

Appendix A: Media Server Commands

\$Date: 2008-07-25 13:25:42 -0400 (Fri, 25 Jul 2008) \$ for MediaServer V03

Command	Details	Parameters (* denotes optional parameter)
nop	do nothing, only used for testing	none
disconnect	mark a stream as being disconnected (stops music) <i>See example disconnect.xml</i>	stream - Index of which stream to use (1-3)
Connect *URL Encode use msconnect	mark a stream as being connected (starts music if something in queue) <i>See example connect.xml</i>	stream - Index of which stream to use (1-3)
Pause *URL encode use mspause	Pause a stream <i>See example pause.xml</i>	stream - Index of which stream to use (1-3)
play	Play a stream (If something in queue) This unpauses a stream. <i>See example play.xml</i>	stream - Index of which stream to use (1-3)
playthisnext	move a song up in the what's next queue	stream - Index of which stream to use (1-3) idQueue - id of queue entry of song to play next (get this from query-whatsnext) * skipsong Flag, if true will cause the currently playing song to abort playing, skipping to the next song
Playselection	adds selection to song	stream - Index of which stream to

Command	Details	Parameters (* denotes optional parameter)
*Cannot be used with URL encoding	queue See examples playselection.xml and playselectionmultiple.xml	use (1-3) type - One of Genre,Artist,Album,Song One of: idselection - id of a Genre,Artist,Album,Song to add to play queue SelectionList – containing an XML list of Genre,Artist,Album,Song ids to add to play queue. Each item in list named <item#>, where # starts with 0 *clearqueue Flag, if true will cause the song queue to be cleared *skipsong Flag, if true will cause the current song to be skipped *enqueuehead Flag, if true will cause the new songs to be in front of existing songs (otherwise they are enqueued after all other user requested songs (but before any PMA enqueued songs)
playquery *Cannot be used with URL encoding	execute a query, then add every result to the song queue. This is deprecated, use playselection instead.	stream - Index of which stream to use (1-3) query – sql query returning list of idSongs to add to song queue *clearqueue Flag, if true will cause the song queue to be cleared *skipsong Flag, if true will cause the current song to be skipped *enqueuehead Flag, if true will cause the new songs to be in front of existing songs (otherwise they are

Command	Details	Parameters (* denotes optional parameter)
		enqueued after all other user requested songs (but before any PMA enqueued songs)
removesongfromqueue *Cannot be used with URL encoding	remove a song from the play queue	stream - Index of which stream to use (1-3) idQueue - id of queue entry (obtained from query-whatsnext)
nextsong	abort playing the current song (this will cause song voting to occur unless override parameter is given) <i>See example next-song-1.xml</i>	stream - Index of which stream to use (1-3) * vote - flag, defaults to true. If set causes this song to get voted down (since it was skipped). So to prevent voting include a vote parameter set to 0
previoussong	abort current song, play last song played <i>See example prev-song-1.xml</i>	stream - Index of which stream to use (1-3)
rewind	restart current song	stream - Index of which stream to use (1-3)
repeatsong	place currently playing song in the song queue again <i>See example repeat-song-2.xml</i>	stream - Index of which stream to use (1-3)
trackedit	modify information about a song	idsong id of song to modify field - field to modify, one of Genre,Artist,Album,Song,trackno,year name - New value to set field to
albumedit	modify information about an album. Changes to artist or genre are applied to all	idalbum id of album to modify field - field to modify, one of

Command	Details	Parameters (* denotes optional parameter)
	songs in the album.	Genre,Artist,Album name - New value to set field to
backuprestore	Initiate a backup or restore. Note that the system reboots after a restore--in the case of a TCP connection, this will cause the connection to be lost.	action one of backup restore
addmood	create a new theme	moodname - name of theme to add
deletemood *Cannot be used with URL encoding	remove a theme	idmood - id of theme to remove (get from query-themes)
setthemeparam	modify a theme setting. <i>See example</i> <i>setthemeparam.xml</i>	idmood id of theme to modify param - name of parameter to modify, one of iconfilename,whattoplay,pmaeffect,pmaprefers,pmaplaylist,automaticadd music value - New value to set parameter to

Command	Details	Parameters (* denotes optional parameter)	
		If param is	The value should be
		iconfilename	Name of icon file in the icons/themes directory
		whattoplay	0 = Single song 1 = 2 in a row 2 = 3 in a row 3 = Entire Album
		pmaeffect	0 = Only Random 1 = Mostly Random 2 = Mostly PMA 3 = Only PMA
		pmaprefers	0 = Popular 1 = Longest Since Played 2 = Recently Added
		pmaplaylist	0=not a playlist 1=theme is a playlist, pma is ignored
		automaticaddmusic	0=music must be added to theme manually 1=music requested by user will get added to theme

Command	Details	Parameters (* denotes optional parameter)
renamemood *Cannot be used with URL encoding	rename a theme	idmood - id of theme to rename (get from query-themes) moodname - new name of theme
addcategoryfromidsongtomood	Looks up the id in the type specified of the song specified by idSong, and adds that category to the theme.	idmood - id of theme to exclude songs to (get from query-themes) type – category of item to exclude. Can be one of: "Song", "Album", "Artist", "Genre" idSong - id of a song. The category id specified by the type parameter is looked up on this song, and that category is added to the theme
excludecategoryfromidsongfrommood	Looks up the category specified of the song specified by idSong, and excludes that category from the theme.	idmood - id of theme to add songs to (get from query-themes) type – category of item to add. Can be one of: "Song", "Album", "Artist", "Genre" idSong - id of a song. The category id specified by the type parameter is looked up on this song, and that category is added to the theme
addcategorytomood	simpler interface to addquerytomood, add all songs of a single category to a theme	idmood - id of theme to add songs to (get from query-themes) type – category of item to add. Can be one of: "Song", "Album", "Artist", "Genre" id - id of item being added. For example, If type is set to "Genre" then this should be set to the value read from an idGenre (but the field is still called 'id')

