



Crestron Unified Driver Manual

©2015-2020 LENBROOK INDUSTRIES LIMITED

All rights reserved.

No part of this publication may be reproduced, stored or transmitted in any form without the written permission of Lenbrook Industries Limited. While every effort has made to ensure the contents are accurate at the time of publication, features and specifications may be subject to change without prior notice.

Revision History			
Version	Date	Author	Description
1.0.0	09/14/2020	Gary S.	First public release. Supports driver v1.0.0.
1.1.0	02/10/2021	Oliver H.	Added player grouping support. Supports driver 1.1.0.
1.5.0	20/01/2023	Oliver H.	Added programmatic control
1.5.1	13/02/2023	Oliver H.	Bug fixes to group management and volume control
1.5.2	24/07/2023	Michael F.	Bug fix – Fixed group order when player is offline

Table of Contents

1.0 Introduction	4
2.0 Installation	4
2.1 Equipment Setup.....	4
3.0 Configuration.....	5
4.0 Programmatic Control	7
5.0 Support	12
6.0 Known Issues.....	12

1.0 Introduction

The Crestron BluOS Unified Driver integration module provides rich control of BluOS streaming devices via either Crestron's Smart Graphics Media Player object or via native SIMPL programming and custom user interface. Functionality includes:

- Now playing information
- Playback controls
- Play queue management
- Source selection (including analog, optical, coaxial and HDMI inputs)
- Content browsing
- Searching
- Preset selection
- Direct input selection

Additionally, it offers support for programmatic control of player volume.

This is a multi-language driver that includes translations for all of the BluOS supported languages.

The included example program provides a simple guide to how multiple devices can be connect to a couple of user interfaces. The touch panel file included is purely for demonstrating the module and not intended to be a complete solution. However, if you modify the module parameters to suit your hardware configuration, the program will compile without errors and be fully functional.

2.0 Installation

Ensure that you have the required version of BluOS on your players. Also ensure that you are using the proper Crestron software. These versions are listed in the Release Notes.

2.1 Equipment Setup

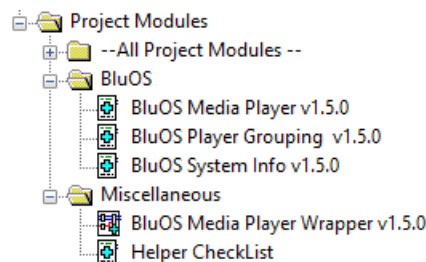
The BluOS Controller app (iOS or Android) must be used to initially configure the player. Using BluOS Controller, you will set up the player name, configure inputs and adjust audio settings. You also use BluOS Controller to login to streaming music services, add network music libraries, create presets etc.

The module supports player discovery using the Lenbrook Service Discovery Protocol (LSDP). It also supports static IP addresses.

3.0 Configuration

Using the driver will be simple to those familiar with other Media Player integration objects. The essential steps are:

1. Add the module to your SIMPL environment. This can be done by adding the modules and SIMPL# library to your SIMPL project directory, or by adding them to the global SIMPL Plus User Directory. You may need to refresh the user database before you see the driver appear in the symbol tree.
2. Add a BluOS Media Player Wrapper module to your SIMPL program for each device you wish to control. The module is designed to sit one-to-one with each controllable BluOS zone. The module can be found under the “BluOS” section of the symbol tree.



Note: If you have a multi-player configuration, such as a stereo pair, you should only add a single module and point it at the “primary” device (defined when you set up the fixed group using the BluOS app).

Note: If you have a device with multiple outputs (such as the NAD CI 580), each OUTPUT will need its own module (see “Zone” below).

