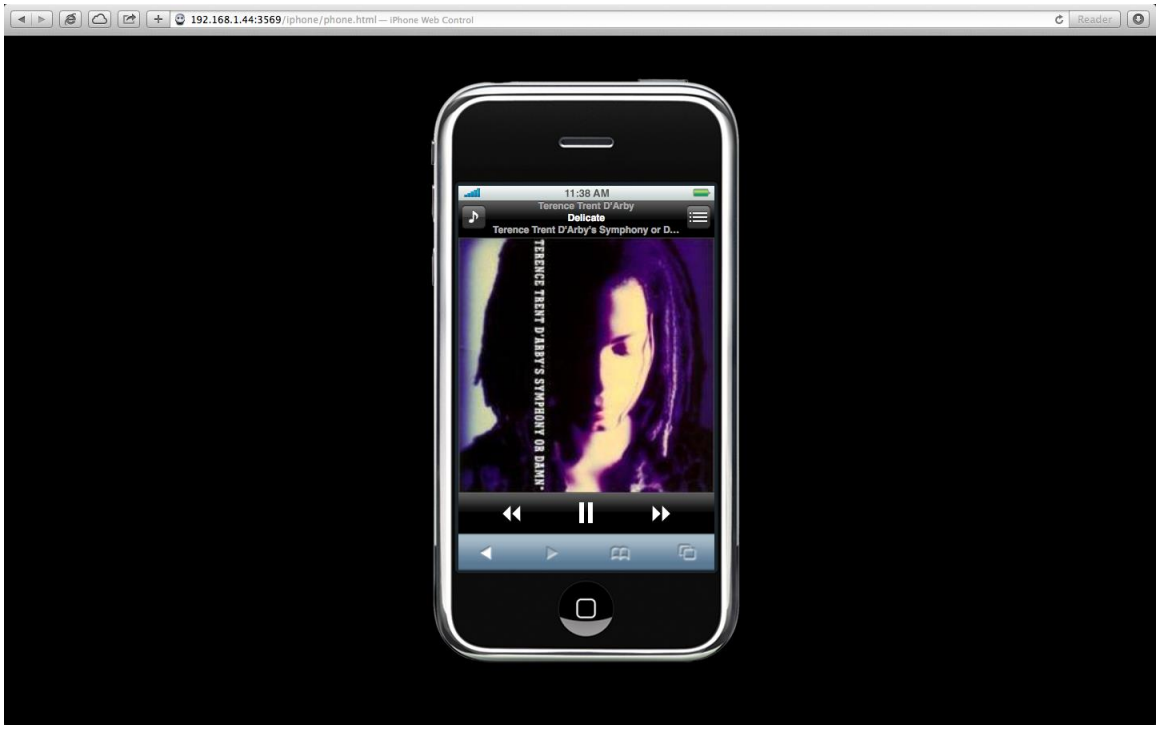
HCA Tech Note 403: Controlling iTunes from HCA

Have you ever wanted to control iTunes from within HCA? This article shows you how to do it. You can play specific artists, playlists, genres, albums and songs, turn on or off individual AirPlay speakers, and play/pause/stop music. Start your favorite playlist when you get home, wake up to your favorite song, pause iTunes when the phone rings with a ring sensor module. If you also have HCA issuing infrared commands via a Global Cache or BitWise device, it is easy to create a user-defined scene like "Entertain" that not only sets your lighting, but also turns on your music system and tells iTunes what to play.

iTunes utilizes DACP (Digital Audio Control Protocol), a non-public Apple protocol. This protocol uses HTTP commands for communication, but requires that the remote device controlling iTunes be "paired" with the iTunes installation being controlled. This pairing process utilizes Bonjour discovery protocol, and its purpose is to ensure that someone trying to remotely control iTunes has physical access to the computer iTunes is running on during the one-time pairing process. This process is moderately complex, and can be avoided by utilizing an inexpensive (currently $24.95) software package known as Signal, available at (http://www.alloysoft.com). Signal takes care of the pairing process, and allows control of iTunes from an iOS device, as well as a browser, using HTTP commands. Note that the computer running iTunes and Signal does not have to be the same computer as the one running HCA. Signal is available for both PC and Mac.



