



OPENHOME

SEARCH

- Overview
- ohMedia
- ohNet
- Forums

**From OpenHome**

## Contents

- 1 Overview
- 2 State Variables
  - 2.1 TransportState
  - 2.2 Repeat
  - 2.3 Shuffle
  - 2.4 Id
  - 2.5 IdArray
  - 2.6 TracksMax
  - 2.7 ProtocolInfo
- 3 Actions
  - 3.1 Play
  - 3.2 Pause
  - 3.3 Stop
  - 3.4 Next
  - 3.5 Previous
  - 3.6 SetRepeat
  - 3.7 Repeat
  - 3.8 SetShuffle
  - 3.9 Shuffle
  - 3.10 SeekSecondAbsolute
  - 3.11 SeekSecondRelative
  - 3.12 SeekId
  - 3.13 SeekIndex
  - 3.14 TransportState
  - 3.15 Id
  - 3.16 Read
  - 3.17 ReadList
  - 3.18 Insert
  - 3.19 DeleteId
  - 3.20 DeleteAll
  - 3.21 TracksMax
  - 3.22 IdArray
  - 3.23 IdArrayChanged
  - 3.24 ProtocolInfo
- 4 Technical Details
  - 4.1 Service Description (XML)

## Overview

The Playlist service provides access to a list of tracks, which can be played in sequence, starting at any point. ‘Shuffle’ and ‘Repeat’ modes also exist, allowing track order to be randomised and playback to loop. If a device's Product service reports a source of type 'Playlist', then that device is guaranteed to bear the Playlist service.

# State Variables

## TransportState

One of Stopped, Playing, Paused or Buffering.

## Repeat

By default a playlist plays each track in sequence then stops after playing the final track. Setting the Repeat property changes this behaviour to loop from the last track back to the first, playing it immediately. The Next and Previous actions also vary their behaviour according to this property at the ends of a playlist.

By default, Repeat is off.

## Shuffle

Tracks are played in random order if the Shuffle property is enabled. As with normal playback, each track is played once then the playlist stops. If Shuffle is enabled mid-way through a playlist, each track in the list is shuffled, including any that have just been played.

When combined with Repeat, each track is played once before any track is played for a second time.

By default, Shuffle is off.

## Id

The id of the current track (the track currently playing or that would be played if the Play action was invoked). Or 0 if the playlist is empty.

## IdArray

An IdArray containing the ids of the tracks currently in the playlist. Contains [0.. TracksMax] elements.

## TracksMax

The maximum number of tracks a playlist can contain. Set on startup and invariant thereafter.

## ProtocolInfo

Audio formats and networking protocols supported by the device. Will be the same as reported by Radio and AVTransport if these services are present. Set on startup and invariant thereafter.

# Actions

## Play

If TransportState is Stopped or Paused, start playing the track indicated by the Id state variable.

If TransportState is Playing or Buffering, re-start playing the current track from the beginning.

If the Playlist source is not active, deactivates the current source and activates Playlist.

Disables Standby in the Product service if that was enabled (and if the product is controllable over the network in this mode).

## Pause

Pause the current track. The TransportState will change to Paused, possibly after a short delay allowing the volume to be ramped down.

This will be converted to Stop if the track cannot be paused e.g. for internet radio.

## Stop

Stop the current track and move the current position to the start of this track. The TransportState will change to Stopped, possibly after a short delay allowing the volume to be ramped down.

## Next

Halt playback of the current track and start playing the next item in the playlist.

If the last track in a playlist is being played and Repeat is off, the playlist will pause. The first track in the playlist will be played from the beginning if the Play action is invoked.

If the last track in a playlist is being played and Repeat is on, the first last track in the playlist is played.

## Previous

Halt playback of the current track and start playing the previous item in the playlist.

If the first track in a playlist is being played and Repeat is off, the playlist will pause. The first track will be played from the beginning if the Play action is invoked.

If the first track in a playlist is being played and Repeat is on, the last track in the playlist is played.

Behaviour when Shuffle is enabled is implementation defined.

## SetRepeat

Enable or disable repeat mode.

## Repeat

Return the value of the Repeat state variable.

## SetShuffle

Enable or disable shuffle mode.

## Shuffle

Return the value of the Shuffle state variable.

