CRESTRON CORE3 MEDIA OBJECTS

REVISION HISTORY

Version	Date	Comments
1.0.0	4/4/2012	Initial Document
1.1.0	4/6/2012	Additional updates
1.2.0	4/11/2012	Updates from Spec Review 1
1.3.0	4/16/2012	Updates from Spec Review 2
1.4.0	4/25/2012	Refactored
1.5.0	7/24/12	Added create action Added clearPreset action Added textLine1 – textLine5 Added shuffle and repeat action Added playerIcon property
1.6.0	7/26/12	Added StatusMsgResponse action Added StatusMsgItems property Added show and numOfItems as parameters to StatusMsg Updated Shuffle and Repeat actions. Added Shuffle and Repeat properties Added QuickList action (Used to quickly display a list of presets)
1.6.1	7/31/2012	Added ScanUp and ScanDown actions
1.7.0	8/5/2012	Removed numOfitems and moved StatusMsgItems property to statusMsg as a parameter Updated Keypad action to be KeypadInput with a string parameter
1.8.0	8/7/2012	Renamed the second instance of Find to Search under list Actions.

1.8.1	8/20/2012	Renamed BandSelectpreviousCategory to Previous Category Re-added TextLine1-5 Changed ShuffleState Property value from 2=All to 2=Album Changed output parameter to return value for IMediaPlayer.GetMenu and IMenu.Find
1.9.0	8/22/2012	Added LocalExit in StatusMsg Added State in StatusMsgResponse
1.9.1	8/27/12	Removed Keypad from ListSpecificFunctions Deleted KeypadInput from the Menu Actions
1.9.2	8.30.2012	Added Band Property
1.9.3	8/31/2012	Added Busy and StatusMsg as a property for the Media Player object
2.0.0	9/12/2012	Implementation vs Spec Update Added Star action Removed AckStatusMsg, UserInput, SavePreset, ClearPreset, RecallPreset actions Added Rating property Removed MaxPresets and PresetNames properties Updated PropertiesSupported description to include AlbumArt and ProgressBar. Updated naming of media player events – StatusMsg to StatusMsgChanged and Busy to BusyChanged GetData changed to an object accepting three items – L1, L2, URL Removed Insert, Remove, and Search actions. Removed InsertSupported and RemoveSupported FindDesired changed to string Updated naming of menu events – Updated to ListChanged, Clear to ClearChanged, and Busy to BusyChanged Changed TextLine1-5 to TextLines array
2.0.1	9/17/2012	Fixed GetData return parameters to be camel cased Actually made FindDesired a string from array as noted in 2.0.0

		Added missing comments in v2.0.0 regarding TextLines Removed obsolete note from some metadata properties
2.0.2	9/25/12	Fixed GetData return parameters to be L1, L2, URL
2.0.3	10/19/12	Updated description for StreamState
2.0.4	10/25/12	Updated description for IMediaPlayer.GetMenu and IMenu.Find
2.1.0	11/30/12	Day 2 Features
2.2.0	12/03/13	Added designation of Object for Rating property
2.3.0	12/10/13	Added UpdateCount property
3.0.0	01/19/15	Day 3 Features - Added Advanced menu action
3.0.1	02/02/15	Added IsMenuAvailable property in the Menu object
3.0.2	04/22/15	Added BackToTop menu action
3.0.3	05/07/15	Updated AlphanumericMasked to be characterMasked. This matches what was implemented in the Media Player and devices.
3.0.4	9/16/15	Updated PlayerIcon to include Spotify and Jukebox
3.0.5	11/9/15	Updated Actions to include Seek and menu actions to include Favorites.
3.0.6	2/8/2016	Added AlbumArtURL property to NowPlaying object. Added URLNAT in return of GetData method for Menu Object.
3.0.7	7/28/2016	Added "none" as a userInputRequired option for Status Messages (Requires Smart Graphics 2.11.6 or later). The localExit property in Status Message events and properties is now deprecated. Added AlbumArt and ProgressBar properties.
3.0.8	10/17/2016	Added external app launch support: LaunchURI & LaunchIconURL properties, and LaunchExt method

TABLE OF CONTENTS

1 ABSTRACT

This document defines several abstract objects used for Core 3 media playback, namely a Menu and a Player object. The Core 3 User Interface (defined elsewhere) interacts with these objects in order to provide the end-user media experience. This document defines these objects solely in terms of Actions, Properties and Events, in keeping with the spirit of the recent Crestron Communications Protocol document.

