standardized DCP
Date: May 13th, 2003

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.

Authors	Company
Larry Stickler	Honeywell
Pete Bergstrom	Honeywell
Andrew Fiddian-Green	Siemens Building Technologies

Contents

List of Tables

1. Overview and Scope

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

This service definition enables the following functions:

- Changing and reading the user operating modes of an HVAC system

1.1. Change Log for: HVAC_UserOperatingMode:1

7/24/00 Changes per 7/17 meeting of Home Automation and Security Working Group; conversion to 0.996 template

8/24/00 Changes to theory of operation, expanding the definition of reserved modes and explaining that implementations may subset the set of reserved mode values.

8/29/00 Added XML, Removed ModeList,

9/28/00 Corrected typos, added Name

2/14/01 Corrected capitalization, Set and Get Target changes to Set and Get ModeTarget, GetStatus changed to GetModeStatus, stand-by changed to standby

2/21/01 Moved to 1.1 Template, added idle mode

2/26/01 Proof read

3/11/01 Renamed Idle to InDeadBand and Standby to EnergySaving, added EnergySavingsHeading and EnergySavingsCooling as allowed values for ModeStatus

4/2/01 Moved to 0.87

8/22/01 XML corrected

5/31/02 Revision marks removed; Moved to 0.9; Test chapter added.

[13 May 2003] v1.0 Converted to Approved Standard.

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:HVAC_UserOperatingMode: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
ModeTarget	R	string	see table	Auto	N/a
ModeStatus	R	string	see table	none	none
Name	O	string		Zero length string	N/a
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.

Table 2 AllowedValueList for ModeTarget

Value	Req. or Opt. [1]
Off	*R*
HeatOn	*Either HeatOn or CoolOn or both is required*
CoolOn	*Either HeatOn or CoolOn or both is required*
AutoChangeOver	*O*
AuxHeatOn	*O*

EconomyHeatOn	*O*
EmergencyHeatOn	*O*
AuxCoolOn	*O*
EconomyCoolOn	*O*
BuildingProtection	*O*
EnergySavingsMode	*O*
Vendor-defined	*R*
Vendor-defined	*O*

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

Table 3 AllowedValueList for ModeStatus

Value	Req. or Opt. [1]
Off	*R*
InDeadBand	*R*
HeatOn	*Either HeatOn or CoolOn or both is required*
CoolOn	*Either HeatOn or CoolOn or both is required*
AutoChangeOver	*O*
AuxHeatOn	*O*
EconomyHeatOn	*O*
EmergencyHeatOn	*O*
AuxCoolOn	*O*
EconomyCoolOn	*O*
BuildingProtection	*O*
EnergySavingsHeating	*O*
EnergySavingsCooling	*O*
Vendor-defined	*R*
Vendor-defined	*O*

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

2.2.1. ModeTarget

Exposes the target operating mode of an HVAC system. Mode values are established by the manufacturer

2.2.2. ModeStatus

Exposes the current operating mode of an HVAC system. Mode values are established by the vendor

2.2.3. Name

This optional variable may be used to capture a friendly name or location for this service.

2.2.4. Relationships Between State Variables

ModeTarget provides a variable for a Control Point to request a new mode. ModeStatus is the current mode value. They may be different.

AutoChangeOver target mode enables the Modestatus to change between heating and cooling depending on demand.

2.3. Eventing and Moderation

Table 4 Eventing & Moderation

Variable Name	Evented	Moderated Event	Max Event Rate[1]	Logical Combination	Min Delta per Event[2]
Name	Yes	No	none	none	On-change
ModeTarget	Yes	No	none	none	On-Change
ModeStatus	Yes	No	none	none	On-change
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).

2.3.1. Event Model

Table 5 Event Model

Variable Name	UI requirements	Async Requirements	Func. Vs max rate tradeoffs	Est of Max rate	Reason not evented
Name	Needed for UI			On set-up only	N/a
ModeTarget	Needed for UI			Very low	N/a
ModeStatus	Needed for UI			Very low	N/a

2.4. Actions

Table 6 Action list

Name	Req. or Opt. [1]
SetModeTarget	*R*
GetModeTarget	*R*
GetModeStatus	*R*
GetName	O
SetName	O
Non-standard actions implemented by an UPnP vendor go here.	X

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

2.4.1. SetModeTarget

Changes the operating mode of the HVAC fan or blower.

2.4.1.1. Arguments

Table 7 Arguments for SetModeTarget

Argument	Direction	relatedStateVariable
NewModeTarget	*In*	ModeTarget

2.4.1.2. Dependency on State (if any)

None

2.4.1.3. Effect on State (if any)

Target mode changes to NewTarget

2.4.1.4. Errors

errorCode	errorDescription	Description
700	Mode not available	The requested mode is not available

2.4.2. GetModeTarget

Provides Mode information to control points or other devices

2.4.2.1. Arguments

Table 8 Arguments for GetModeTarget

Argument	Direction	relatedStateVariable
CurrentModeTarget	*Out*[R]	ModeTarget

R- Return Value

2.4.2.2. Dependency on State (if any)

Depends on ModeTarget

2.4.2.3. Effect on State (if any)

None

2.4.2.4. Errors

errorCode	errorDescription	Description
None		

2.4.3. GetModeStatus

Gets the current mode status

2.4.3.1. Arguments

Table 9 Arguments for GetModeStatus

Argument	Direction	relatedStateVariable
CurrentModeStatus	*Out*[R]	ModeStatus

R- Return Value

2.4.3.2. Dependency on State (if any)

Depends on ModeStatus.

2.4.3.3. Effect on State

None.