## SeekSecondAbsolute

Seek to an absolute second within the current track. This is only available if the current track has a finite length.

## SeekSecondRelative

Seek to a relative second within the current track. This is only available if the current track has a finite length.

## SeekId

Halt playback of the current track and switch to the track with the specified id.  
Should return a 800 fault code if the id is invalid.

## SeekIndex

Halt playback of the current track and switch to the track with the specified index.  
Use of this function is discouraged as it results in an unpredictable track being played in the presence of multiple control points. Try to use SeekId in preference to this.  
May return a 800 fault code if the index is beyond the end of the playlist.

## TransportState

Return the value of the TransportState state variable.

## Id

Return the value of the Id state variable.

## Read

Report the uri and metadata for a given track id.  
Returns a 800 fault code if the given id is not in the playlist.

## ReadList

Given a space separated list of track Id's, report their associated uri and metadata in the following xml form:

```
<TrackList>
  <Entry>
    <Id></Id>
    <Uri></Uri>
    <Metadata></Metadata>
  </Entry>
</TrackList>
```

Any ids not in the playlist are ignored.

## Insert

Adds the given uri and metadata as a new track to the playlist.  
Set the AfterId argument to 0 to insert a track at the start of the playlist.  
Reports a 800 fault code if AfterId is not 0 and doesn't appear in the playlist.  
Reports a 801 fault code if the playlist is full (i.e. already contains TracksMax tracks).

## DeleteId

Delete the track with a given id from the playlist.  
If the id to be deleted is currently playing, the next track becomes current.  
Returns a 800 fault code if the given id is not in the playlist.

# DeleteAll

Delete all tracks from the playlist.

# TracksMax

Return the value of the TracksMax state variable.

# IdArray

Return the value of the IdArray state variable. Also returns a token which can be used with the IdArrayChanged action.

# IdArrayChanged

Check to see if the id array has changed since gathering the specified Token. This Token must have been previously collected from the IdArray action.

This mechanism is provided specifically for clients unable to partake in UPnP eventing.

# ProtocolInfo

Return the value of the ProtocolInfo state variable.

# Technical Details

Domain	: av.openhome.org
Name	: Playlist
Version	: 1

# Service Description (XML)

```
<?xml version="1.0" encoding="utf-8"?>

<scpd xmlns="urn:schemas-upnp-org:service-1-0">

  <specVersion>
    <major>1</major>
    <minor>0</minor>
  </specVersion>

  <actionList>
    <action>
      <name>Play</name>
    </action>
    <action>
      <name>Pause</name>
    </action>
    <action>
      <name>Stop</name>
    </action>
    <action>
      <name>Next</name>
    </action>
    <action>
      <name>Previous</name>
    </action>
    <action>
      <name>SetRepeat</name>
      <argumentList>
        <argument>
```

```

        <name>Value</name>
        <direction>in</direction>
        <relatedStateVariable>Repeat</relatedStateVariable>
    </argument>
</argumentList>
</action>
<action>
    <name>Repeat</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>out</direction>
            <relatedStateVariable>Repeat</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>SetShuffle</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>in</direction>
            <relatedStateVariable>Shuffle</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>Shuffle</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>out</direction>
            <relatedStateVariable>Shuffle</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>SeekSecondAbsolute</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>in</direction>
            <relatedStateVariable>Absolute</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>SeekSecondRelative</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>in</direction>
            <relatedStateVariable>Relative</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>SeekId</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>in</direction>
            <relatedStateVariable>Id</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>SeekIndex</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>in</direction>
            <relatedStateVariable>Index</relatedStateVariable>