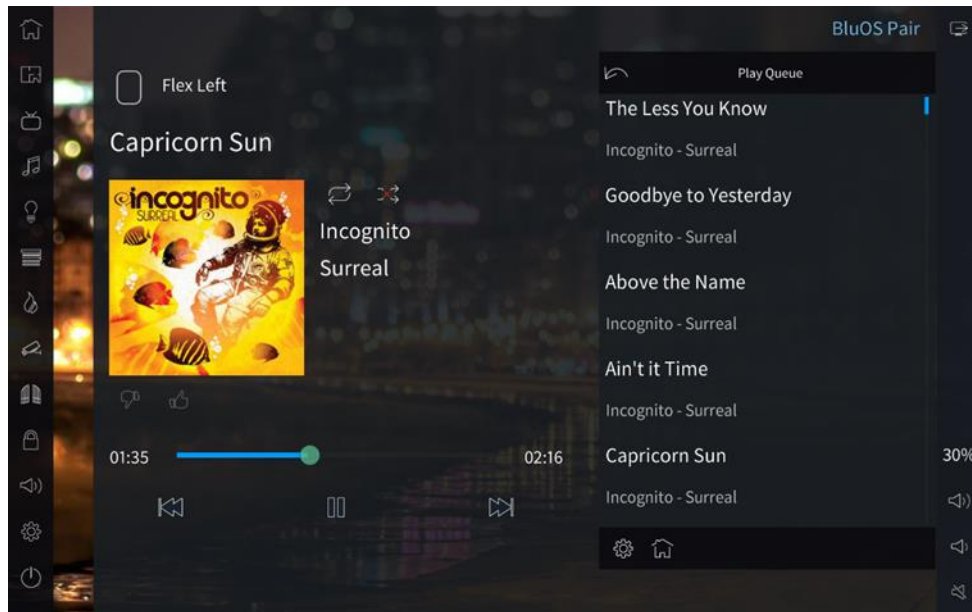
Note: Please use the “Wrapper” User Module, rather than the Media Player SIMPL+ module as this provides convenient grouping and parameter options.

3. For each module instance, you will need to configure the connection parameters as follows:

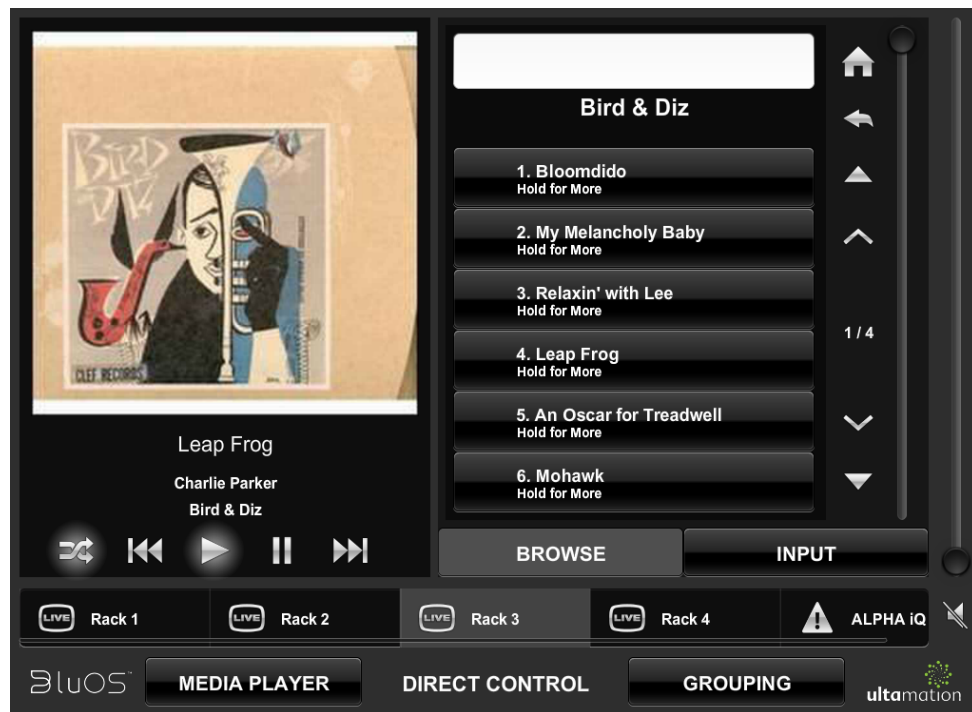
NameOrIp	192.168.1.101
Zone	Output 1
CrcpPort	48005d
Adapter	LAN Adapter
Language	English [UK]