This document is intentionally protocol (serialization) agnostic. The purpose of defining things this way is to allow us to more easily perform a higher-level design first before getting bogged down in lower-level details.

2 MEDIAPLAYER OBJECT

This object is an entity that has access to some audiovisual content and is capable of playing it.

TBD: Move out the user prompting/interaction into a separate object? Not sure if radio-specific stuff belongs in here or not.

Note: Shuffle and Repeat functions have been intentionally omitted.

2.1 ACTIONS

Except where noted otherwise, every action returns a **result** boolean indicating success or failure of the requested action.

action	*Description*		
Play	Starts audio playback.		
PlayAll	Starts playback of all content that player has access to.		
Pause	Temporarily halts audio playback.		
Stop	Permanently halts audio playback.		
Rewind	Seeks backward in the current track.		
Ffwd	Seeks ahead in the current track.		
NextTrack	Jumps to the next track.		
PreviousTrack	Jumps to the most recent track.		
NextPreset	Starts playing the next preset.		
PreviousPreset	Starts playing the previous preset.		
TuneUp	For radios, increases the frequency.		
TuneDown	For radios, decreases the frequency.		
ScanUp	For radios, can be used for scanning through or seeking the next station in the direction of increasing frequency.		
ScanDown	For radios, can be used for scanning through or seeking the next station in the direction of decreasing frequency.		
NextCategory	For Sirius/XM, tunes to the next category		
PreviousCategory	For Sirius/XM, tunes to the previous category		
BandSelect			
input parameters	*name*	*type*	*Description*
(Optional)	**band**	string	Desired band: AM/FM/XM/etc.
Shuffle	Changes shuffle mode. Mode is specified in **ShuffleState**		

Repeat	Changes repeat mode. Mode is specified in **RepeatState**		
ThumbsUp	Records positive user feedback for the current track.		
ThumbsDown	Records negative user feedback for the current track.		
BookmarkArtist	Records the name of the artist for the current track.		
BookmarkTrack	Records the name of the current track.		
LaunchExt	Notifies the player the UI has launched an external app		
input parameters	*name*	*Type*	*Description*
	instance	*number*	*Unique identifier from the Menu Object used by the client*
Star	Records number of star feedback for the current track.		
input parameters	*name*	*Type*	*Description*
	id	*number*	*Star rating given Ex. Value of 1 would mean 1 star, value of 5 would mean 5 star*
UserInput	Textual input from the user. Could be used to enter a stream provider-specific PIN number for example. Player will use the **statusMsg** event to prompt the user.		
input parameters	*name*	*type*	*Description*
	text	string	Text entered by the user.
StatusMsgResponse	User action on a status message displayed		
input parameters	*name*	*type*	*Description*
	id	number	Which button was selected. Value can be from 1-5, or -1 if the status message was closed without using a button.
	userInput	string	Textual input from the user. Could be used to enter a stream provider-specific PIN number for example.
	state	number	Valid values: 0 – closed 1 – open Used to indicate status message was closed due to local actions on the interface
	localExit	boolean	True if statusmsg was locally dismissed on the client side. Ex. Either from timeout or cancel button or other means of dismissing message (touching outside message)

DirectTune	Used for real radios to tune directly to the specified frequency.		
input parameters	*name*	*type*	*Description*
	band	string	AM/FM/XM etc.
	frequency	string	frequency of desired station
GetMenu	Returns the unique object instancename of the Menu object associated with the specified client UUID. If there is no existing association, it will find an unassociated object and return that instancename, creating an association.		
input parameters	**uuid**	string	The uuid of the client
Return value	**instancename**	string	Returned value is a dictionary consisting of a single item whose key is “instancename” and whose value is the menu ID to use for future actions.
Seek	Jumps to a specified time in the current track.		
Input paramters	*name*	*type*	*Description*
	time	number	Desired time to jump to.
Return value	**ElapsedSec**	number	Returned value is the time at which the track should start playing from.

