

Audio:1 Sample Service Template

For UPnP™ Version 1.0

Status: Preliminary Design (TPD)

Date: ~~23 August 2000~~ 5 February 2001

This sample service template is being made available to UPnP Members pursuant to Section 2.1(c)(ii) of the UPnP Membership Agreement for review and comment by Members to the UPnP Steering Committee regarding the Steering Committee's consideration of the Proposed template as a Standardized service. Pursuant to Section 3.1 of the UPnP Membership Agreement, Member has limited rights to use or reproduce the Proposed template during the comment period and only in furtherance of this review and comment. 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 PROPOSED TEMPLATES, IMPLEMENTATIONS OR IN ANY ASSOCIATED TEST SUITES. THE SAMPLE SERVICE TEMPLATE IS PROVIDED "AS IS" AND "WITH ALL FAULTS". THE UPNP FORUM MAKES NO WARRANTIES, EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE PROPOSED SERVICE TEMPLATE 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.

© 2000-~~2001~~ Contributing Members of the UPnP™ Forum. All Rights Reserved.

Author	Company
Jeffrey Schlimmer	Microsoft

Contents

1. OVERVIEW AND SCOPE.....	3
1.1. CHANGE LOG.....	3
2. SERVICE MODELING DEFINITIONS.....	4
2.1. SERVICE TYPE.....	4
2.2. STATE VARIABLES.....	4
2.2.1. <i>Volume</i>	4
2.2.2. <i>Treble</i>	4
2.2.3. <i>Bass</i>	4
2.2.4. <i>Balance</i>	5
2.2.5. <i>Fade</i>	5
2.2.6. <i>A_ARG_TYPE_Delta</i>	5
2.2.7. <i>Non-Standard State Variables Implemented by an UPnP Vendor</i>	5
2.2.8. <i>Relationships Between State Variables</i>	5
2.3. EVENTING AND MODERATION	5
2.3.1. <i>Event Model</i>	6
2.4. ACTIONS.....	6
2.4.1. <i>AddVolume</i>	6
2.4.2. <i>Set Actions</i>	7
2.4.3. <i>GetAudio</i>	8
2.4.4. <i>(No AddTreble, AddBass, AddBalance, AddFade)</i>	8
2.4.5. <i>Non-Standard Actions Implemented by an UPnP Vendor</i>	8
2.4.6. <i>Relationships Between Actions</i>	8
2.4.7. <i>Common Error Codes</i>	8
2.5. THEORY OF OPERATION.....	9
3. XML SERVICE DESCRIPTION.....	10
4. TEST.....	13

List of Tables

Table 1: State variables	4
Table 2: Event moderation	5
Table 3: Actions	6

1. Overview and Scope

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

Audio:1 provides programmatic control for volume, tone, and spatial balance of a device with audio output.

Audio:1 enables the following functions:

- Setting the volume, treble, bass, left versus right and front versus rear balance.

Audio:1 does not enable:

- Multiple band equalization.

The audio signal is analog and is sent over whatever output ports / plugs provided by the device. If there are multiple sets of output ports / plugs on the device, switching from one set to another is done using another type of service described elsewhere.

This is only a sample. It does not replace the work of an UPnP Forum working committee. It is only intended to illustrate the use of device and service templates.

1.1. Change Log

[14 Aug 2000] Created v0.01.

[17 Aug 2000] v0.02. Moved Relationships to Other Service Types to device template.

[22 Aug 2000] v0.03. Rearranged order of argument sub elements. Eliminated duplicate retval markers.

[23 Aug 2000] v0.04. Clarified that there is no effect on state by an action if an error occurs.

[5 Feb 2001] v0.04. Fixed <retval /> element in XML Service Description.

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:Audio:1](#)

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

2.2. State Variables

Defines state variables for volume, tone (treble and bass), and spatial balance (left-to-right, front-to-rear). Specifics for these state variables are listed in the table and subsections below.