- a. NameOrIp: This should preferably be the fixed IP address of the player. You can also use the Player's name, though this will then require the device to be discovered on the network which may add delay to the start-up process.
- b. Zone: For single zone players, this should be left at the default (Output 1). For devices with multiple outputs, such as the NAD CI 580, you should select the appropriate output to control.
- c. CrcpPort: This must be an unused port for the Media Player object to communicate with the processor module. We suggest using a unique port for each instance of the module as this will ensure the current player is referenced via the Media Server Object Router (Crestron module).
- d. Adapter: Set this to the LAN adapter on which the BluOS devices are located. This will normally be the LAN adapter (default).
- e. Language: You can define the language that the BluOS device will use to localise various elements of the browse menu. This can also be changed dynamically via the Language_Code\$ signal.

4. Each module must be connected to the Media Server Object Router which is explained in the Crestron documentation. The essential signals are Is_Offline_Fb, CRPC_To_Device\$ and CRPC_From_Device\$.
5. With a properly configured Media Server Object Router and user interface, you will now be able to browse libraries, services and presets and control the playback of the BluOS device.



Or using a suitable SIMPL program, control the BluOS device from a custom interface.



4.0 Programmatic Control

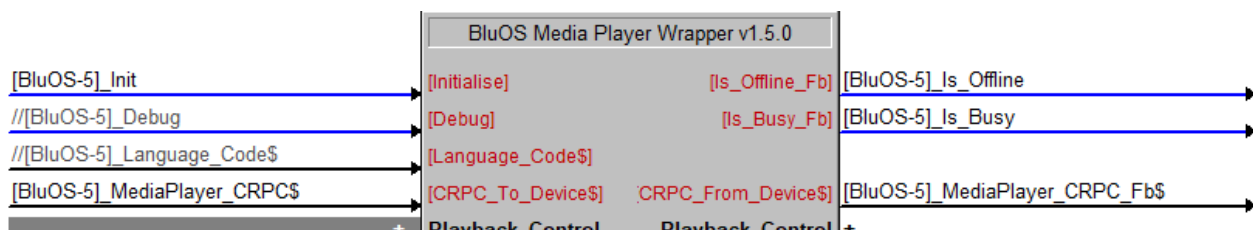
The following section describes how the BluOS system can be controlled from SIMPL.

4.1 BluOS Player Control

The module also provides for programmatic control over the BluOS device.

Note: Signals cue positions/layout may be revised for release.

Note: Settings in the SIMPL module will control the player but do not affect the Media Player visual layout.

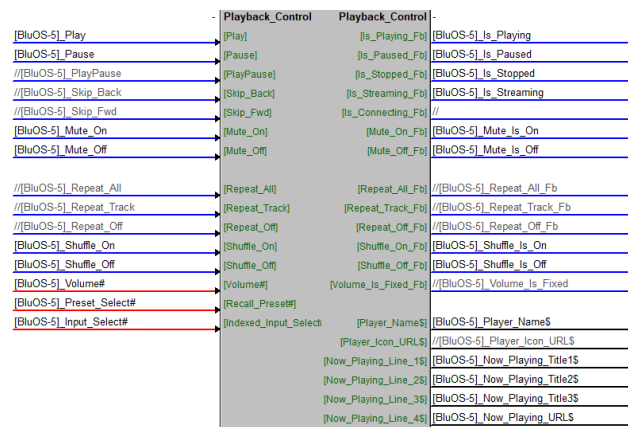


- Initialise – Added with v1.1.0 – pulse this ONCE at program start-up to initialise the module. A '1' can be generally used here though you may wish to stagger module initialisation to manage program start load and MediaPlayer client ordering.
- Debug – when enabled, this will provide additional output to the console which may aid in diagnosing communication issues.
- Is_Offline_Fb – when high, indicates that the player is currently offline
- Language_Code\$ - Set the device language dynamically by setting this signal to one of the following:

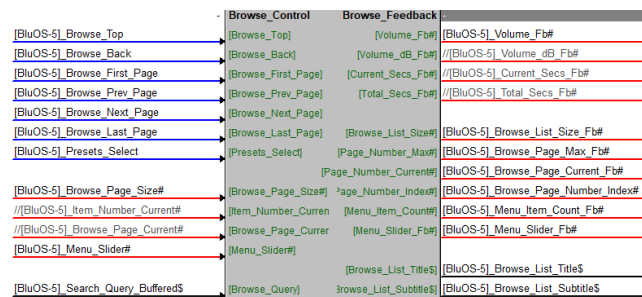
enGB	English (UK)
enUS	English (US)
zh	Chinese
cs	Czech
da	Danish
nl	Dutch
fi	Finnish
frFR	French (FR)
crCA	French (CA)
de	German
hu	Hungarian
it	Italian
ja	Japanese
ko	Korean
pl	Polish
pt	Portuguese
ru	Russian
sk	Slovak
es	Spanish
sv	Swedish

- CRPC_To/From_Device\$ - connection to the Media Player CRPC Router (if used)

- Play, Pause, PlayPause, Skip_* - Trigger these signals to control the player.
- Play_Fb etc. – Provides information about the current state of the device.
- Mute On/Off – controls the mute status.
- Volume_Is_Fixed_Fb – if high, the player's volume is fixed and mute/volume control will be unavailable. This can be used to show/hide volume controls.
- Repeat/Shuffle – control the play queue behaviour.
- Volume#, Volume_Fb# - the current player volume (if not fixed) as a full-scale value from 0-100% (0-65535). This controls the volume between the volume limits set for the specific device, so volume ranges can be balanced across devices and spaces.
- Current/Total_Secs_Fb# - the now playing time in seconds.
- Player_Name\$ - the name of the player.
- Player_Icon_URL\$ - a URL to the icon for the player.
- Now_Playing_x\$ - Now playing information used to drive non-Media Player UI information.



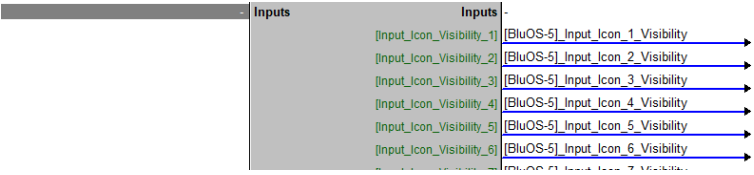
- Browse_* - control the browse context menu.
- Browse_Page_Size# - define the “page size” (number of entries) for the custom UI.
- Browse_Page_Current#, Menu_Slider# - control the location of the browse menu.
- Browse/Page_*_Fb# - various feedback elements describing the position in the current menu.
- Browse_Query\$ - send a serial string into this signal to initiate a search of the current menu.
- Browse_List_Title\$/Subtitle\$ - the title/subtitle of the current menu.



- The Browse_Items section provides selection and visibility of the elements in the current page (up to the Browse_Page_Size#) and drives item title, subtitle and icon (if available). Selection supports Press and Hold.



- The Inputs section provides a list of the available inputs (physical and services, where supported) so that they can be selected using the “Indexed_Input_Selection#” input.