2.2 PROPERTIES

name	*type*	access	*description*
Version	number	read-only	A numeric version number of this object.
Instance	number	read-only	Numerical value uniquely identifying this player on the device where it resides.
Language	string	read-write	Can be used to select the current desired language. Choices are: English, French, Spanish, Dutch, Russian, Chinese, Japanese
ActionsSupported	array of strings	read-write	A list of optional functions that are supported. A player user interface could query this property and create buttons as needed. The base set of functions correspond to the actions listed above, e.g. Play, Pause, Stop, Rewind, Ffwd, PreviousTrack, NextTrack,

			Previous Preset, NextPreset, ThumbsUp, ThumbsDown, BookmarkTrack, BookmarkArtist, BandSelect, Shuffle, Repeat, QuickList, Rating, Star, Seek. This could be used to accommodate future unknown functions from new streaming audio providers. Radios could expose band select actions here. A player user interface could set this property to disable certain features, for example showing certain streaming providers or setting the top level browse menu.
ActionsAvailable	array of strings	read-only	A list of all of the supported functions which are currently available. Any strings listed here must have been defined in **actionsSupported**.
PropertiesSupported	array of strings	read-only	A list of the properties supported by this object. Can be used by the user interface to customize itself. Mandatory Properties: Busy
RewindSpeed	number	read-write	How fast the rewind is in terms of 1x, 2x, 3x etc.
FfwdSpeed	number	read-write	How fast the fast forward is in terms of 1x, 2x, 3x etc.
ProviderName	string	read-only	Name of the current streaming audio provider, if any. Examples include Pandora, Sirius XM, Internet Radio, Librivox, etc.
PlayerState	string	read-only	Indicates what the player is doing at the present moment. The set of states includes: **stopped**, **playing**, **paused**, **rewinding**, and **forwarding, scanning....** Note that user interfaces can use the ...Available properties for enabling or greying out specific buttons.
PlayerIcon	number	read-only	Indicates which icon to show

			0. Default player image 1. XM 2. Sirius 3. AMFM 4. ADMS 5. iPod 6. Internet Radio 7. Satellite Radio 8. Pandora 9. Librivox 10. Spotify 11. Jukebox 12.
PlayerIconURL	String	Read-only	URL of PlayerIcon associated with the currently controlling server
PlayerName	string	read-only	Name of the current player, if any. On the CEN-ISERVER it is the user’s custom name, e.g. Jimmy’s iPod
LaunchURI	String	Read-Only	URI to be used by the UI when launching an external app.
LaunchIconURL	String	Read-Only	Location of an icon to show for the LaunchExt action. Will override PlayerIconURL if the LaunchExt action is available on the UI.
StreamState	string	read-only	Indicates what the audio stream is doing at the present moment: **idle, resolving, connecting, buffering,** and **streaming.** This also can be used to display statuses such as “Device Offline”, “External Storage Offline”, “You Cannot Skip More Than 6 Times”
MediaType	string	read-only	Indicates the kind of media playing. Choices are **video** or **audio**.
Title	string	read-only	Title of the current track
Artist	string	read-only	Artist for the current track
Album	string	read-only	Album for the current track.
Genre	string	read-only	Genre for the current track.
Composer	string	read-only	Composer for the current track.
AlbumArt	boolean	read-only	Toggles between showing Album Art and the Keypad. **True** to show the Album Art,

			False to show the Keypad. **DirectTune** may override this property.
AlbumArtUrl	string	read-only	URL of album art associated with the current track.
AlbumArtUrlNAT	string	read-only	Secondary URL of album art associated with the current track. This URL is used for userPAT
StationName	string	read-only	Mainly for internet radio, the name of radio station which is playing the current track.
ProgressBar	boolean	read-only	Hides or displays the progress bar. **True** to display the progress bar, **False** to hide it.
ElapsedSec	number	read-only	How much time has passed in the current track. A player **StateChanged** event including this property will be sent at the beginning of the track and whenever an action is invoked which would cause this number to change in an unexpected way, e.g. **Ffwd**, R**ewind**, etc. The player user interface may read this property periodically in order to ensure that the displayed time is reasonably synchronized with the actual player time.
TrackSec	number	read-only	How many seconds long the current track is.
TrackNum	number	read-only	Indicates track number within a set of tracks.
TrackCnt	number	read-only	Indicates the total number of tracks within the current set.
NextTitle	string	read-only	The title of the upcoming track.
MediaReady	boolean	read-only	Used to indicate whether or not the "media" is ready to be accessed by the player. In the case of the CEN-ISERVER this would be used to indicate that an iPod was docked and ready to be played. For a product with removable storage media, it could indicate that the media was mounted and ready to be used. If the value of this property is