Signal interface in a browser window.

In addition to Signal, you need a piece of freeware, GNU Wget from https://www.gnu.org/software/wget/ . This piece of software allows you to send HTTP GET and

POST commands from a command line. Because HCA has a Run program element that can run Windows programs external to HCA, it can be used to run a batch file from a command line via Windows cmd.exe. This batch file can contain multiple commands, including running Wget, and can be passed parameters upon invocation to specify song, playlist, transport function, speaker, speaker state, etc. Because these commands are being issued from a command line, controlling iTunes doesn't require the use of any of HCA's eight interface ports.

There is a .zip file available on the HCA website that contains the .bat files described below. If you wish to use them, you need to edit them to use the IP address of the computer running iTunes, and to match the folder structure you use for HCA.

The first step is to install Signal on the machine that runs iTunes and Wget on the computer that runs HCA. That could, of course, be the same machine. Next create a folder – a good place is in your HCA documents area - and then unzip the .bat files into that folder. Edit the .bat files so they contain the IP address of the computer running Signal. The .bat files also contain two path names: One to the location of the WGET program and the other to the folder you created. Adjust them as needed.

In the discussion that follows, a number of .bat files are described. You can download a zip file containing these from www.HCATech.com/download/V12/Doc/HCA_iTunesBatchFiles.zip

NOTE: In the zip file the .bat files have file type ".bar". Anti-virus programs can get upset about zip files downloaded that contain .bat files. To make sure that doesn't happen, the file type for each of the files has been set to ".bar". After you unzip them, rename them to have file type ".bat". Windows may complain about changing the file type but just make the rename happen.

TRANSPORT COMMANDS

POST and GET commands are sent to Signal to issue commands to iTunes. Signal, by default, listens on port 3659. Transport commands to Signal running on a computer at IP address xxx.xxx.xxx.xxx take the form:

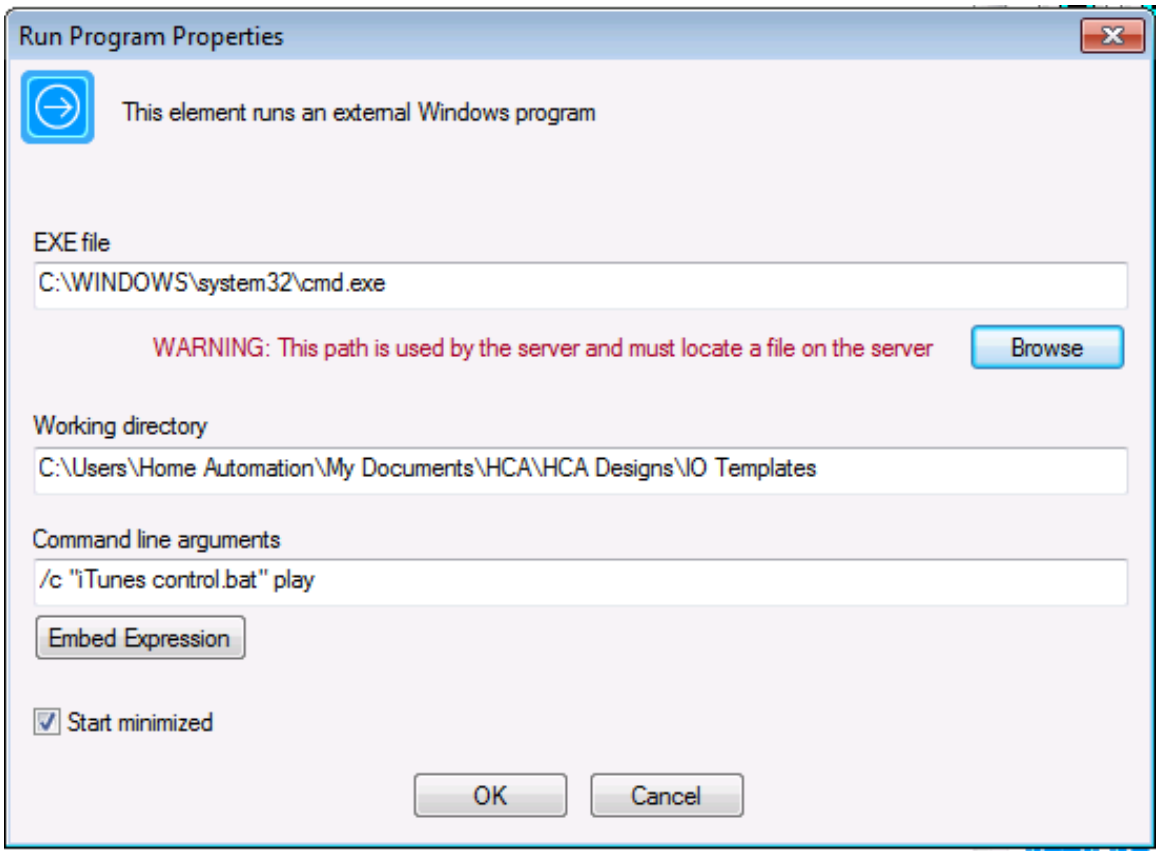
```
http://xxx.xxx.xxx.xxx:3659/[command]
```