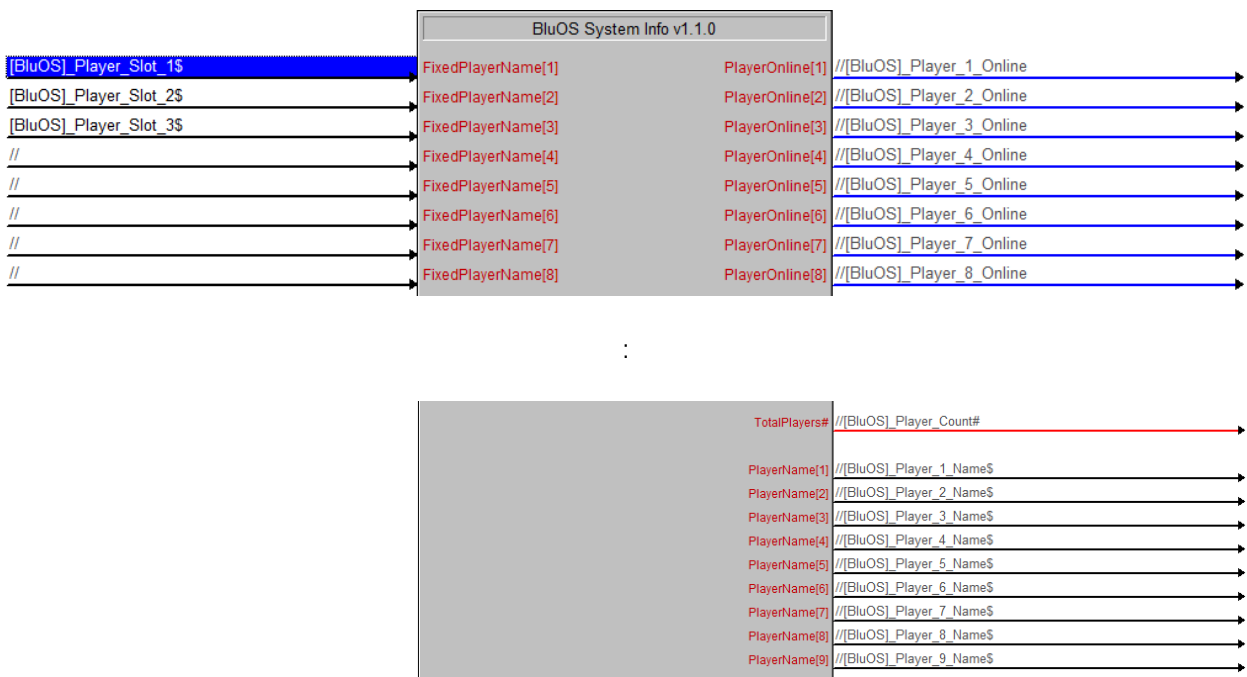


4.2 System Information

The System Info module provides dynamic information about the players currently available to the system.

It lists each player's name, the total number of players and the current online state of the player.

Note: Images show 1.1.0, but this is unchanged in the later release.



If the “FixedPlayerName” signals are not used/left blank, the order that BluOS players appear in the list is determined by the order in which each player responds to the discovery protocol. Do not rely on the position in the list to remain consistent across system reboots.

If the “FixedPlayerName” signals are populated with the EXACT name of each player in the system, then this will force players, once discovered, to populate a particular slot/determined order which provides a more consistent user experience and also allows for predefined groups to be managed.

The order of the list is consistent with the player indices in the grouping module – the system info module should therefore be used in conjunction with the grouping modules to present the player names for user selection.

