

# HVAC\_SetpointSchedule: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
Larry Stickler	Honeywell
Pete Bergstrom	Honeywell
Andrew Fiddian-Green	Siemens Building Technologies

## Contents

<b>1. OVERVIEW AND SCOPE .....</b>	<b>3</b>
1.1. CHANGE LOG FOR: HVAC_SETPOINTCHEDULE .....	3
<b>2. SERVICE MODELING DEFINITIONS .....</b>	<b>4</b>
2.1. SERVICE TYPE .....	4
2.2. STATE VARIABLES .....	4
2.2.1. A_ARG_TYPE_DayOfWeek .....	5
2.2.2. A_ARG_TYPE_EventName .....	6
2.2.3. A_ARG_TYPE_StartTime .....	6
2.2.4. A_ARG_TYPE_HeatingSetpoint .....	6
2.2.5. A_ARG_TYPE_CoolingSetpoint .....	6
2.2.6. EventsPerDay .....	6
2.2.7. Relationships Between State Variables .....	6
2.3. EVENTING AND MODERATION .....	7
2.3.1. Event Model .....	7
2.4. ACTIONS .....	8
2.4.1. SetEventParameters .....	8
2.4.2. GetEventsPerDay .....	9
2.4.3. Non-Standard Actions Implemented by a UPnP Vendor .....	10
2.4.4. Relationships Between Actions .....	10
2.4.5. Common Error Codes .....	10
2.5. THEORY OF OPERATION .....	11
<b>3. XML SERVICE DESCRIPTION .....</b>	<b>13</b>
<b>4. TEST .....</b>	<b>16</b>

## List of Tables

Table 1 State Variables .....	4
Table 2 AllowedValueList for A_ARG_TYPE_DayOfWeek .....	4
Table 3 AllowedValueList for A_ARG_TYPE_EventName .....	5
Table 4 Eventing & Moderation .....	7
Table 5 Event Model .....	7
Table 6 Values of EventsPerDay String .....	8
Table 7 Action list .....	8
Table 8 Arguments for SetEventParameters .....	9
Table 9 Arguments for GetEventsPerDay .....	9
Table 10 Common Error Codes .....	10
Table 11 Example Table .....	12

# 1. Overview and Scope

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

This service provides the following variables:

- A\_ARG\_TYPE\_DayOfWeek variable indicating the the day for which a set of timed setpoints is established.
- A\_ARG\_TYPE\_EventName variable indicating the named events for which timed setpoints may be established.
- A\_ARG\_TYPE\_StartTime variable indicating the starting time of a scheduled change in setpoints.
- A\_ARG\_TYPE\_HeatingSetpoint variable to be part of a scheduled change in a heating setpoint
- A\_ARG\_TYPE\_CoolingSetpoint variable to be part of a scheduled change in a cooling setpoint
- EventsPerDay a delimited string indicating the scheduled setpoint changes on a given day

This service provides the following actions:

- SetEventParameters establishes a new or updates an existing event with the appropriate parameters.
- GetEventsPerDay returns a list of the events for the specified day of week.

This service does not provide the following:

- The interface between the schedule table and setpoint objects or services

## 1.1. Change Log for: HVAC\_SetpointSchedule

7/24	Changes per 7/17 meeting of Home Automation and Security Working Group; conversion to 0.996 template
8/18	Clean up per Steve Timm's (7/2700) suggestions
11/29/00	Moved to Template Design Complete, re-named DayList to EventsPerDay, added AccessToken argument, expanded theory of operation, added XML
2/14/01	Corrected per review TDC checklist review. Corrected capitalization, expanded overview, added relationships between variables, corrected XML, expanded theory of operation
2/21/01	Moved to 1.1 Template, added Sunrise and Sunset as optional events.
2/26/01	Proof read
3/8/01	Corrected variables to be A_ARG_TYPE's
4/2/01	Changed diagram in Theory of Operation, moved to 0.87
7/27/01	Capitalization of the word "Setpoint" harmonized. 0.88
8/8/01	Explanation of the different contents of EventsPerDay for a) eventing and b) GetEventsPerDay. Font and other formatting tidied up. Version 0.89
8/16/01	Wild card symbol "*" added to allowed values of A_ARG_TYPE_DayOfWeek. Version 0.891