where [command] can be:

play	(unpauses a paused song)
pause	(toggles between play and pause)
playpause	(toggles between play and pause)
previous	(jump to the previous song)
next	(jump to the next song)
stop	(stop whatever is currently playing)

In all the examples below, Signal is assumed to be running on a computer with IP address 192.168.1.44. Your computer IP address will be different.

An example screenshot of the Run program element to tell iTunes to "play" is shown here:



(This is the only screen shot that is shown in this document, since for subsequent commands, only the entry for “Command line arguments” in the dialog box changes.)

The EXE file specified is cmd.exe. This Windows system program executes commands as though they were typed in a Command Prompt window by running the .bat file.

The working directory is the location where the .bat file to be run is located. In this case, it is two levels below the HCA folder in \HCA Designs\IO Templates. You can place the .bat files wherever you want; just be sure the Working Directory field contains this location.

The last entry is the name of the batch file, and the parameter(s) to be passed into it. In the above example, there is one parameter: the “play” command. “/c” tells Windows to terminate the command window after the command is run.

If you have downloaded the zip file with the .bat files, unzipped them, and edited them for your system then you are ready to go. Here are the contents of the “iTunes control.bat” file.

Note: To make it easier to read the .bat file contents in this document, the end of each line is show by <CR><LF> since some wrap to the next line.)

```
REM HTTP POST command to Signal, with %1 specifying the control
function<CR><LF>
```

```
"C:\Program Files\GnuWin32\bin\wget" --post-data=""
http://192.168.1.44:3569/%1<CR><LF>

REM Delete the response file from Signal<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\%1"<CR><LF>
```

A response file is returned by the Signal HTTP server to the computer that runs the .bat file, and these eats up space if they aren't deleted. If you wanted to stop or pause playback, you would substitute "stop" or "pause" for "play" in the above Run Program Properties dialog box.

SPEAKER CONTROL

Speaker control commands to Signal running on a computer at IP address 192.168.1.44 take the form:

http://192.168.1.44:3569/xml?xml=<EnableSpeakerRequest><id>[SpeakerID]</id><enabled>[True/False]</enabled></EnableSpeakerRequest>

Before you can start working with speaker control you need to determine the ids assigned to your speakers. This command does that.

http://192.168.1.44:3569/speakers.json

Issuing this command results in a file named "speaker.json" in the folder from where the GET command is issued. When you open this file, it will contain a JSON (Javascript Object Notation) formatted list of the iTunes speakers, and the ID that iTunes uses to reference each speaker. (The developer of Signal has stated that these iTunes speaker IDs can change over time, although the author has not observed this.) You need to set the iTunes speaker control (via either iTunes itself or via the Signal interface in a browser) to "multiple speakers" if you want to stream to multiple speakers simultaneously. iTunes requires at least one speaker be enabled at all times if this setting is chosen. If you always use the computer speaker as the always-enabled speaker, you can then disable/enable all other speakers at will. Here are the contents of "iTunes speaker list.bat", a .bat file to do this:

```
REM HTTP POST command to Signal to list the speaker IDs<CR><LF>

"C:\Program Files\GnuWin32\bin\wget"
http://192.168.1.44:3569/speakers.json<CR><LF>

REM This will result in a response contained in a file named
"speakers.json" in the directory from which this file is run. <CR><LF>
```