Table 1: State variables

Variable Name	Req. or Opt. ¹	Data Type	Allowed Value	Default Value	Eng. Units
Volume	R	ui1	>=0, <=255	(none)	n/a
Treble	R	i1	>=-127, <=127	0	n/a
Bass	R	i1	>=-127, <=127	0	n/a
Balance	R	i1	>=-127, <=127	0	n/a
Fade	R	i1	>=-127, <=127	0	n/a
A_ARG_TYPE_Delta	R	i2	>=-255, <=255	(none)	n/a
<i>Non-standard state variables implemented by an UPnP vendor go here.</i>	X	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>	<i>TBD</i>

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

2.2.1. Volume

Indicates the relative volume of audio playback. The minimum value (0) is silent. The maximum value (255) is maximum volume. There is no default value. An implementation should retain the value of the Volume state variable and re-use it when the device is powered up.

2.2.2. Treble

Indicates the emphasis of treble tone control relative to mid-range audio. The middle value (0) is treble at the same volume as mid-range audio; values less than this suppress treble; values greater, emphasize treble. The default value (0) indicates treble control should be neutral when the device is powered up.

2.2.3. Bass

Indicates the emphasis of bass tone control relative to mid-range audio. The middle value (0) is bass at the same volume as mid-range audio; values less than this suppress bass; values greater, emphasize bass. The default value (0) indicates bass control should be neutral when the device is powered up.

2.2.4. Balance

Indicates bias in left versus right spatial balance. The middle value (0) is evenly balanced between left and right; values less than this are louder on the left than right; values greater, louder on the right than left. The default value (0) indicates left and right should be balanced evenly when the device is powered up.

2.2.5. Fade

Indicates bias in front versus back spatial balance. The middle value (0) is evenly balanced between front and rear; values less than this are louder in the rear than front; values greater, louder in the front than rear. The default value (0) indicates front and rear should be balanced evenly when the device is powered up.

2.2.6. A_ARG_TYPE_Delta

Defined only to provide typing for one or more action arguments defined below. (Does not model any state of the service. No get action defined to query for this variable's value.)

2.2.7. Non-Standard State Variables Implemented by an UPnP Vendor

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

2.2.8. Relationships Between State Variables

The values of Volume, Treble, Bass, Balance, and Fade are independent of one another. The values of A_TYPE_ variables are independent of all other state variables.

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

2.3. Eventing and Moderation

As the table below summarizes, Audio:1 defines moderated eventing for one of its standard state variables and no eventing for the rest.

Table 2: Event moderation

Variable Name	Evented	Moderated Event	Max Event Rate ¹	Logical Combination	Min Delta per Event ²
Volume	yes	yes	5	n/a	n/a
Treble	no	no	n/a	n/a	n/a
Bass	no	no	n/a	n/a	n/a
Balance	no	no	n/a	n/a	n/a
Fade	no	no	n/a	n/a	n/a
A_ARG_TYPE_Delta	no	no	n/a	n/a	n/a
<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>

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

² (N) * (allowedValueRange Step).

2.3.1. Event Model

Some control points may wish to display user interface reflecting the audio state variables. The most common is Volume, so the Volume state variable is evented. In an effort to curb unnecessary network traffic, moderation is used to limit events to at most one per 5 seconds; any longer would likely be noticeable to a user watching the interface. Other state variables, while useful in a user interface, would likely only be displayed while the user was actually changing the values. In this case, the control point would query for the value of the state variable(s) prior to displaying a control for changing them; when the user wanted to change a value, the control point would send out the appropriate action while updating its own local copy for display in the user interface.

2.4. Actions

As the table below summarizes, Audio:1 defines actions to set and get the standard state variables.

Volume is manipulated frequently, so an additional action is defined to allow incrementing (or decrementing) it without querying for its value.

Anticipating interface that will provide per-variable control, a separate set action is defined for each variable. Since manipulation of state variables (other than volume) is infrequent, making incremental changes requires first querying for the value of the appropriate state variable. A single get action is defined to retrieve the value of all state variables.

Except where noted, if an action is an error, calling the action will have no effect on state.

Immediately following the 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
AddVolume	R
SetVolume	R
SetTreble	R
SetBass	R
SetBalance	R
SetFade	R
GetAudio	R
<i>Non-standard actions implemented by an UPnP vendor go here.</i>	X

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

2.4.1. AddVolume

Increments (or decrements) the current volume level.

2.4.1.1. Arguments

Takes a value to add to the Volume state variable and returns the new value of the Volume state variable.