Command	Details	Parameters (* denotes optional parameter)
		Note that the list of songs is not directly removed from the theme, but instead a query is stored, and this query is evaluated when tracking what songs are in a theme. If the results of the query ever change (for example, some songs are added or removed from a genre, then the theme will reflect this).
excludecategoryfrommood	simpler interface to addquerytomood, add all songs of a single category to a theme	idmood - id of theme to exclude songs from (get from query-themes) type – category of item to exclude. Can be one of: "Song", "Album", "Artist", "Genre" id - id of item being excluded. For example, If type is set to "Genre" then this should be set to the value read from an idGenre (but the field is still called 'id')
addquerytomood *Cannot be used with URL encoding	add a query returning a song list to this theme. This is deprecated, use addcategorytomood instead.	idmood - id of theme to add to (get from query-themes) query - sql query returning list of idSongs to add. Only supports certain queries, see the Query table in section 5.2 in XML Interface Protocol Note that the list of songs is not directly added to the theme, but instead this query is stored, and this query is evaluated when tracking what songs are in a theme. If the results of the query ever change (for example, some songs are added or removed from a genre, then the theme will reflect this).
excludequeryfrommood *Cannot be used with URL encoding	add a query returning song list to this theme, this list will be used to exclude songs from this theme. This is deprecated, use	idmood - id of theme to exclude from (get from query-themes) query - sql query returning list of idSongs to exclude. Only supports

Command	Details	Parameters (* denotes optional parameter)
	excludecategoryfrommood instead.	certain queries, see the Query table in section 5.2 in XML Interface Protocol Note that the list of songs is not directly removed from the theme, but instead this query is stored, and this query is evaluated when tracking what songs are in a theme. If the results of the query ever change (for example, some songs are added or removed from a genre, then the theme will reflect this).
deletemooditem *Cannot be used with URL encoding	permanently remove a selection rule from a theme	idmood - id of theme to remove the selection rule from idmoodmember id of selection rule to remove. Obtained from a themeselectionlist query.
deletesongsfromcategory *Cannot be used with URL encoding	permanently remove songs from the system described by category (it is assumed that user confirmation (are you sure?) has already occurred) Obviously this should be used with care. Songs cannot be recovered once deleted.	type – category of items to delete. Can be one of: "Song", "Album", "Artist", "Genre" id - id of item being deleted. For example, If type is set to "Genre" then this should be set to the value read from an idGenre (but the field is still called 'id'). All songs in that genre will be permanently deleted.
deletesongs *Cannot be used with URL encoding	permanently remove songs from the system described by sql query returning song list (it is assumed that user confirmation (are you sure?) has already occurred) This is deprecated, use deletesongsfromcategory	query - sql query returning a list of idSong to be permanently removed from the system

Command	Details	Parameters (* denotes optional parameter)
	instead.	
votesong *Cannot be used with URL encoding	move song up or down in popularity in theme	idmood - id of theme to vote in (get from query-themes or status update) idsong - id of song to vote on (get from status update or various song queries) direction - vote up or down, 1 = up (increase ranking, make more popular), -1 = down
ordermooditem *Cannot be used with URL encoding	modify where in the theme list (priority) a song appears	idmood - id of theme to reorder song in (get from query-themes or status update) idsong - id of song to move (note it is assumed that this song is actually IN this theme) offset - how far up or down to move this song in this theme. Positive numbers move the song to a higher numbered position in the list, which is later in a playlist, or less popular in a theme list. Negative numbers move the song to a lower numbered position, which is sooner in a playlist, or more popular in a theme list
selecttheme *Cannot be used with URL encoding	switch specified stream to specified theme <i>See example set-theme.xml</i>	stream - Index of which stream to use (1-3) idtheme - id of theme to switch to * abortcurrentsong - flag, defaults to 1. If set, aborts the current song, if clear, then current song is allowed to finish
selectfavorite	switch specified stream to theme specified by the given favorite index	stream - Index of which stream to use (1-3)

Command	Details	Parameters (* denotes optional parameter)
	See example select-favorite.xml	idFavorite - which favorite (mapped to keys on the remote control), currently 1-3
settime	Sets the current time	timestring – HH:MM <i>*24 hour format</i>
addmessage	Create a message in the messaging system. If you use this, be sure to make it clear that the message came from your system, and not the SMS3.	messagetitle – short text of message which appears in the message list messagebody – text of message which appears when the message is viewed. There is no scrolling, so the message must fit on a single screen. HTML is allowed. *messageseverity – How important this message is. Defaults to 3. 1 = announcement: Messages from Russound about new features, etc. 2 = informational: Nothing is wrong, just want the user to know something (probably make updates use this severity) 3 = warning: The system hasn't failed, but was unable to perform some action (like Ripping a CD when there isn't enough hard disk space left) 4 = error: something went wrong, but we can recover from it (like it overheated) 5 = critical: something went wrong, and we can't recover from it (like a corrupt hard drive)