The contents of speakers.json looks something like this:

```
{"speakers":[{"id":"0","name":"MyComputer","enabled":true},
             {"id":"97125487562928","name":"AppleTV","enabled":false},
             {"id":"73624592185","name":"Bath","enabled":true},
             {"id":"73624569483","name":"Kitchen","enabled":true},
             {"id":"87265770397","name":"Living Room","enabled":true},
             {"id":"137221033588","name":"Master Bedroom","enabled":true}]}
```

The Command line for the Run program element to tell iTunes to enable Bath speaker ID 73624592185 is:

```
/c "iTunes speaker.bat" 73624592185 true
```

Here are the contents of the "iTunes speaker.bat" file:

```
REM HTTP POST command to Signal, with %1 specifying the speaker ID and %2
specifying true or false for enabling or disabling the speaker<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<EnableSpeakerRequest><id>%1</id><enabl
ed>%2</enabled></EnableSpeakerRequest>"<CR><LF>

REM Delete the response file from Signal<CR><LF>
del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\xml@xml*.*"<CR><LF>
```

PLAYING ARTISTS, PLAYLISTS, GENRES, OR ALBUMS BY NAME

It is possible using Signal to tell iTunes to play all the songs by an artist, a particular playlist, songs of a particular genre, or a specific album.

Artists

This is the command to play all the songs by a particular artist:

```
http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Playli
sts/[ArtistName]</libPath></PlayMediaRequest>
```

where the artist's name replaces [ArtistName].

If you wanted to play all the songs by the artist "Sade", you would reference it by name in a Run element. The Command line for the Run program element would be:

```
/c "iTunes artist by name.bat" Sade
```

Here are the contents of the "iTunes artist by name.bat" file:

```
REM Parameter %1 below is the artist name<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Artis
ts/%1</libPath></PlayMediaRequest>"<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\xml@xml*.*"<CR><LF>
```

Playlists

This is the command to play all the songs in a particular playlist:

```
http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Playli
sts/[PlaylistName]</libPath></PlayMediaRequest>
```

where the playlist's name replaces [PlaylistName].

If you wanted to play the playlist "Chris' Favorites", you would reference it by name in a Run element. The Command line for the Run program element would be:

```
/c "iTunes playlist by name.bat" Chris'%%20Favorites
```

In the example above there is something very important to take note of, and that is the "%%20" between "Chris'" and "Favorites". Because this parameter is being passed into a URL in the batch file, the space needs to be encoded as %20, and because the "%" character is used to specify parameters in batch files, it is necessary to escape the"%" with a second "%".

Here are the contents of the "iTunes playlist by name.bat" file:

```
REM Parameter %1 below is the playlist name<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Playl
ists/%1</libPath></PlayMediaRequest>"<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\xml@xml*.*"<CR><LF>
```

Genre

This is the command to play all songs of a particular genre:

```
http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Playli
sts/[Genre]</libPath></PlayMediaRequest>
```

where the genre replaces [Genre].

If you wanted to play all songs of genre "Latin Jazz", you would reference it by name in a Run element. The Command line for the Run program element would be:

```
/c "iTunes genre by name.bat" Latin%%20Jazz
```

Here are the contents of the "iTunes genre by name.bat" file:

```
REM Parameter %1 below is the artist name<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Genre
s/Latin%%20Jazz</libPath></PlayMediaRequest>"<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\xml@xml*.*"<CR><LF>
```

Albums

This is the command to play all songs of a particular album:

```
http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Playlists/[Album]</libPath></PlayMediaRequest>
```

where the album name replaces [Album].

If you wanted to play the album “View From The Ground”, you would reference it by name in a Run element. The Command line arguments for the Run program element would be:

```
/c "iTunes album by name.bat" View%%20From%%20The%%20Ground
```

Here are the contents of the “iTunes album by name.bat” file:

```
REM Parameter %1 below is the artist name<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><libPath>/Music/Albums/%1</libPath></PlayMediaRequest>"<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA Designs\IO Templates\xml@xml*.*"<CR><LF>
```

PLAYING A SPECIFIC SONG

There are two ways to play a specific song. One way is to put the song into its own playlist, and then play the playlist by referencing its name as shown previously.