			false, no Actions can be performed.
ShuffleState	number	Read-only	Used to indicate shuffle mode. 0 → Off 1 →Tracks 2 → Album
RepeatState	number	Read-only	Used to indicate repeat mode. 0 → Off 1 →Tracks 2 → All
TextLines	String array	read-only	Text to be displayed on meta data lines. Strings should be listed in order of importance.
Band	String array	read-only	Used to indicate all bands (providers) supported "AM" "FM" Etc.
StatusMsg	Object	read-only	Contains the information associated with the last StatusMsg event that was sent. Values in this property will be used to show Status Messages to connecting clients.
parameters	*name*	*type*	*Description*
	text	string	The message to the user.
	timeoutSec	Number	The recommended number of seconds that the message should be displayed for. The user interface may choose to ignore this number. 0 indicates to keep the message displayed indefinitely
	userInputRequired	String	Indicates type of user input required. Choices are **confirmation** (yes/no/cancel choices should be specified in textForItems), **alphanumeric, numeric**, **characterMasked**, **numericMasked,** and **none** (only the text field will be displayed, no buttons will show, requires SG 2.11.6 or later)
	initialUserInput	String	Optional, used to indicate a string being modified by a user.
	show	Boolean	Set to true when message is shown. Set to false when message should not be

			displayed
	textForItems	Array of strings	The text to show on the buttons. 0 to 5 items are allowed.
	localExit	Boolean	Deprecated - Indicates whether to show or hide the "X" on the message.
Busy	object		Indicates whether or not the player is busy and can accept any actions at the moment.
parameters	*name*	*type*	*Description*
	on	boolean	Whether the busy condition is active or not.
	timeoutSec	number	Time in seconds by when the player should definitely be ready again. If a B**usy on=false** has not been sent by this timeout, an error has likely occurred.
Rating	object		Indicates the rating system and current rated state
parameters	*name*	*type*	*Description*
	current	integer	Indicates whether current media has been rated. -1 – Disliked 0 – not rated 1 – Liked Note: Value can be any integer greater than -1. Note: This property holds values for both Star and Thumbs Up/Down actions. Only Star or Thumbs Up/Down should be specified in **ActionsSupported** - not both. Ex. If value is 1, and **Star** is specified as an **ActionsSupported,** the value will represent the a "one star" rating. Ex. If the value is -1 and **Thumbs Up/Down** is specified as an **ActionsSupported,** the value will represent a rating of disliked or thumbs down.

	max	integer	If Star is specified as an ActionsSupported, this property is valid. Values can range from 1-n
	system	integer	Indicates which actions to display. 1 – ThumbsUp /Thumbs Down 2 - Stars 3 – Hearts (Ex. Last.FM)

2.3 EVENTS

StateChanged	Sent to indicate that the state of the player has changed. Where possible, property changes should be grouped to minimize the number of StateChanged events.		
parameters	*Name*	*type*	*description*
	Could include one or more of the properties above	various	This is a mechanism by which the player user interface can learn of changes in the player in order to do such things as enable/disable controls, show new information to the user, etc. The player will decide which information to include here and when to send it. For example, when a track changes the player might send the new title and artist. The player user interface can of course read any of the player's properties whenever it sees fit. The player may indicate a "priority" of items by the order in which they appear, however the user interface is always free to ignore this order!

StatusMsgChanged	Sent to indicate a specific message to the user regarding the player. An example might be that the current stream will stop playing soon unless the user confirms that they are still listening.		
parameters	*Name*	*type*	*description*
	Text	string	The message to the user.
	timeoutSec	number	The recommended number of seconds that the message should be displayed for. The user interface may choose to ignore this number. 0 indicates to keep the message displayed indefinitely
	userInputRequired	string	Indicates type of user input required. Choices are **confirmation** (yes/no/cancel choices should be specified in textForItems),

			alphanumeric, numeric, **characterMasked**, **numericMasked,** and **none** (only the text field will be displayed, no buttons will show, requires SG 2.11.6 or later)
	initialUserInput	string	Optional, used to indicate a string being modified by a user.
	Show	boolean	Set to true when message is shown. Set to false when message should not be displayed
	textForItems	Array of strings	The text to show on the buttons. 0 to 5 items are allowed.
	localExit	boolean	Deprecated - Indicates whether to show or hide the "X" on the message.
BusyChanged	Sent to indicate that the player is occupied and cannot perform any actions at the moment.		
parameters	*name*	*type*	*description*
	on	boolean	Whether the busy condition has just started or just cleared.
	timeoutSec	number	Time in seconds by when the player should definitely be ready again. If a B**usy on=false** has not been sent by this timeout, an error has likely occurred.
StateChangedByBrowseContext	Sent by the player to indicate the some browse context did something which cause a change in the player state – probably someone picked a playable item in a list.		
parameters	*name*	*type*	*description*
	instance	number	The number of the browse context which caused the change. User interface might use this to flip to a now playing page, if this number matches its own instance number.