Argument(s)	Direction	relatedStateVariable
Delta	IN	A_ARG_TYPE_Delta

NewVolume	OUT ^R	Volume
-----------	------------------	--------

^R Return value.

2.4.1.2. Effect on State

Adds the value of the in argument to the Volume state variable. (It does not change any other state variables.) If delta is positive, Volume increases; if negative, decreases. If adding delta would make the value of the Volume state variable greater (or less) than its maximum (or minimum), Volume is set to the maximum (or minimum). (No error.) That is,

```
IF (Volume + Delta > maximum) THEN ASSIGN(Volume, maximum)
ELSE IF (Volume + Delta < minimum) THEN ASSIGN(Volume, minimum)
ELSE ASSIGN(Volume, Volume + Delta)
```

2.4.1.3. Errors

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

2.4.2. Set Actions

To simplify this template, this subsection describes five set actions in this single section: SetVolume, SetTreble, SetBass, SetBalance, SetFade. These actions behave similarly; when they differ, an enumerated list of the behaviors is listed separately followed by the term "respectively". For instance, these actions set the current volume, treble, bass, balance, and fade level, respectively.

2.4.2.1. Arguments

Each of the set actions takes a single in argument and sets the Volume, Treble, Bass, Balance, and Fade state variable, respectively.

Argument(s)	Direction	relatedStateVariable
NewVolume, NewTreble, NewBass, NewBalance, or NewFade, respectively	IN	Volume, Treble, Bass, Balance, or Fade, respectively

2.4.2.2. Effect on State

Sets the Volume, Treble, Bass, Balance, or Fade state variable to the value of the in argument. (It does not change any other state variables.) If in argument is larger than the maximum, sets Volume, Treble, Bass, Balance, or Fade to the maximum. If in argument is less than the minimum, sets Volume, Treble, Bass, Balance, or Fade to the minimum. (No error.) That is,

```
IF (in argument > maximum) THEN
  ASSIGN(Volume, Treble, Bass, Balance, or Fade, respectively, maximum)
ELSE IF (in argument < minimum) THEN
  ASSIGN(Volume, Treble, Bass, Balance, or Fade, respectively, minimum)
ELSE ASSIGN(Volume, Treble, Bass, Balance, or Fade, respectively, in argument)
```

2.4.2.3. Errors

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

501	Action Failed	See UPnP Device Architecture section on Control.
800-899	TBD	(Specified by UPnP vendor.)

2.4.3. GetAudio

Queries for the current volume, treble, bass, balance, and fade level.

2.4.3.1. Arguments

Returns the value of the Volume, Treble, Bass, Balance, and Fade state variables.

Argument(s)	Direction ¹	relatedStateVariable
CurrentVolume	OUT	Volume
CurrentTreble	OUT	Treble
CurrentBass	OUT	Bass
CurrentBalance	OUT	Balance
CurrentFade	OUT	Fade

¹ (None of the OUT arguments are marked as a return value.)

2.4.3.2. Effect on State

(None.)

2.4.3.3. Errors

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

2.4.4. (No AddTreble, AddBass, AddBalance, AddFade)

Must query for the value of the appropriate state variable using a get action and then set using a set action.

2.4.5. 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).

2.4.6. 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.

2.4.7. 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.

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

2.5. Theory of Operation

To allow a user to raise (or lower) the volume, a control point sends an action to add a positive (or negative) delta to the volume.

```
// Invoke AddVolume with 10
//
```

To mute volume, a control point sets the volume to 0.

```
// Invoke SetVolume with 0
//
```

To allow a user to set the treble, a control point queries for the value of the audio state variables and displays a user interface with an appropriate control. When the user wanted to change the treble, the control point would send out an action to set the treble and update its own local copy for display in the user interface. Setting bass, balance, and fading is similar.

```
// Query for the value of the audio state variables
//   Invoke GetAudio
// Initialize user interface with current values
//   Use CurrentTreble out argument
// When user changes treble send action
//   Invoke SetTreble
//
```

