Sonos Media Players

IMPORTANT:
UPDATING TO VERSION 2.x REQUIRES A FIRMWARE UPDATE TO VERSION 21.9.13 OR HIGHER

Updating the Driver from v.1.24:
Updating the driver from version 1.24 can be accomplished by downloading the driver from the RTI driver store and updating the existing Apex file with the new driver. Apex files built on v.1.24 should continue to work.

This driver provides network control of Sonos Media Players. It was built and tested using a Play:1 and a Connect: Amp. Version 1.1 was built and tested with a Connect, 2 Play 1's, a PLAYBAR and a Sub.

Note: The 'thumbs up' and 'thumbs down' buttons will not work because the Sonos API has not provided support for them.

Release History:
1.0: The original release of the driver.

1.01 adds driver events for transport changes, song changes and player volume up and down controls. Fixes an issue where not configuring one of your players caused metadata to stop

1.02 Makes player names in the configuration case-**in**sensitive, fixes icons not working for presets of local music, triggers transport events for slave players, improves UTF (non-English) display in favorites list and adds booleans for the selected player.

1.03 fixes a problem where some favorites don't show correctly and adds missing events

1.1: Properly handles players configured as stereo sources. Properly recognizes attached subwoofers. Properly recognizes surround speakers. Adds variable for which players are following a specific Player/Master. It changes some metadata logic so most tunes that used to not show metadata now will. It adds a command to access the optical input of a Playbar, along with a variable to show when a player, or the selected player, is a Playbar, it adds audio commands (volume and mute) for groups.

To upgrade to version 1.1 or later from an earlier version you MUST re-do your driver configuration page. The new driver requires that the serial number of the specific Sonos players be entered.

1.11 fixes a problem with the definition of the Repeat command that meant it was undefined when dragged onto a button but COULD be added from the properties pull downs.

1.14 Expands the number of instances to 32 and adds commands to adjust Bass, Treble, and Loudness

1.15 fixes Stop command not working and Sonos Playlist Artwork not showing up, and fixes instance number not incrementing in APEX. Adds visibility variables for transport function commands/buttons as well as repeat/shuffle. Adds a device icon representing the device.

1.17 fixes problems with some Preset and Direct Favorites icons not working

1.18 fixes an issue where changing the preset list would screw up the list in the driver.

1.19 deletes the Preset information if the Favorite it points to is deleted.

1.2: Changes the way the serial number is parsed to identify the player and changes the documentation to make this clearer. See the **Identifying Players** section below.

1.21: Adds some APEX changes to the events, there are no functional changes to commands or variables.

2.0: Added numerous features and improvements including:

- Support for web sockets
- Able to display a list of playlists and favorites and play them.
- A 'toggle group' command has been added as has the ability to add and remove groups.

-Newer devices area able to play a chime sound or a ‘clip’ from a URL.
- A command, ‘Play TV Input’ has been added under the ‘Play External Input’ section that allows for the Sonos device to switch back to the TV input (HDMI or optical).
2.1: Fixed an issue where a missed ping would result in connectivity loss. Removed unreachable “Stop” system event. Fixed issue where pause variables / system events would not trigger if listening to a radio station.
2.2: Fixed issue where driver would fail to recover if Sonos disconnected from network.
2.3: Updated minimum runtime version.

Identifying Players:

The driver doesn't need any network information, feel free to keep the system using DHCP. The driver needs to know the serial numbers of the players. The serial number can be found on the device or in the 'About My Sonos System' of any of the apps. It’s 19 characters long starting with 6 pairs of characters and ending with a single character. They should be separated by dashes, colons, or hashtags (See the very end of this file for a more detailed description). A subwoofer does not need to be added to the configuration as it will never need to be sent commands on its own. Surrounds don't need to be configured either if they are always surrounds. The driver can adjust to surround speakers going back and forth from being surrounds to individual speakers to stereo pairs, but this does increase the amount of processing power the driver uses so if they will always be surrounds feel free to leave them out of the configuration. The names in the configuration will show when programming in ID. When the system is running there are two 'Name' variable to choose from. You can either use the name for the player given in the configuration or the current name of the player in the system itself.

Stereo Pairs:

When two players are to be connected into a stereo pair you have to be cautious. You cannot do this in the driver, it has to be done in the app but the order in which you perform the steps is important. Stereo pairs are created in the Settings menu in the apps. You select one of the players and go to the selection called Create Stereo Pair. There you will select the device to be the other half of the pair. Once you've done that it will ask you to select which one of the two will be the left speaker in the pair. This seems obvious and innocent but has some important ramifications. The device selected as the left speaker also becomes the coordinator, or master of the pair and all the commands must go to it. The other speaker is considered 'invisible' in the system and cannot respond to any commands. The driver can only send commands to devices in its configuration so if you create a stereo pair, you only have to have one of them in the configuration, but it MUST BE the master (or left speaker) of the pair. This may not be the player that remains in the app. If both players are left in the configuration the driver will automatically send commands to the correct player.

Audio Player Control:

The driver gives you all the audio controls for up to 32 players, including a 'selected' player: Transport, volume, play mode (shuffle, repeat) etc. It also allows you to choose any of the programmed 'Favorites' in the Sonos System.

Favorites List:

The driver keeps a copy of the Sonos Systems Favorites list. It checks for updates to this list every 30 seconds, so any added favorites get quickly added. The main interface to this list is through the Favorites List variable. This is a complete list of Favorites on the Sonos System. Selecting an item in this list using the Select Favorite from List command plays the chosen favorite on the Sonos player.