[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. ServiceType

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

**urn:schemas-upnp-org:service:HVAC\_SetpointSchedule:1**

### 2.2. State Variables

**Table 1 State Variables**

Variable Name	Req. or Opt.	Data Type	Allowed Value <sup>2</sup>	Default Value <sup>2</sup>	Eng. Units
A_ARG_TYPE_DayOfWeek	R	string	See Table 2	All	none
A_ARG_TYPE_EventName	R	string	See Table 3	Home	none
A_ARG_TYPE_StartTime	R	ui2	Minimum = 0 Maximum <= 1439 Step = 1	0	Minutes from mid night
A_ARG_TYPE_HeatingSetpoint	R	i4	Minimum: vendor defined Maximum: vendor defined Step=1	Vendor defined	.01 degrees Celsius
A_ARG_TYPE_CoolingSetpoint	R	i4	Minimum: vendor defined Maximum: vendor defined Step=1	Vendor defined	.01 degrees Celsius
EventsPerDay	R	string	N/a	Zero length string	none
<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 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 A\_ARG\_TYPE\_DayOfWeek**

Value	Req. or Opt. <sup>1</sup>
-------	---------------------------

Sun	O
Mon	O
Tue	O
Wed	O
Thu	O
Fri	O
Sat	O
All	R
Weekdays	O
Weekend	O
Standby	O
*	R
<i>Vendor-defined</i>	<i>R/O</i>

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

**Table 3 AllowedValueList for A\_ARG\_TYPE\_EventName**

Value	Req. or Opt. <sup>1</sup>
Home	R
Wake	R
Sleep	R
Away	R
Sunrise	O
Sunset	O
<i>Vendor-defined</i>	<i>R/O</i>

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

### 2.2.1. A\_ARG\_TYPE\_DayOfWeek

Defined only to provide typing for one or more action arguments. This variable serves as the primary index to the schedule list. DayOfWeek values are in general defined by the manufacturer, however there are two required values:

- “All” – this is for the basic functionality of a one-day schedule i.e. when all days of the week follow the same time schedule.
- “\*” (wildcard symbol) – this is a special symbol that used only in the GetEventsPerDay action for retrieving the complete schedule in a single operation.

### 2.2.2. A\_ARG\_TYPE\_EventName

Defined only to provide typing for one or more action arguments. This variable is the secondary index to the list. EventNames are established by the manufacturer.

### 2.2.3. A\_ARG\_TYPE\_StartTime

Defined only to provide typing for one or more action arguments. This variable provides the time from midnight (in minutes) to the start of an event

### 2.2.4. A\_ARG\_TYPE\_HeatingSetpoint

Defined only to provide typing for one or more action arguments. This variable provides the heating setpoint for this event

### 2.2.5. A\_ARG\_TYPE\_CoolingSetpoint

Defined only to provide typing for one or more action arguments. This variable provides the heating setpoint for this event.

### 2.2.6. EventsPerDay

This string variable is a comma-delimited list of the events, start times, heating setpoints, and cooling setpoints for a specific day, or for a specific event.

### 2.2.7. Relationships Between State Variables

This service generates a list of tuples (StartTime, CoolingSetpoint, HeatingSetpoint) indexed by A\_ARG\_TYPE\_DayOfWeek and by A\_ARG\_TYPE\_EventName

An event is set using the SetEventParameters action. This action uses five arguments: (SubmittedDayOfWeek, SubmittedEventName, NewStartTime, NewHeatingSetpoint, NewCoolingSetpoint).

This allows a Control Point or other devices to set start times and both heating and cooling setpoints for each event in each day of the week.

The tuples for a given DayOfWeek may be read using the GetEventsPerDay action. Here the Control Point submits a DayOfWeek and the service returns all of the scheduled events for the submitted DayOfWeek along with their associated tuples as a delimited string.

## 2.3. Eventing and Moderation

Table 4 Eventing & Moderation

Variable Name	Evented	Moderated Event	Max Event Rate <sup>1</sup>	Logical Combination	Min Delta per Event <sup>2</sup>
A_ARG_TYPE_DayOfWeek	No				
A_ARG_TYPE_EventName	No				
A_ARG_TYPE_StartTime	No				
A_ARG_TYPE_HeatingSetpoint	No				
A_ARG_TYPE_CoolingSetpoint	No				
EventsPerDay	Yes	No			
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

<sup>1</sup> Determined by N, where Rate = (Event)/(N secs).