```

```

    </argument>
  </argumentList>
</action>
<action>
  <name>TransportState</name>
  <argumentList>
    <argument>
      <name>Value</name>
      <direction>out</direction>
      <relatedStateVariable>TransportState</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>Id</name>
  <argumentList>
    <argument>
      <name>Value</name>
      <direction>out</direction>
      <relatedStateVariable>Id</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>Read</name>
  <argumentList>
    <argument>
      <name>Id</name>
      <direction>in</direction>
      <relatedStateVariable>Id</relatedStateVariable>
    </argument>
    <argument>
      <name>Uri</name>
      <direction>out</direction>
      <relatedStateVariable>Uri</relatedStateVariable>
    </argument>
    <argument>
      <name>Metadata</name>
      <direction>out</direction>
      <relatedStateVariable>Metadata</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>ReadList</name>
  <argumentList>
    <argument>
      <name>IdList</name>
      <direction>in</direction>
      <relatedStateVariable>IdList</relatedStateVariable>
    </argument>
    <argument>
      <name>TrackList</name>
      <direction>out</direction>
      <relatedStateVariable>TrackList</relatedStateVariable>
    </argument>
  </argumentList>
</action>
<action>
  <name>Insert</name>
  <argumentList>
    <argument>
      <name>AfterId</name>
      <direction>in</direction>
      <relatedStateVariable>Id</relatedStateVariable>
    </argument>
    <argument>
      <name>Uri</name>
      <direction>in</direction>
      <relatedStateVariable>Uri</relatedStateVariable>
    </argument>
    <argument>
      <name>Metadata</name>

```

```

        <direction>in</direction>
        <relatedStateVariable>Metadata</relatedStateVariable>
    </argument>
    <argument>
        <name>NewId</name>
        <direction>out</direction>
        <relatedStateVariable>Id</relatedStateVariable>
    </argument>
</argumentList>
</action>
<action>
    <name>DeleteId</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>in</direction>
            <relatedStateVariable>Id</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>DeleteAll</name>
</action>
<action>
    <name>TracksMax</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>out</direction>
            <relatedStateVariable>TracksMax</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>IdArray</name>
    <argumentList>
        <argument>
            <name>Token</name>
            <direction>out</direction>
            <relatedStateVariable>IdArrayToken</relatedStateVariable>
        </argument>
        <argument>
            <name>Array</name>
            <direction>out</direction>
            <relatedStateVariable>IdArray</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>IdArrayChanged</name>
    <argumentList>
        <argument>
            <name>Token</name>
            <direction>in</direction>
            <relatedStateVariable>IdArrayToken</relatedStateVariable>
        </argument>
        <argument>
            <name>Value</name>
            <direction>out</direction>
            <relatedStateVariable>IdArrayChanged</relatedStateVariable>
        </argument>
    </argumentList>
</action>
<action>
    <name>ProtocolInfo</name>
    <argumentList>
        <argument>
            <name>Value</name>
            <direction>out</direction>
            <relatedStateVariable>ProtocolInfo</relatedStateVariable>
        </argument>
    </argumentList>
</action>

```



```
</actionList>
```

```
<serviceStateTable>
```

```
  <stateVariable sendEvents="yes">
```

```
    <name>TransportState</name>
```

```
    <dataType>string</dataType>
```

```
    <allowedValueList>
```

```
      <allowedValue>Playing</allowedValue>
```

```
      <allowedValue>Paused</allowedValue>
```

```
      <allowedValue>Stopped</allowedValue>
```

```
      <allowedValue>Buffering</allowedValue>
```

```
    </allowedValueList>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="yes">
```

```
    <name>Repeat</name>
```

```
    <dataType>boolean</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="yes">
```

```
    <name>Shuffle</name>
```

```
    <dataType>boolean</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="yes">
```

```
    <name>Id</name>
```

```
    <dataType>ui4</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="yes">
```

```
    <name>IdArray</name>
```

```
    <dataType>bin.base64</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="yes">
```

```
    <name>TracksMax</name>
```

```
    <dataType>ui4</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="yes">
```

```
    <name>ProtocolInfo</name>
```

```
    <dataType>string</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>Index</name>
```

```
    <dataType>ui4</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>Relative</name>
```

```
    <dataType>i4</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>Absolute</name>
```

```
    <dataType>ui4</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>IdList</name>
```

```
    <dataType>string</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>TrackList</name>
```

```
    <dataType>string</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>Uri</name>
```

```
    <dataType>string</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>Metadata</name>
```

```
    <dataType>string</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>IdArrayToken</name>
```

```
    <dataType>ui4</dataType>
```

```
  </stateVariable>
```

```
  <stateVariable sendEvents="no">
```

```
    <name>IdArrayChanged</name>
```

```
    <dataType>boolean</dataType>
```

```
  </stateVariable>
```

</serviceStateTable>

</scpd>

Retrieved from "http://wiki.openhome.org/wiki/Av:Developer:PlaylistService"

---

- Login