The second method is to reference the song by the ID number that iTunes uses to uniquely identify it. All song, playlists, and albums have a unique ID. In order to find out what the ID is, it is necessary to send a particular command to Signal, and Signal returns a file containing all of the song IDs. You can then open that file in a text editor like Notepad, search for the song, and find its corresponding ID.

A batch file “iTunes song list.bat” to do this is:

```
REM HTTP GET command to Signal to list the songs IDs (up to 3000)
<CR><LF>

"C:\Program Files\GnuWin32\bin\wget"
http://192.168.1.44:3569/media/Music/songs.json?start=0&count=3000&includeTotalCount=true<CR><LF>

REM This will result in a response contained in a file named
"songs.json@start=0"<CR><LF>
```

This results in a file in the folder from which the .bat file is run and is named songs.json@start=0. Here are the partial contents of this file from my machine.

```
{"path":"/Music/songs","start":0,"totalCount":-
1,"items":[{"mediaId":2559,"type":"music","title":"A - La -
```

```
Ke"},{"mediaId":1239,"type":"music","title":"A.D.I.D.A.S."},{"mediaId":53
0,"type":"music","title":"Aaron's Party (Come Get
It)"},{"mediaId":1976,"type":"music","title":"Abilene"},{"mediaId":509,"t
ype":"music","title":"About A
Girl"},{"mediaId":2136,"type":"music","title":"Abracadabra"},{"mediaId":2
165,"type":"music","title":"Abracadabra"},{"mediaId":537,"type":"music","
title":"Absolutely (Story Of A
Girl)"},{"mediaId":1275,"type":"music","title":"Achilles Last
Stand"},{"mediaId":1804,"type":"music","title":"Across the
View"},{"mediaId":1396,"type":"music","title":"Act of Contrition"}
```


If you wanted to play the song “Abracadabra”, you would reference its ID of 2136 in a Run element. Find “Abracadabra “ in the above output and you can see that its media id is 2136. The Command line arguments for the Run program element would be:

```
/c "iTunes media by ID.bat" 2136
```

Here are the contents of the “iTunes media by ID.bat” file:

```
REM Parameter %1 is the media ID of the song, playlist, or album to
play<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<PlayMediaRequest><id>%1</id></PlayMedi
aRequest>"<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\xml@xml*.*"<CR><LF>
```

ENABLING OR DISABLING REPEAT AND SHUFFLE PLAY

Repeat

Commands to enable/disable repeat play take the form:

```
http://192.168.1.44:3569/xml?xml=<SetRepeatStateRequest><repeatState>[ena
ble/disable]</repeatState></SetRepeatStateRequest>"
```

If you wanted to enable repeat play, you would set the parameter to the batch file to “true”. Setting it to false would disable repeat play. The Command line arguments for the Run program element would be:

```
/c "iTunes repeat.bat" true
```

Here are the contents of the “iTunes repeat.bat” file:

```
REM Parameter %1 below is the repeat state<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<SetRepeatStateRequest><repeatState>%1<
/repeatState></SetRepeatStateRequest>"<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\xml@xml*.*"<CR><LF>
```

Shuffle

Commands to enable/disable shuffle play take the form:

```
http://192.168.1.44:3569/xml?xml=<SetShuffleStateRequest><shuffleState>>[
enable/disable]</shuffleState></SetShuffleStateRequest
```

If you wanted to enable shuffle play, you would set the parameter to the batch file to “true”. Setting it to false would disable shuffle play. The Command line arguments for the Run program element would be:

```
/c "iTunes shuffle.bat" true
```

Here are the contents of the “iTunes shuffle.bat” file:

```
REM Parameter %1 below is the shuffle state<CR><LF>

"C:\Program Files\GnuWin32\bin\wget" --post-data=""
"http://192.168.1.44:3569/xml?xml=<SetShuffleStateRequest><shuffleState>%
1</shuffleState></SetShuffleStateRequest>"<CR><LF>

del "C:\Documents and Settings\Home Automation\My Documents\HCA\HCA
Designs\IO Templates\xml@xml*.*"<CR><LF>
```