<sup>2</sup> (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
A_ARG_TYPE_DayOfWeek					Variable used is an argument only
A_ARG_TYPE_EventName					Variable is reported as part of a return value
A_ARG_TYPE_StartTime					Variable is reported as part of a return value
A_ARG_TYPE_HeatingSetpoint					Variable is reported as part of a return value
A_ARG_TYPE_CoolingSetpoint					Variable is reported as part of a return value
EventsPerDay	UI needs to know when a				

Variable Name	UI requirements	Async Requirements	Func. Vs max rate tradeoffs	Est of Max rate	Reason not evented
	scheduled event is added, removed or changed				

The service must send event messages for the EventsPerDay state variable whenever a) a new event is added to the setpoint schedule, b) an existing event is modified, or c) an existing event is deleted. Event messages must be sent for EventsPerDay in response to a) changes made via the SetEventParameters action, and b) changes made through another user interface.

EventsPerDay returns a comma-delimited string for a *single* event in the form defined in the following table below. NOTE: This differs from the behavior of EventsPerDay under the UPnP Control Action “GetEventsPerDay” (paragraph 2.4.2) where *zero or more* events are concatenated.

If a series of individual changes are made in the setpoint schedule, then a corresponding series of event messages must be transmitted.

**Table 6 Values of EventsPerDay String**

Cause of Event	Value of EventsPerDay
New event is added	Day,Event,StartTime,HeatingSetpoint,CoolingSetpoint
Existing event is modified	Day,Event,newStartTime,newHeatingSetpoint,newCoolingSetpoint
Existing event is deleted	Day,Event,0,0,0

## 2.4. Actions

**Table 7 Action list**

Name	Req. or Opt. <sup>1</sup>
SetEventParameters	Req
GetEventsPerDay	Req
<i>Non-standard actions implemented by an UPnP vendor go here.</i>	<i>X</i>

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

### 2.4.1. SetEventParameters

This action establishes a new or overwrites an existing event with the appropriate parameters. It is event and day of week specific.

If NewStartTime is 0, the event is removed from the list.



### 2.4.1.1. Arguments

**Table 8 Arguments for SetEventParameters**

Argument	Direction	relatedStateVariable
SubmittedDayOfWeek	In	A_ARG_TYPE_DayOfWeek
SubmittedEventName	In	A_ARG_TYPE_EventName
NewStartTime	In	A_ARG_TYPE_StartTime
NewHeatingSetpoint	In	A_ARG_TYPE_HeatingSetpoint
NewCoolingSetpoint	In	A_ARG_TYPE_CoolingSetpoint

### 2.4.1.2. Dependency on State

None

### 2.4.1.3. Effect on State

Changes the respective set point schedule entry.

### 2.4.1.4. Errors

errorCode	errorDescription	Description
700	Day of Week not available	Request for invalid DayOfWeek
701	EventName not available	Request for invalid EventName

## 2.4.2. GetEventsPerDay

This action returns a comma-delimited string in the form:

Day,Event1,StartTime1,HeatingSetpoint1,CoolingSetpoint1,Day,Event2,StartTime2,HeatingSetpoint2,CoolingSetpoint2, ... until all events in the specified DayOf Week have been reported.

In other words, the action returns a concatenation of **zero or more** events. NOTE: This differs from the behavior of EventsPerDay under the UPnP Event Model (see paragraph 2.3.1) where a **single** event is transmitted.

If there are no events matching the specified DayOfWeek, then the action returns a zero length string.

If SubmittedDayOfWeek is "\*" (wildcard symbol), then the action returns a concatenation of **all** events in the schedule.

### 2.4.2.1. Arguments

**Table 9 Arguments for GetEventsPerDay**

Argument	Direction	relatedStateVariable
SubmittedDayOfWeek	In	A_ARG_TYPE_DayOfWeek
CurrentEventsPerDay	Out <sup>R</sup>	EventsPerDay

<sup>R</sup> - Return Value

#### 2.4.2.2. *Dependency on State*

CurrentEventsPerDay must return the full comma-delimited list of all the events in the setpoint schedule that match the respective SubmittedDayOfWeek (i.e. zero or more events).