3 MENU OBJECT

This object encapsulates a set of lists which exist at multiple levels. This could be used to expose a menuing system built-in to a device, for example the media content on an iPod. This is essentially a restatement of the previously defined Dynamic List Control, however the definition here is protocol agnostic.

Except where noted otherwise, every action returns a **result** boolean indicating success or failure of the requested action.

3.1 ACTIONS

Reset	Resets the context of the list to the highest level (1).
Select	Chooses an item from a list, usually causing the current menu level to descend by one.

	An exception might be if the current level is already at the lowest level.		
input parameters	*name*	*type*	*description*
	item	number	Which item is being selected in the current list. 1-based.
PressAndHold	Indicates to the list that the user has pressed and held an item. The meaning of press-and-hold may be list-specific.		
input parameters	*name*	*type*	*description*
	item	number	Which item is being pressed and held in the current list. 1-based.
Back	Ascends one level in the current hierarchy. If already at the highest level, this has no effect.		
GetData	Retrieves multiple items at once. Returns an array of items. Each item could be an array of subitems on certain lists.		
input parameters	*name*	*type*	*description*
	item	number	First item being retrieved. 1-based
	count	number	How many items are being retrieved
return value	object		
	name	*type*	*description*
	L1	String	Text for line 1
	L2	String	Text for line 2
	URL	string	Text for image
	URLNAT	string	UserPAT - Text for image
Find	Aids in navigating the current list by returning the item number corresponding to the specified string, if any. The type of find performed is according to the **findDesired** property below.		
input parameters	query	string	Text which corresponds to the desired item.
Return value	Item	Number	Returned value is a dictionary consisting of a single item whose key is "item" and whose value is the index of the found item. -1 is returned if the item is not found.
Create	Used to start the station creation process.		
Search	Deprecated - Used to indicate Search intended		
PlayAll	Used to playall		
Edit	Deprecated - Used to enable editing the currently shown list		
QuickList	Used to quickly display a list of presets		
Advanced	Used to quickly display configuration options		
BackToTop	Ascends to the highest level in the current hierarchy. If already at the highest level, this has no effect.		
Favorites	Used to quickly display a list of favorite songs/stations.		
StatusMsgResponseMenu	User action on a status message displayed		
input parameters	*name*	*Type*	*Description*
	id	number	Which button was selected. Value can be from 1-5, or -1 if the status message was closed without

			using a button.
	userInput	string	Textual input from the user. Could be used to enter a stream provider-specific PIN number for example.
	localExit	boolean	True if statusmsg was locally dismissed on the client side. Ex. Either from timeout or cancel button or other means of dismissing message (touching outside message)

3.2 PROPERTIES

name	*type*	Access	*description*
Version	number	read-only	A numeric version for this object.
Language	string	read-write	Can be used to select the current desired language. Choices are: English, French, Spanish, Dutch, Russian, Chinese, Japanese
Instance	number	read-only	Numerical value uniquely identifying this list on the device where it resides. This could correspond to a browse context on devices which support multiple of those.
Title	string	read-only	Title of the current list.
Subtitle	string	read-only	Subtitle of the current list
ItemCnt	number	read-only	Indicates the number of items in the current list.
FindDesired	string	read-write	Defines the type of find which will be executed by the **Find** action. Default is **start**.
FindSupported	array of strings	read-only	Indicates which kinds of finds are permitted by the device. Multiple kinds may be supported on one device. **start**: matches beginning of item only **substring**: matches anywhere in item
FindAvailable	boolean	read-only	Whether or not the find action can be executed on the current list.
Level	number	read-only	Indicates the current depth in the hierarchy of lists. 1 is the highest level.
Sorted	string	read-only	Indicates how the current list is sorted. **none**: list is not sorted in any known order. **alphabetical**: list is sorted A-Z in ascending order.
MaxReqItems	number	read-only	The maximum number of items which may be requested at once.
ListSpecificFunctions	array of strings	read-only	A list of **extra functions** which are available for the current list. This could include things like PlayAll for media lists, possible sorting functions, or even jumping to specific menus such as

