LOXONE

Date: 2025-08-31
Author: LOXONE

Dynamic Audio Grouping WebAPI Documentation

Introduction

This document outlines the audio zoning for a modular setup, controlled by function block logic. We will examine a real use case in detail using the example of setup for a ballroom.

Purpose of Dynamic Grouping

The purpose of dynamic grouping is to ensure that an audio system can automatically adapt to changing room configurations and usage scenarios. Spaces in modern buildings are often reconfigurable through movable walls or flexible layouts, and the audio system must mirror these changes without requiring manual intervention.

Dynamic grouping enables:

- **Consistency:** Audio playback always aligns with the current physical room setup.
- **Flexibility:** Zones can operate independently or in combined groups, depending on the needs of the event.
- **Efficiency:** Reconfiguration is seamless and requires minimal effort from operators.
- **User Experience:** Listeners enjoy a coherent and uninterrupted sound environment that matches the space.

In essence, dynamic grouping links physical space configuration with audio distribution logic, providing a scalable solution for any environment where flexibility is required.

Miniserver Requirements

To enable dynamic grouping of audio zones, the Miniserver and Audioserver must be configured with the following requirements:

Miniserver Configuration

- The Miniserver requires a Virtual Output with the configured receiver address (hostname or IP address) and port 7091 of the respective Audioserver.
  Example: http://as0a01:7091/ (do not forget the trailing “/”).

- If the Miniserver Compact internal audio service is used, the own hostname or local address 127.0.0.1 must be specified.
  Example: http://ms368d:7091/ (do not forget the trailing “/”).

API Command Setup

Under the Virtual Output, the respective API commands are created to manage grouping logic.A template file is available for reference:
“VO_Dynamic Grouping AudioZones Ballroom Example.xml”

API Command Syntax

audio/cfg/dgroup/create/<Master>,<Player 1>[,<Player N>]

audio/cfg/dgroup/delete/<Master>

Creating and updating

audio/cfg/dgroup/create/1,2

-> creates a dynamic group of 1 and 2, Master is 1 and decides what to play

audio/cfg/dgroup/create/1,2,3

-> extends existing group with player 3

audio/cfg/dgroup/create/1,2

-> detaches player 3

audio/cfg/dgroup/create/4,1,2

-> creates an entirely new group with Master 4 and defines a new Source

(equivalent to App; join an existing group)

Deleting a group

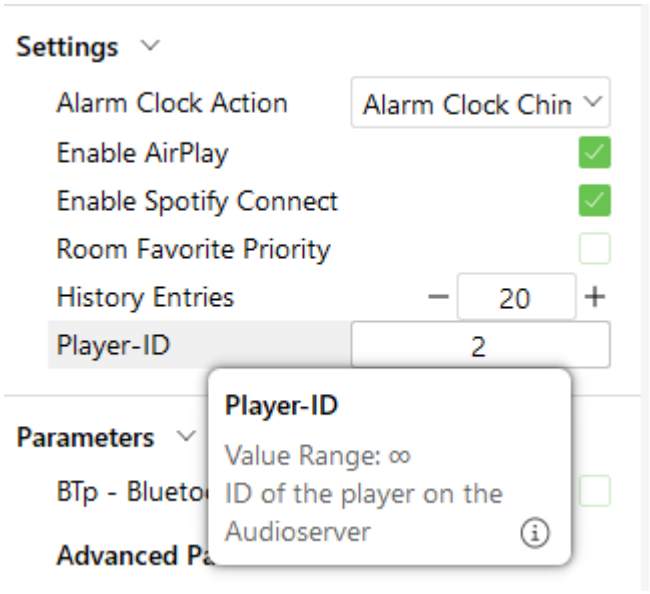
audio/cfg/dgroup/delete/4

-> this removes all players in the same dynamic group of player 4 and stops music (equivalent to App; delete entire group)

Note: while it is possible to use any player in the group for deleting the group, it is considered best practice to use the master player

Determine Player ID

This is a property of the player object



API Summary Table

Action	API Command	Description
Create Group	audio/cfg/dgroup/create/<Player1>,<Player2>,<Player3>	Creates a group with Players 1, 2, and 3. Player1 acts as group master.
Remove Player from Group	audio/cfg/dgroup/create/<Player1>,<Player3>	Updates the group by removing Player2. Player1 remains group master.
Delete Group	audio/cfg/dgroup/delete/<Player1>	Dissolves the group led by Player1.

Multiple Setup and static Groups

Multiple Audioservers

Ensure consistency of which Audioserver you send your commands to.

Always use the Audioserver that is in charge of the Master Player.

Behavior with static groups

- Static groups can never be dissolved or extended.
- A create command that has players of one or more static groups listed, then the command will be internally extended to all players of the static group
- The same rule applies for deletion.

