

Interface Documentation

HALws

Document Version 1.61 (3/24/2016)

© 2015-2016 Home Automated Living, Inc



Home
Automated Living™
automated living™
HOME

Toll Free (855) 442-5434
www.automatedliving.com

Table of Contents

1.0 Overview	3
1.1 Background	
1.2 Interface Summary	
1.3 Command Structure Overview	
1.4 Response Structure Overview	
1.5 HTTP Request/Response	
1.6 Special Characters	
1.7 Reserved Characters	
2.0 Device Area.....	5
2.1 DeviceService	
2.2 DeviceData	
3.0 Mode Area.....	9
3.1 ModeService	
3.2 ModeData	
4.0 Scene Area.....	11
4.1 SceneService	
4.2 SceneData	
5.0 Macro Area.....	13
5.1 MacroService	
5.2 MacroData	
6.0 Group Area	15
6.1 GroupService	
6.2 GroupData	
7.0 HVAC Area	17
7.1 HVACService	
7.2 HVACData	
8.0 Infrared Area.....	20
8.1 IrService	
8.2 IrData	
9.0 HAL Area.....	23
9.1 HALService	
10.0 Security Area	24
10.1 SecurityService	
10.2 SecurityData	
11.0 Telephone Area	28
11.1 TelephoneService	
11.2 TelephoneData	
11.3 VoiceMsg	
12.0 Digital Music Center Area	33
12.1 DMCSERVICE	
12.2 DMCDATA	
13.0 Speech Area.....	37
13.1 SpeechService	
13.2 SpeechData	
13.3 SpeechLastAudio	

1.0 Overview

This document is organized in a way that will make efficient use of your time and adequately convey the specifications for the **HALws** (Web Services) interface.

1.1 Background

Home Automated Living (HAL) produces voice controlled digital home systems integration software. With HAL software operating on a PC, you can control all the systems of your home, including lights, security, climate, telephone, home theatre, and the Internet. And you can control it all by voice – from anywhere in the home – or anywhere in the world. You can talk to the computer - to manipulate devices like thermostats and lights or to retrieve information like stock like stock quotes – in conversational speech.

Since many consumers are looking to expand the capabilities of the HAL software for specific purposes unique to their own circumstances, Home Automated Living developed **HALi** an ActiveX control to extend the events and methods within the HAL software. **HALi** required the custom interface to HAL be written as an application. Unlike HALi, HALws provides a means of retrieving the status and information about a device configured in HAL as well as a means of control through the use of a RESTful api that can be called from anywhere.

1.2 Interface Summary

HALws is broken into two types of interaction for each of the various areas of relevant control. Service type commands perform control functions whereas Data type commands return or modify data. The various Command Area that HALws currently provides functionality for are define below.

Command Areas

DeviceService	DeviceData
ModeService	ModeData
SceneService	SceneData
GroupService	GroupData
MacroService	MacroData
HVACService	HVACData
SecurityService	SecurityData
IrService	IrData
TelephoneService	TelephoneData
DMCService	DMCData
DVCService	DVCData
ShoppingService	ShoppingData
SpeechService	SpeehData
	ScheduleData
EMCService	EMCData
	RulesData
InternetService	InternetData
HALService	HALData

1.3 Command Structure Overview

A HALws http command starts with the IP address and port that is used to make connections to HomeNet. This address and port is configured in HAL System Settings, HomeNet. A token is also required as part of the command. This token is used to authorize permission to HALws and is generated in HAL System Settings, HomeNet. A HALws is composed of the type of Command Area followed by any required Parameters and is completed with the authorization Token. The Parameters can appear in any order and are prefaced with an exclamation "!". The authorization Token must be the last detail of the command and begins with a question mark "?". Spaces are permitted in the URL as are their hexadecimal representation %20 which you may see in the example commands in this documentation.

```
{Command Area}{!Parameter=value}{!Parameter=value}{?Token=value}
```

```
http://192.168.1.156:82/DeviceService!DeviceCmd=Set!DeviceName=Front%20Porch%20Light!DeviceAction=On!DeviceColor=ff75a1?Token=aj278kw72b4cd2s
```

1.4 Response Structure Overview

HALws responses will be in the form of JSON. Noted in this documentation for each Command Area are the possible responses for each type of call. Most responses will appear as follows.

```
{"Status":"OK", "DeviceCmd":"SETDEVICE", "DeviceName":"FRONT PORCH LIGHTS", "DeviceAction":"ON"}
```

1.5 HTTP Request/Response

HALws accepts requests following the HTTP 1.0 or 1.1 specifications. Requests should be encoded in UTF-8. Responses are also encoded in UTF-8.

1.6 Special Characters

Special Characters should be converted to hex to insure the request is processed correctly.

Forward Slash	/	%2F
Space		%20
Colon	:	%3A

1.7 Reserved Characters

Special Characters should be converted to hex to insure the request is processed correctly.

Exclamation Mark	!	%21
Question Mark	?	%3F

2.0 Device Area

2.1 DeviceService

Description Used to control or retrieve the state of a device

Method Get

Syntax /DeviceService

The parameters are:

Parameter	Description
!DeviceName	Name of Device to be referenced by method.
!DeviceCmd	Command to be performed.
!DevicePercent	A numeric value between 0 and 100. Required if using the "Dim" DeviceCmd.
!DeviceValue	A value/action to be assigned/performed to the device referenced by DeviceName.
!DeviceColor	A 6 Digit Hex value representing desire RGB Color.
?Token	Required string expression authorizing access HALws.