In driver properties there is an option to change the formatting of the Favorites List formatting. The three options include:

- No description (title only)
- Description in parenthesis after title

- Description in second line (requires smaller text)

Direct Favorites Icon:

The first 20 Favorites are paired up with their metadata to create 20 buttons that can be labelled with titles, description, and artwork. The Play Direct Favorite from Icon command will play the Favorite for that Favorite Icon.

Preset Favorites:

In addition to the Direct Favorites there are a set of Preset Favorite Icons that can be altered by the end user to play whichever favorite they desire. Similarly, to the Direct Favorites there is metadata that can be used to create labeled buttons and a command (Select Preset Favorite) to play that Favorite. Unlike the Direct Favorite however, the end user can change the Favorite attached to these buttons.

A command called “Toggle Programming Preset Favorite on This Panel” starts the process of programming a Preset Favorite. As the name suggests this mode is only active on the panel that issues the command and pressing the button again will cancel the mode. There is a variable in the [Preset Favorites] section called Preset Write Mode that can be used to make it visible on the panel that the mode has been activated. The sample file uses it to expose the Favorites List which is hidden on that page unless being used to program the buttons. The user can then select one of the Preset Favorite buttons (the Select Preset Favorite Icon command), this will clear the current Favorite assigned to that button. After that the user selects a Favorite from the Favorites List. There is no need to change the commands on either the list or the buttons as those commands are both aware of the fact that the programming mode has been toggled on. Once the Favorite has been chosen from the list its information will be transferred to the button and the programming mode will be turned off.

The Preset Favorites are not view based, meaning they are the same 32 Presets across all devices. The advantage is that you can create an interface on a two-way device to reprogram the buttons on an RK1 or another one-way interface if desired.

Favorite Icon Metadata:

The graphics images returned as the icon for these buttons is a little inconsistent (some Services return it, some don't.) so it is best to have a 'generic' icon behind each of the dynamic images in case the Icon returned is blank.

Line Input Selection:

Commands exist to have any player play the Line Input of any other device. You can also send a command to switch a Playbar to its optical input.

Groups:

Any player can be synced to any other. When a player is following another its ‘is a follower’ variable will be set, when it is a Group Master its Group Master variable will be set. There are also boolean variables that will tell which players are following a given player (or the selected player). A separate command exits for removing that player from the group.

There are separate commands for player volume and group volume, as well as player mute and group mute. A Group volume command will change the volume of all the players in the players group, even if it's the only one. A regular volume command will change the individual player’s volume without changing the volume of any other players in the group. Player volume will let you adjust the balance between members of a group, group volume changes the volume of the group as a whole.

Driver Events:

Driver events will be triggered on a change in the players transport status. Events are available for Stop, Pause, Play and Buffering. An additional event is available for a change in song title.

A driver event will also be triggered every time a player is sent a volume up or down command. This will help when volume commands are being sent to selected players, one of which is a Sonos Connect. In that situation the events can trigger macros that will change volume on your receiver. Similar events exist for the mute command.

Transport Control:

Starting with version 1.15 of the driver there are visibility variables accompanying each of the transport commands. Using the variables allow you to display only the buttons for currently relevant commands. For instance, the Show Previous Command will be false when a Pandora station is playing. If you use that as visibility on a Skip Back button it will disappear when a Pandora Station is playing. In addition, a similar variable is available for Repeat and Shuffle; using that variable will allow your repeat buttons to only appear when they are relevant.

APEX Template Information:

The driver uses the following template: Sonos, Music Player

Detailed information on the configuration of players and their serial numbers:

Technically the driver just needs the MAC address of the players, and that MAC address is embedded in the serial number. Internally the driver uses the 12 characters of the MAC address without any separators at all, so it just dumps them once in gets them from the config. The driver uses the following rules to get the MAC address out of the text it pulls in from the configuration, if it fails at any of these steps it will not be able to find that player. It will print an error message to the log,

First it looks for two characters in the range of 0-9, a-f, or A-F (hex characters).

This has to be immediately followed by 0 or 1 of the characters: ':', '-', or "#" (colon, dash, or hash tag*)

Next is two more hex characters (pair 2 of the MAC)

0 or 1 of the separators

Two more hex characters (pair 3)

0 or 1 of the separators

Two more hex characters (pair 4)

0 or 1 of the separators

Two more hex characters (pair 5)

0 or 1 of the separators

Two more hex characters (pair 6)

All the following entries will work, and all point to the same player:

> B8-E9-37-B0-F6-28:7 -- This is the serial number cut and pasted from the About My Sonos System selection in the Help menu of the Sonos app
>
> B8-E9-37-B0-F6-28-7 -- This is the serial number shown on the bottom of a Play:1
>
> B8-E9-37-B0-F6-28 – It doesn't matter what, if anything, is after the last pair
>
> B8-E9-37-B0-F6-28YOURMAMAHATESSONOS – Like I said, it doesn't matter
>
> B8E937B0F628 – There don't have to be separators
>
> B8:E9:37:B0:F6:28 – For traditionalists, you can use colons
>
> b8:e9:37:b0:f6:28 – Hex characters can be upper or lower case
>
> B8E9-37:B0F6#28 – any combination of 1 or none of the three separators
>
> b8#E937:B0-f628asOPIJLK – Did I mention that the end doesn't matter?

* I accidently wrote in the description of an earlier version that there should be a dash or a hash tag when I had written the code to look for a dash or a colon (don't know what I was thinking, sorry). Instead of just changing the text to say colon and dash, I changed both the description and the code to allow any of the three.