#### 2.4.2.3. *Effect on State*

None

#### 2.4.2.4. *Errors*

errorCode	errorDescription	Description
700	Day of Week not available	Request for invalid DayOfWeek

### 2.4.3. 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).

### 2.4.4. Relationships Between Actions

None.

### 2.4.5. 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 10 Common Error Codes**

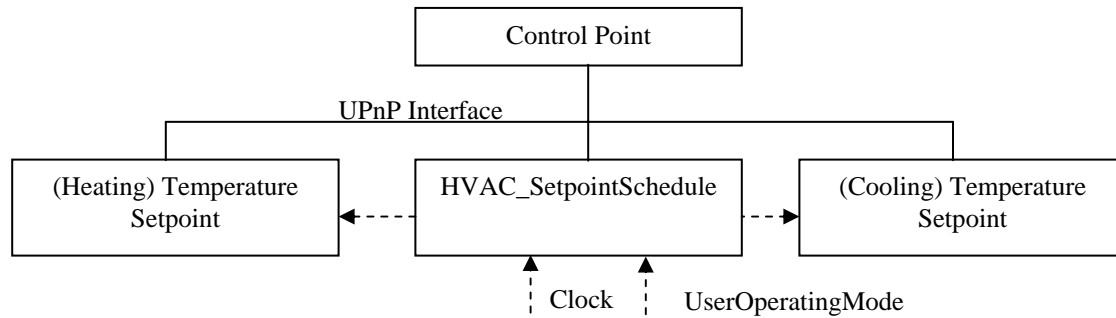
errorCode	errorDescription	Description
401	Invalid Action	See UPnP Device Architecture section on Control.
402	Invalid Args	See UPnP Device Architecture section on Control.
404	Invalid Var	See UPnP Device Architecture section on Control.
501	Action Failed	See UPnP Device Architecture section on Control.
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 provides a means to establish a logical schedule table where each row represents a separate event. Control points may add, remove or modify an event by using the SetEventParameters action. Control points may discover the events established for a given period by using the GetEventPerDay action.

“All” events repeat all days of the week, each day of the week, weekly or on weekends.

The logic or connection to the temperature setpoint controllers is not exposed by this design. It is presumed that internal actions of an implementation of this service will cause the appropriate setpoints to



be modified.

Note: Dotted lines indicate internal interfaces and are not necessarily visible from the UPnP interface

The manufacturer establishes the Day of Week list and the EventName list. The list of allowed Day of Week values is a superset of what may be implemented by a particular vendor. The list of EventNames may be extended by a manufacturer. The allowed-value XML description provides the implemented days and event names for Control Points.

This service generates a list or table of tuples (StartTime, CoolingSetpoint, HeatingSetpoint) indexed by A\_ARG\_TYPE\_DayOfWeek and by A\_ARG\_TYPE\_EventName

An event is set using the SetEventParameters action. This action uses five arguments: (SubmittedDayOfWeek, SubmittedEventName, NewStartTime, NewHeatingSetpoint, NewCoolingSetpoint).

This allows a Control Point or other devices to set start times and both heating and cooling setpoints for each event in each day of the week.

The tuples for a given DayOfWeek may be read using the GetEventsPerDay action. Here the Control Point submits a DayOfWeek and the service returns all of the scheduled events for the submitted DayOfWeek along with their associated tuples as a delimited string.

Table 11 Example Table

DayOfWeek	EventName	StartTime	HeatingSetpoint	CoolingSetpoint
Mon	Wake	440	2065	2389
Mon	Leave	540	1833	2667
Mon	Home	1020	2222	2389
Mon	Sleep	1320	1833	2389
Tue	Wake	440	2222	2389
Tue	Sleep	1320	1833	2389
Wed	Wake	440	2222	2389
Wed	Leave	540	1833	2667
Wed	Home	1020	2222	2389
Wed	Sleep	1320	1833	2389
Thu	Wake	440	2222	2389
Thu	Sleep	1320	1833	2389
Fri	Wake	440	2222	2389
Fri	Sleep	1320	1833	2389
Weekend	Wake	540	2222	2389
Weekend	Sleep	1320	1833	2389

Example response to GetEventsPerDay (Tue)