			creating a new station or searching. ONLY Functions: Create Find **P**layAll QuickList Advanced Favorites BackToTop
ActionsSupported			DEPRECATED, use ListSpecificFunctions instead. A list of optional functions that are supported. A player user interface could query this property and create buttons as needed. The base set of functions correspond to Quicklist. This could be used to accommodate future unknown functions from new streaming audio providers.
ActionsAvailable			DEPRECATED, use ListSpecificFunctions instead. A list of all of the supported functions which are currently available. Any strings listed here must have been defined in **actionsSupported**.
Busy	object		Indicates whether or not the menu is busy and can accept any actions at the moment.
parameters	*name*	*type*	*Description*
	on	boolean	Whether the busy condition is active or not. When true, the UI will delay certain property requests and methods until busy is false.
	timeoutSec	number	Time in seconds by when the menu should definitely be ready again. If a B**usy on=false** has not been sent by this timeout, an error has likely occurred.
EditMode	Boolean	Read-only	Indicates mode of editing True – List is editable False – List is not editable
StatusMsgMenu	Object	read-only	Values in this property will be used to show Status Messages to connecting clients that may have missed the StatusMsgMenuChanged event.
parameters	*name*	*type*	*Description*
	text	string	The message to the user.
	timeoutSec	Number	The recommended number of seconds

			that the message should be displayed for. The user interface may choose to ignore this number. 0 indicates to keep the message displayed indefinitely
	userInputRequired	String	Indicates type of user input required. Choices are **confirmation** (yes/no/cancel choices should be specified in textForItems), **alphanumeric, numeric**, **characterMasked**, **numericMasked,** and **none** (only the text field will be displayed, no buttons will show, requires SG 2.11.6 or later)
	initialUserInput	String	Optional, used to indicate a string being modified by a user.
	show	Boolean	Set to true when message is shown. Set to false when message should not be displayed
	textForItems	Array of strings	The text to show on the buttons. 0 to 5 items are allowed.
	localExit	Boolean	Deprecated - Indicates whether to show or hide the "X" on the message.
ListItemIcons	boolean	read-only	Whether or not the current list contains items with URLs.
UpdateCount	Number	Read-only	Incremented whenever Menu data is updated/changes Useful for when the client goes out of wifi range and returns. At this point the client will request all Media Player object properties and this one to sync up with the server. The client will compare this value to the one it last stored. If they are different, it will request a new menu.
IsMenuAvailable	Boolean	Read-only	Whether or not the Back action is available at that Level (depth in the hierarchy). Overrides the behavior where a level of 1 disables the Back action.

3.3 EVENTS

ListChanged	DO NOT USE – May be implemented in the future, not currently supported. Sent to indicate that the data for the specified range has changed.		
Parameters	*name*	*type*	*description*

	item	number	The first item which has changed.
	count	number	How many changed items there are, including the first one.
ClearChanged	Sent to indicate that the current list no longer exists. Use this event when modifying a list when the level does not change.		
BusyChanged	Indicates if the list control is capable of performing actions at the moment. It might not be able to if it is in the middle of retrieving data to satisfy a **GetData** request, for example.		
Parameters	*name*	*type*	*description*
	on	boolean	Whether the busy condition has just started or just cleared. When true, the UI will delay certain property requests and methods until busy is false.
	timeoutSec	number	Time in seconds by when the list should definitely be ready again. If a **Busy on=false** has not been sent by this timeout, an error has likely occurred.
StateChanged	Sent to indicate that the state of the list has changed.		Where possible, property changes should be grouped to minimize the number of StateChanged events.
Parameters			*description*
	Could include one or more of the properties above	various	This is a generic mechanism by which the player user interface can learn of changes in the list.
StatusMsgMenuChanged	Sent to indicate a specific message to the user regarding the player. An example might be that the current stream will stop playing soon unless the user confirms that they are still listening.		
parameters	*Name*	*type*	*description*
	Text	string	The message to the user.
	timeoutSec	number	The recommended number of seconds that the message should be displayed for. The user interface may choose to ignore this number. 0 indicates to keep the message displayed indefinitely
	userInputRequired	string	Indicates type of user input required. Choices are **confirmation** (yes/no/cancel choices should be specified in textForItems), **alphanumeric, numeric**, **characterMasked**, **numericMasked,** and **none** (only the text field will be displayed, no buttons will show, requires SG 2.11.6 or later)
	initialUserInput	string	Optional, used to indicate a string being modified by a user.
	Show	boolean	Set to true when message is shown. Set to false when message

			should not be displayed
	textForItems	Array of strings	The text to show on the buttons. 0 to 5 items are allowed.
	localExit	boolean	Deprecated - Indicates whether to show or hide the “X” on the message.