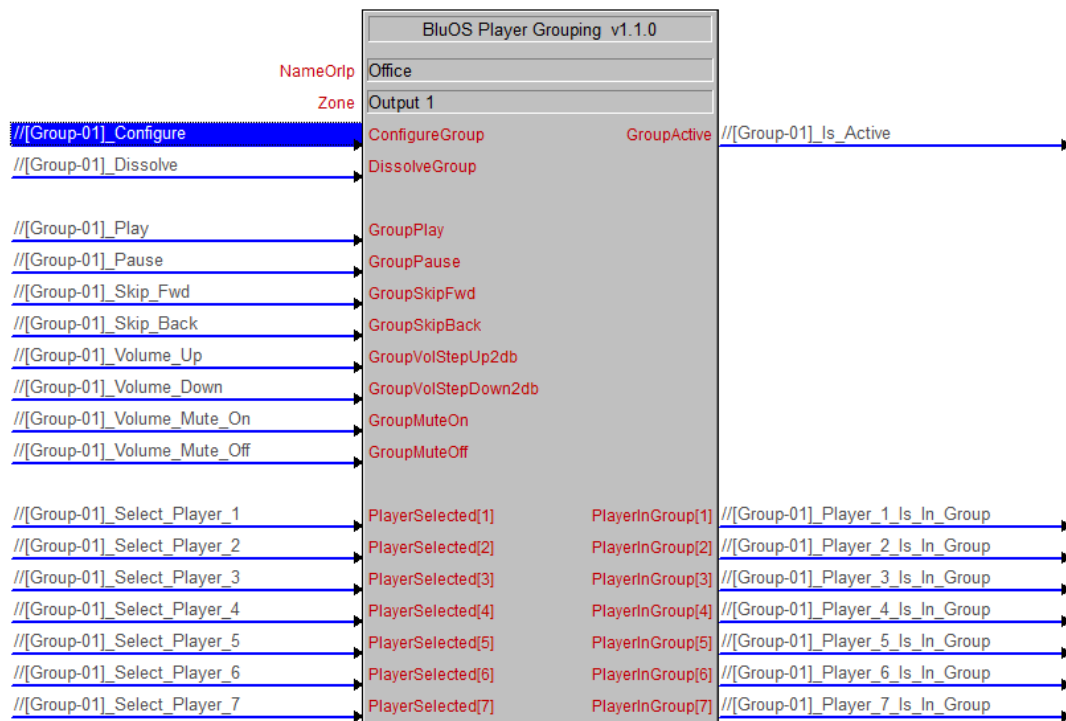
4.3 Player Grouping

The player grouping module provides a mechanism to allow users to select players to be grouped with a master player.

You can create as many groups as the system permits. Each module must specify a master player (identified by both the name, or static IP address, and the zone output – applicable when a multi-output device is used).

The module will reflect the current grouping state for the master player dynamically – i.e. it will change whether the group is modified via Crestron or one of the other BluOS client applications.

Note: Images show 1.1.0, but this is unchanged in the later release.



The signals are defined as follows:

- **ConfigureGroup** – pulse this signal to “take” the group configuration. Any PlayerSelected signal that is high at this time will ADD the player as a slave to the master player. Any PlayerSelected that is low at this time will be released from the group. If the master player is already a group master, then only the changes to the group will be requested.
- **DissolveGroup** – pulse to remove all slaves from the group and release the master.
- **Group[control]** – these controls provide convenient transport and volume control for the group.
NOTE: If any player in a group is defined with fixed volume, then the volume controls will not function for that player.
NOTE: If the group master is defined with fixed volume, then volume controls are non-functional.
- **PlayerSelected[x]** – hold high to select this player as a slave. Selecting the master player in this list does no harm, but also has no effect. E.g. If the group master (defined by the NameOrIp parameter) is player 4 in the system info module, then asserting PlayerSelected[4] has no meaning. PlayerInGroup[4] WILL go high when a group is active.
- **GroupActive** – this will go high when the group master is managing a group.
- **PlayerInGroup[x]** – indicates if the player (as denoted by the System Info module) is currently active in this group.
- **GroupSize#** - the number of players in the active group
- **GroupLabel\$** - the label of the current group (empty if the group is inactive). E.g. “Lounge + Kitchen”

5.0 Support

For technical support issues, contact Bluesound technical support at <http://support.bluesound.com> or email support@bluesound.com.

6.0 Known Issues

The following issues are known to exist.

Driver Version	Issue	Workaround
All	Some issues have been reported for content using extended characters.	None