

WiiM UPnP API

Version 1.0

Device discovery

WiiM devices are designed for effortless auto-discovery within your local network using mDNS, identifiable by the service type '_linkplay._tcp.'. Additionally, they are compatible with the standard UPnP M-SEARCH method for device discovery. Once a device is discovered, communication between the controller app and LinkPlay devices is facilitated through the publish/subscribe (SUB/PUB) model as defined in the UPnP standard.

UPnP device type and service

The proprietary UPnP device type for WiiM is identified as **urn:schemas-upnp-org:device:LinkplayRenderer:1**.

This includes a range of proprietary services. Device manager is reserved for WiiM internal use and Qplay is to support Tencent QQ Music.

Service	Description
urn:schemas-linkplay-com:service:DeviceManager;1	DeviceManager
urn:schemas-linkplay-com:service:PlayQueue:1	PlayQueue
urn:schmas-tencent-com:server:QPlay:1	QPlay

It also functions as a standard media render that has the following device type and service list.

Device Type : **urn:schemas-upnp-org:device:MediaRenderer:1**

Service List

Service	Description
urn:schemas-upnp-org:service:RenderingControl:1	RenderingControl
urn:schemas-upnp-org:service:ConnectionManager:1	ConnectionManager

urn:schemas-upnp- org:service:AVTransport:1	AVTransport
--	-------------

The standard UPnP AVTransport interface is supported.

AVTransport Service

The following interfaces defined in the UPnP standards are supported. Please refer to UPnP AVTransport service spec. for details.

Actions	Description	Parameters
GetCurrentTransportActions	Returns CurrentTransportActions state variables	Return action state variable: "Play,Stop,Seek,X_DLNA_SeekTime,Pause"
GetDeviceCapabilities	Returns device capability of playback	Please refer to PlaybackStorageMedium for more information on PlayMedia. Support extension on PlaybackStorageMedium.
GetMediaInfo	Returns info. associated with the current media	The return is as follows: NrTracks MediaDuration CurrentURI CurrentURIMetaData NextURI NextURIMetaData PlayMedium
SetAVTransportURI	Set URL for playback	CurrentURI

		CurrentURIMetaData
GetTransportInfo	Returns current transport state	CurrentTransportState CurrentTransportStatus CurrentSpeed
GetTransportSettings	Returns current settings such as the current play mode	Returns PlayMode : "NORMAL,SHUFFLE, REPEAT_ONE, REPEAT_ALL, RANDOM, DIRECT_1"
GetPositionInfo	Returns playback position information	Returns the current position of the transport: Track TrackDuration /TrackMetaData TrackURI RelTime / AbsTime RelCount /AbsCount
Play	Play	
Stop	Stop	
Pause	Pause	
Seek	Seek to the relative time or the specified track number	Target: REL_TIME TRACK_NR: seek to the specific number of track
Next	Play next song	
Previous	Play previous song	
SetPlayMode	Set new play mode	NewPlayMode

AVTransport service state table

State variable	Description
TransportState	The transport state is as follows: {"STOPPED","PAUSED_PLAYBACK","PLAYING","TRANSITIONING","NO_MEDIA_PRESENT"}
PlaybackStorageMedium	The following storage medium is defined: "NONE, UNKNOWN, STATION-NETWORK, SONGLIST-NETWORK, SONGLIST-LOCAL, SONGLIST-LOCAL_TF, THIRD-DLNA, AIRPLAY, RADIO-NETWORK, QPLAY, ALEXA_PANDORA, ALEXA_AUDIBLE, ALEXA, SPOTIFY, LINE-IN, OPTICAL, EXTERNAL_USB, BLUETOOTH, RCA, FM, SECORD_LINEIN, XLR, HDMI, MULTIROOM-SLAVE"
CurrentPlayMode	The play mode is defined: "NORMAL,SHUFFLE, REPEAT_ONE, REPEAT_ALL, RANDOM, DIRECT_1"
NumberOfTracks	1~65535
CurrentTrack	1~65535
CurrentTrackDuration	Current track duration
CurrentTrackMetaData	Current track metadata
CurrentTrackURI	Current track URI

NextAVTransportURI	
NextAVTransportURIMet aData	
RelativeTimePosition	Position of current playback session
CurrentTransportActions	

RenderingControl Service

Following the standard actions in the UPnP media render's RenderingControl

Action	Description	Variable
GetMute	Return the mute status	CurrentMute
SetMute	Set the mute status	DesiredMute
GetVolume	Return the current volume	CurrentVolume
SetVolume	Set the current volume	DesiredVolume
GetChannel	Return the current channel	CurrentChannel
SetChannel	Set the current channel	DesiredChannel
GetEqualizer	Return the current EQ	CurrentEqualizer
SetEqualizer	Set the current EQ	DesiredEqualizer

ServiceStateTable in RendingControl

Variable	Detail
Mute	1,0
Volume	0~100
Equalizer	0~4 EQ could be self-defined
Channel	0~2 corresponding to LF,RF,stereo