Tue,Wake,440,2222,2389,Tue,Sleep,1320,1833,2389

### 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>SetEventParameters</name>
      <argumentList>
        <argument>
          <name>SubmittedDayOfWeek</name>
          <direction>in</direction>

          <relatedStateVariable>A ARG TYPE DayOfWeek</relatedStateVariable>
        </argument>
        <argument>
          <name>SubmittedEventName</name>
          <direction>in</direction>

          <relatedStateVariable>A ARG TYPE EventName</relatedStateVariable>
        </argument>
        <argument>
          <name>NewStartTime</name>
          <direction>in</direction>
          <relatedStateVariable>A ARG TYPE StartTime</relatedStateVariable>
        </argument>
        <argument>
          <name>NewHeatingSetpoint</name>
          <direction>in</direction>
          <relatedStateVariable>A ARG TYPE HeatingSetpoint</relatedStateVariable>
        </argument>
        <argument>
          <name>NewCoolingSetpoint</name>
          <direction>in</direction>
          <relatedStateVariable>A ARG TYPE CoolingSetpoint</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>GetEventsPerDay</name>
      <argumentList>
        <argument>
          <name>SubmittedDayOfWeek</name>
          <direction>in</direction>
          <relatedStateVariable>A ARG TYPE DayOfWeek</relatedStateVariable>
        </argument>
        <argument>
          <name>CurrentEventsPerDay</name>
          <direction>out</direction>
          <retval />
          <relatedStateVariable>EventsPerDay</relatedStateVariable>
        </argument>
      </argumentList>
    </action>
  </actionList>
</scpd>

```

*Declarations for other actions added by UPnP vendor (if any) go here*

```
</actionList>
<serviceStateTable>
  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_DayOfWeek</name>
    <dataType>string</dataType>
    <defaultValue>All</defaultValue>
    <allowedValueList>
      <allowedValue>All</allowedValue>
      <allowedValue>*</allowedValue>
```

*The following allowed values are optional*

```
      <allowedValue>Sun</allowedValue>
      <allowedValue>Mon</allowedValue>
      <allowedValue>Tue</allowedValue>
      <allowedValue>Wed</allowedValue>
      <allowedValue>Thu</allowedValue>
      <allowedValue>Fri</allowedValue>
      <allowedValue>Sat</allowedValue>
      <allowedValue>Weekend</allowedValue>
      <allowedValue>Weekdays</allowedValue>
```

*Other allowed values defined by UPnP vendors (if any) go here*

```
</allowedValueList>
</stateVariable>
<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_EventName</name>
  <dataType>string</dataType>
  <defaultValue>Home</defaultValue>
  <allowedValueList>
    <allowedValue>Home</allowedValue>
    <allowedValue>Wake</allowedValue>
    <allowedValue>Sleep</allowedValue>
    <allowedValue>Away</allowedValue>
```

*The following allowed values are optional*

```
      <allowedValue>Sunrise</allowedValue>
      <allowedValue>Sunset</allowedValue>
```

*Other allowed values defined by UPnP vendors (if any) go here*

```
</allowedValueList>
</stateVariable>
<stateVariable sendEvents="no">
  <name>A_ARG_TYPE_StartTime</name>
  <dataType>ui2</dataType>
  <defaultValue>0</defaultValue>
  <allowedValueRange>
    <minimum>0</minimum>
    <maximum>1439</maximum>
    <step>1</step>
  </allowedValueRange>
```

```

</stateVariable>
  <stateVariable sendEvents="no">
    <name>A ARG TYPE HeatingSetpoint</name>
    <dataType>i4</dataType>
    <defaultValue>vendor defined</defaultValue>
    <allowedValueRange>
      <minimum>vendor defined</minimum>
      <maximum>vendor defined</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>A ARG TYPE CoolingSetpoint</name>
    <dataType>i4</dataType>
    <defaultValue>vendor defined</defaultValue>
    <allowedValueRange>
      <minimum>vendor defined</minimum>
      <maximum>vendor defined</maximum>
      <step>1</step>
    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>EventsPerDay</name>
    <dataType>string</dataType>
    <defaultValue></defaultValue> ; zero length string
  </stateVariable>

  Declarations for other state variables added by UPnP vendors (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\_SetpointSchedule1.xml*
- The UPnP Test Tool service template test file: *HVAC\_SetpointSchedule1.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.