3. XML Service Description

```

<?xml version="1.0"?>
<scpd xmlns="urn:schemas-upnp-org:service-1-0">
  <specVersion> <!-- UPnP version 1.0 -->
    <major>1</major>
    <minor>0</minor>
  </specVersion>
  <actionList>
    <action> <!-- add delta to volume -->
      <name>AddVolume</name>
      <argumentList>
        <argument>
          <name>Delta</name>
          <relatedStateVariable>A ARG TYPE Delta</relatedStateVariable>
          <direction>in</direction>
        </argument>
        <argument>
          <name>NewVolume</name>
          <relatedStateVariable>Volume</relatedStateVariable>
          <direction>out</direction>
        </argument>
      </argumentList>
    </action>
    <action> <!-- absolute set -->
      <name>SetVolume</name>
      <argumentList>
        <argument>
          <name>NewVolume</name>
          <relatedStateVariable>Volume</relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetTreble</name>
      <argumentList>
        <argument>
          <name>NewTreble</name>
          <relatedStateVariable>Treble</relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetBass</name>
      <argumentList>
        <argument>
          <name>NewBass</name>
          <relatedStateVariable>Bass</relatedStateVariable>
          <direction>in</direction>
        </argument>
      </argumentList>
    </action>
    <action>
      <name>SetBalance</name>

```

```

    <argumentList>
      <argument>
        <name>NewBalance</name>
        <relatedStateVariable>Balance</relatedStateVariable>
        <direction>in</direction>
      </argument>
    </argumentList>
  </action>
  <action>
    <name>SetFade</name>
    <argumentList>
      <argument>
        <name>NewFade</name>
        <relatedStateVariable>Fade</relatedStateVariable>
        <direction>in</direction>
      </argument>
    </argumentList>
  </action>
  <action> <!-- query for state variables -->
    <name>GetAudio</name>
    <argumentList>
      <argument>
        <name>CurrentVolume</name>
        <relatedStateVariable>Volume</relatedStateVariable>
        <direction>out</direction>
      </argument>
      <argument>
        <name>CurrentTreble</name>
        <relatedStateVariable>Treble</relatedStateVariable>
        <direction>out</direction>
      </argument>
      <argument>
        <name>CurrentBass</name>
        <relatedStateVariable>Bass</relatedStateVariable>
        <direction>out</direction>
      </argument>
      <argument>
        <name>CurrentBalance</name>
        <relatedStateVariable>Balance</relatedStateVariable>
        <direction>out</direction>
      </argument>
      <argument>
        <name>CurrentFade</name>
        <relatedStateVariable>Fade</relatedStateVariable>
        <direction>out</direction>
      </argument>
    </argumentList>
  </action>
  <!-- Declarations for other actions implemented by an -->
  <!-- UPNP vendor (if any) go here. -->
</actionList>
<serviceStateTable>
  <stateVariable sendEvents="yes">
    <name>Volume</name>
    <dataType>ui1</dataType>
    <allowedValueRange>
      <minimum>0</minimum>
      <maximum>255</maximum>
    </allowedValueRange>
  </stateVariable>
</serviceStateTable>

```

```

    </allowedValueRange>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>Treble</name>
    <dataType>i1</dataType>
    <allowedValueRange>
      <minimum>-127</minimum>
      <maximum>127</maximum>
    </allowedValueRange>
    <defaultValue>0</defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>Bass</name>
    <dataType>i1</dataType>
    <allowedValueRange>
      <minimum>-127</minimum>
      <maximum>127</maximum>
    </allowedValueRange>
    <defaultValue>0</defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>Balance</name>
    <dataType>i1</dataType>
    <allowedValueRange>
      <minimum>-127</minimum>
      <maximum>127</maximum>
    </allowedValueRange>
    <defaultValue>0</defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>Fade</name>
    <dataType>i1</dataType>
    <allowedValueRange>
      <minimum>-127</minimum>
      <maximum>127</maximum>
    </allowedValueRange>
    <defaultValue>0</defaultValue>
  </stateVariable>
  <stateVariable sendEvents="no">
    <name>A_ARG_TYPE_Delta</name>
    <dataType>i2</dataType>
    <allowedValueRange>
      <minimum>-255</minimum>
      <maximum>255</maximum>
    </allowedValueRange>
  </stateVariable>
  <!-- Declarations for other state variables implemented by an -->
  <!-- UPNP vendor (if any) go here. -->
</serviceStateTable>
</scpd>

```

4. Test

TBD.