PlayQueue Service

PlayQueue service is WiiM proprietary that includes

- Basic play queue API
- Music service login
- Music action such as setting, thumb-up, thumb-down etc.
- Playlist preset actions such as BackUpQueue, SetKeyMapping, SetSpotifyPreset etc.
- Favorite on songs or playlist
- Music alarm

Actions	Description	Parameter
CreateQueue	Create the play queue	QueueContext The playlist in XML format (see details below)
DeleteQueue	Delete the play queue	QueueName
BrowseQueue	Obtain the current play queue	QueueName
PlayQueueWithIndex	Play the playlist, starting with the music with Index	QueueName Index
BackUpQueue	Copy the queue context to the buffer specified with QueueContext	QueueContext
RemoveTracksInQueue	Remove the tracks from the playlist of QueueName	QueueName RangStart RangEnd
AppendTracksInQueue	Insert the tracks to current play queue	QueueContext Direction

SetKeyMapping	Set the preset key for the playlist specified by QueueContext	QueueContext
GetKeyMapping	Get the preset key	
UserLogin	User login with username and password	AccountSource UserName PassWord
GetUserInfo	Get user account information	AccountSource
UserLogout	Logout user on the device	AccountSource
GetUserAccountHistory	Get the account information from the device	AccountSource
SetUserFavorites	Set the content as the favorite on speaker	AccountSource Action MediaType MediaID
GetUserFavorites	Get the user's favorite contents from speaker	AccountSource MediaType Filter
SetSpotifyPreset	Set Spotify preset	KeyIndex
SetAlarmQueue	Add playlist to alarm as alarm source	AlarmContext
GetAlarmQueue	Get the alarm source	AlarmName
DeleteAlarmQueue	Delete the playlist information from the alarm named AlarmName	AlarmName

PlayQueue ServiceStateTable

Variable	Detail
CurrentPlayListName	The name of current playlist
CurrentIndex	The index of current playing track
TotalPage	The number of pages of the playlist
CurrentPage	The current page number of playlist is being played
MusicSource	It includes Amazon Music, Tidal, iHeartradio, Pandora, Napster, Aldilife, Qobuz, Deezer, Audible, Tunein, VTuner, Radio.com etc.
UserInfo	User information from one specific music service

The QueueContext syntax example:

```
<PlayList>
  <ListName>list1 </ListName>
  <ListInfo>
    <SourceName>TTPod</SourceName>
    <TrackNumber>2</TrackNumber>
    <Radio>0</Radio> // If it's for radio station (such as mms://xxx or https://xxx), put this value as
1
  </ListInfo>
  <Tracks>
    <Track1>
      <URL>http://a.ali.dongting.com/mp3\_128\_3/53/6c/537ac0584fbf20e27cdf1effa2b87c6c.mp3?k=353fe85562a6d993&t=1421057228</URL>
      <Source>slegetank (3)_ttpod_search</Source>
      <Metadata><?xml version="1.0" encoding="UTF-8"?><DIDL-Lite
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:upnp="urn:schemas-upnp-org:metadata-1-0/upnp/" xmlns:song="www.wiimu.com/song/" xmlns:custom="www.wiimu.com/custom/"
```

```
xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-
Lite/"> <upnp:class>object.item.audioItem.musicTrack</upnp:class> <item> <dc:title>Track
Title</dc:title> <upnp:artist>Artist Name</upnp:artist> <upnp:album>Album Name
</upnp:album> <custom:url>http://a.ali.dongting.com/mp3\_128\_3/53/6c/537ac0584fbf20e27cdf
1effa2b87c6c.mp3?k=353fe85562a6d993&t=1421057228</custom:url> </item> </DIDL-
Lite> </Metadata>
```

```
</Track1>
```

```
<Track2>
```

```
<URL>http://nmo.ouj.yymommy.com/mp3\_128\_14/0c/17/0cdf45a4ddfcc31b2f1912fb09489b17.
mp3?k=c28317404ae9f97a&t=1421057228</URL>
```

```
<Source>slegetank (3)_ttpod_search</Source>
```

```
<Metadata><?xml version="1.0" encoding="UTF-8"?> <DIDL-Lite
xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:upnp="urn:schemas-upnp-org:metadata-1-
0/upnp/" xmlns:song="www.wiimu.com/song/" xmlns:custom="www.wiimu.com/custom/"
xmlns="urn:schemas-upnp-org:metadata-1-0/DIDL-
Lite/"> <upnp:class>object.item.audioItem.musicTrack</upnp:class> <item> <dc:title>Track
Title</dc:title> <upnp:artist>Artist Name</upnp:artist> <upnp:album>Album Name
</upnp:album> <custom:url>http://nmo.ouj.yymommy.com/mp3\_128\_14/0c/17/0cdf45a4ddfcc3
1b2f1912fb09489b17.mp3?k=c28317404ae9f97a&t=1421057228</custom:url> </item> </D
IDL-Lite> </Metadata>
```

```
</Track2>
```

```
</Tracks>
```

```
</PlayList>
```

Typical sequence flow to play music service

1. GetUserInfo – Get the user login information on the device
2. UserLogin – Login the service on the device
3. CreateQueue – Create the play queue
4. PlayQueueWithIndex – Start the playback with the specific track
5. UserLogout – User could logout the service on the device