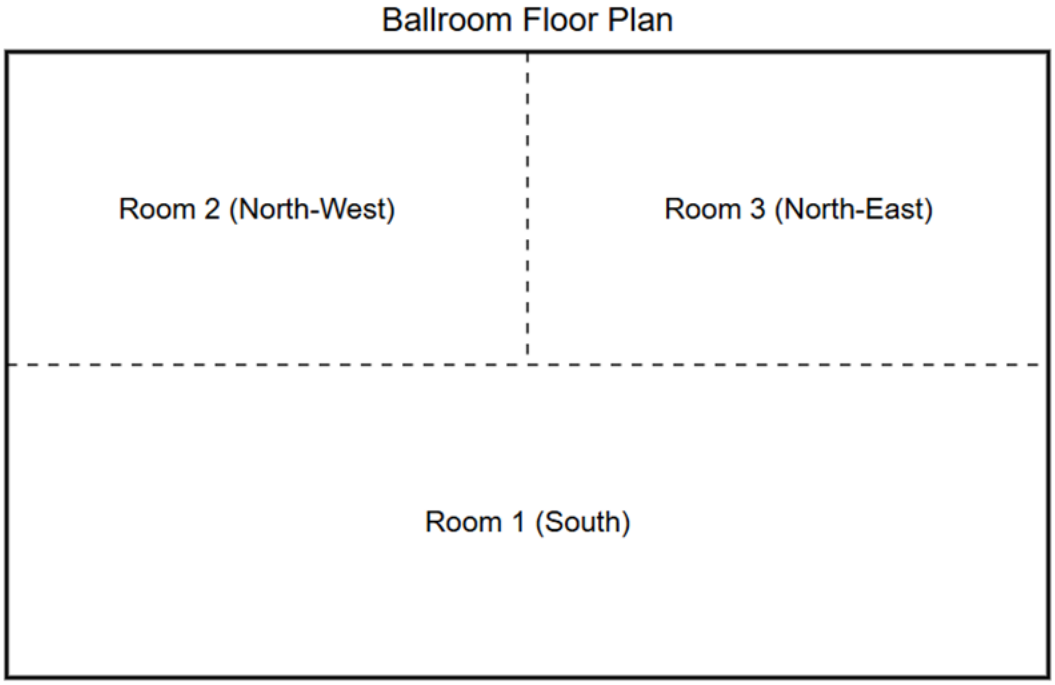
Use Case Ballroom Example

The ballroom can operate either as a single large venue or be divided into smaller rooms, each with independent or grouped audio playback. By combining audio players dynamically based on the partition state, the system ensures seamless adaptability for different event formats, ranging from a unified ballroom experience to fully independent smaller rooms.

The following sections describe the physical layout, possible partition configurations, and the corresponding audio grouping logic. A simplified floor plan illustrates the relationship between the rooms and partitions.

The ballroom can be divided by movable partitions into three zones:

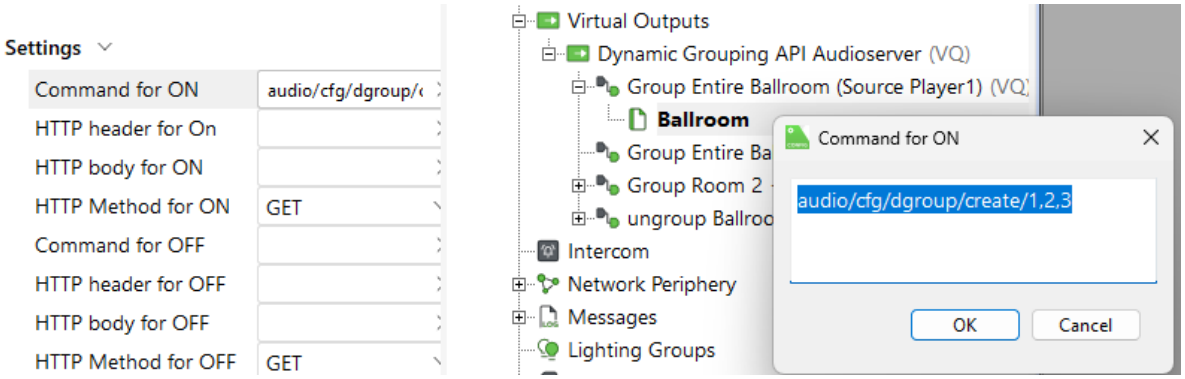
- Room 1 (South): lower half of the ballroom
- Room 2 (North-West): upper left quarter.
- Room 3 (North-East): upper right quarter.



Case 1: Playback in the entire Ballroom

- **Partitions:** None (all walls open).

- **Audio Logic:** All three players (Room 1 + Room 2 + Room 3) are grouped into one Ballroom Audio Group.
- **Result:** Synchronized playback across the entire ballroom.



Virtual Output command:
audio/cfg/dgroup/create/1,2,3

Case 2: Playback Horizontal Partition

- **Partitions:** Horizontal wall between North and South.
- **Audio Logic:**
- Room 1 (South) plays independently.
- Room 2 and Room 3 are grouped together as North Ballroom.
- **Result:** South can run separate audio while North remains unified.

Virtual Output command:
audio/cfg/dgroup/create/2,3

Case 3. Playback Horizontal + Vertical Partition

- **Partitions:** Horizontal wall + vertical wall splitting North-West and North-East.
- **Audio Logic:**
- Room 1 = independent
- Room 2 = independent
- Room 3 = independent
- **Result:** Three fully independent audio zones.

Here we delete both possible groups that can be created by combinations. This ensures that MS Logic always creates the desired assignment (without any possibly in the meantime app changes).

Virtual Output command:

audio/cfg/dgroup/delete/1;audio/cfg/dgroup/delete/2

Changelog

- 2025-08-31: Initial Description with Use Case Example