2.4.3.4. Errors

errorCode	errorDescription	Description
none		

2.4.4. GetName

Provides the Name value to a control point or other UPnP device

2.4.4.1. Arguments

Table 10 Arguments for GetName

Argument	Direction	relatedStateVariable
CurrentName	*Out*[R]	Name

[R] Return Value

2.4.4.2. Dependency on State (if any)

Depends on Name

2.4.4.3. Effect on State

None

2.4.4.4. Errors

errorCode	errorDescription	Description
none		

2.4.5. SetName

Provides a new value for the Name variable.

2.4.5.1. Arguments

Table 11 Arguments for SetName

Argument	Direction	relatedStateVariable
NewName	*In*	Name

2.4.5.2. Dependency on State (if any)

None

2.4.5.3. Effect on State

Changes Name

2.4.5.4. Errors

errorCode	errorDescription	Description
none		

2.4.6. 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).

2.4.7. Relationships Between Actions

None.

2.4.8. 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 6: 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

This service allows a Control Point to set and observe the operating mode of a HVAC system. Reserved operating modes are:

- Off – System not active
- InDeadBand – System active but not currently heating or cooling.
- Heating – controlling to a heating setpoint
- Cooling – controlling to a cooling setpoint
- AutoChangeOver – Heating or cooling depending on demand and setpoints.
- Emergency Heat – Often used with Heat Pumps to provide heat from a secondary source.
- AuxHeatOn – see Emergency heat

- AuxCooling- Used when a secondary cooling mechanism is available.
- EconomyHeatOn – controlling to a heating setpoint that is less than the current heating temperature setpoint. The delta value is implementation dependent.
- EconomyCoolingOn - controlling to a cooling setpoint that is more than the current cooling temperature setpoint. The delta value is implementation dependent.
- BuildingProtection – controlling to a default temperature that is intended to keep the water pipes and fixtures from freezing.
- EnergySavings – controlling to default temperatures that are–less (for heating) or more (for cooling) than normal at-home temperatures.

Different vendors and different geographies employ different modes of operation. This service allows vendors to implement a subset of the total set of reserved mode values.

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>SetModeTarget</name>
      <argumentList>
        <argument>
          <name>NewModeTarget</name>
          <direction>in</direction>
<relatedStateVariable>ModeTarget</relatedStateVariable>
        </argument>
      </argumentList>
    </action>

<actionList>
    <action>
    <name>GetModeTarget</name>
      <argumentList>
        <argument>
          <name>CurrentModeTarget</name>
          <direction>out</direction>
          <retval/>
          <relatedStateVariable>ModeTarget</relatedStateVariable>
        </argument>
      </argumentList>
    </action>

<actionList>
    <action>
    <name>GetModeStatus</name>
      <argumentList>
        <argument>
          <name>CurrentModeStatus</name>
          <direction>out</direction>
            <retval/>
          <relatedStateVariable>ModeStatus</relatedStateVariable>
        </argument>
      </argumentList>
    </action>

<action>
    <name>GetName</name>
      <argumentList>
        <argument>
          <name>CurrentName</name>
          <direction>out</direction>
          <retval/>
          <relatedStateVariable>Name</relatedStateVariable>
        </argument>
      </argumentList>
    </action>

    <action>
    <name>SetName</name>
      <argumentList>
        <argument>
          <name>NewName</name>
          <direction>in</direction>
          <relatedStateVariable>Name</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    Declarations for other actions added by UPnP vendor (if any) go here
  </actionList>

  <serviceStateTable>
    <stateVariable sendEvents="yes">
      <name>ModeTarget</name>
      <dataType>string</dataType>
      <defaultValue>Off</defaultValue>
      <allowedValueList>
       <allowedValue>Off</allowedValue>
        HeatOn or CoolOn or both state variables are required
       <allowedValue>HeatOn</allowedValue>
       <allowedValue>CoolOn</allowedValue>
      The following state variables are optional
      <allowedValue>AutoChangeOver</allowedValue>
        <allowedValue>AuxHeatOn</allowedValue>
        <allowedValue>EconomyHeatOnallowedValue>
        <allowedValue>EmergencyHeatOn</allowedValue>
        <allowedValue>EconomyCoolOn</allowedValue>
        <allowedValue>AuxCoolOn</allowedValue>
        <allowedValue>BuildingProtection</allowedValue>
        <allowedValue>EnergySavingsMode</allowedValue>
      </allowedValueList>
    </stateVariable>

   <serviceStateTable>
    <stateVariable sendEvents="yes">
      <name>ModeStatus</name>
      <dataType>string</dataType>
      <allowedValueList>
        <allowedValue>Off</allowedValue>
        <allowedValue>InDeadBand</allowedValue>
        HeatOn or CoolOn or both state variables are required
       <allowedValue>HeatOn</allowedValue>
       <allowedValue>CoolOn</allowedValue>
       The following state variables are optional
        <allowedValue>AuxHeatOn</allowedValue>
        <allowedValue>AutoChangeOver</allowedValue>
        <allowedValue>EconomyHeatOn</allowedValue>
        <allowedValue>EmergencyHeatOn</allowedValue>
        <allowedValue>EconomyCoolOn</allowedValue>
        <allowedValue>AuxCoolOn</allowedValue>
        <allowedValue>BuildingProtection</allowedValue>
        <allowedValue>EnergySavingsCooling</allowedValue>
        <allowedValue>EnergySavingsHeating</allowedValue>
      </allowedValueList>
    </stateVariable>

      <stateVariable sendEvents="yes">
      <name>Name</name>
      <dataType>string</dataType>

    </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: ***HVAC_UserOperatingMode1.xml***
- The UPnP Test Tool service template test file: ***HVAC_UserOperatingMode1.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.