DeviceCmd Values

String	Description
"SetDevice"	Set a device to a state.
"SetFlag"	Set a flag to true or false.
"SetSensor"	Set a sensor to a value.
"SetCounter"	Set a counter to a value.
"SetTimer"	Set a timer to a value.
"SetVariable"	Set a variable to a value.
"Get"	Request the status of the device referenced by DeviceName.

DeviceValue Values

String	Description
"On"	The device referenced by DeviceName is to be set to On.
"Off"	The device referenced by DeviceName is to be set to Off.
"Dim"	The device referenced by DeviceName is to be Dimmed to DevicePercent.
"True"	The Flag referenced by DeviceName is to be set to True.
"False"	The Flag referenced by DeviceName is to be set to False.
"9999"	The Timer or Counter referenced by DeviceName is to be set to this value.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing DeviceCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid DeviceCmd"}</pre>
<pre>{"Status":"Error", "Description":"Missing DeviceAction"}</pre>
<pre>{"Status":"Error", "Description":"Invalid DeviceAction"}</pre>
<pre>{"Status":"Error", "Description":"Missing DevicePercent"}</pre>
<pre>{"Status":"Error", "Description":"Invalid DevicePercent"}</pre>
<pre>{"Status":"Error", "Description":"Invalid DeviceColor"}</pre>
<pre>{"Status":"Error", "Description":"Missing DeviceName"}</pre>
<pre>{"Status":"Error", "Description":"HAL Server is not loaded"}</pre>
<pre>{"Status":"OK", "DeviceCmd":"<devicecmd>", "DeviceName":"<devicename>", "DeviceValue":"<devicevalue>"}</pre>

Example Commands and Responses

Set a device to On.

```
/DeviceService!DeviceName=Bar%20Lights!DeviceCmd=SetDevice!DeviceValue=On?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SET", "DeviceName":"BAR LIGHTS", "DeviceValue":"On"}
```

Set a device to On and Set the Color.

```
/DeviceService!DeviceName=Basement%20Lights!DeviceCmd=SetDevice!DeviceValue=On!DeviceColor=ff7591?Token=alj278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SETDEVICE", "DeviceName":"BASEMENT LIGHTS", "DeviceValue":"On", "DeviceColor":"FF7591"}
```

Set a device to a Dim percentage.

```
/DeviceService!DeviceName=Livingroom%20Lights!DeviceCmd=SetDevice!DeviceValue=Dim!DevicePercent=75?Token=alj278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SETDEVICE", "DeviceName":"LIVINGROOM LIGHTS", "DeviceValue":"Dim", "DevicePercent":"75"}
```

Set a device to Off.

```
/DeviceService!DeviceName=Bar%20Lights!DeviceCmd=SetDevice!DeviceValue=Off?Token=alj278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SETDEVICE", "DeviceName":"BAR LIGHTS", "DeviceValue":"Off"}
```

Set a HAL flag to True.

```
/DeviceService!DeviceName=TTS%20Door%20Announce!DeviceCmd=Set!DeviceValue=True?Token=alj278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SETFLAG", "DeviceName":"TTS DOOR ANNOUNCE", "DeviceValue":"TRUE"}
```

Set a HALtimer to a value.

```
/DeviceService!DeviceName=Bar%20Motion%20Timer!DeviceCmd=SetTimer!DeviceValue=300?Token=alj278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SETTIMER", "DeviceName":"BAR MOTION TIMER", "DeviceValue":"300"}
```

Set a HALcounter to a value.

```
/DeviceService!DeviceName=Driveway%20Activity%20Counter!DeviceCmd=SetCounter!DeviceValue=28?Token=alj278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SETCOUNTER", "DeviceName":"DRIVEWAY ACTIVITY COUNTER", "DeviceValue":"28"}
```

Set a HALvariable to a value.

```
/DeviceService!DeviceName=Color%20Variable!DeviceCmd=SetVariable!DeviceValue=Blue?Token=alj278kw72b4cd2s
```

```
{"Status":"OK", "DeviceCmd":"SETVARIABLE", "DeviceName":"COLOR VARIABLE", "DeviceValue":"BLUE"}
```

Get the status of a device.

```
/DeviceService!DeviceName=Front%20Porch%20Lights!DeviceCmd=Get?Token=alj278kw72b4cd2s
```

```
{"Status":"ON", "DeviceName":"FRONT PORCH LIGHTS"}
```

Get the status of a flag.

```
/DeviceService!DeviceName=TTS%20Door%20Announce!DeviceCmd=Get?Token=alj278kw72b4cd2s
```

```
{"Status":"TRUE", "DeviceName":"TTS DOOR ANNOUNCE"}
```

Get the status of a timer.

```
/DeviceService!DeviceName=Bar%20Motion%20Timer!DeviceCmd=Get?Token=alj278kw72b4cd2s
```

```
{"Status":"187", "DeviceName":"BAR MOTION TIMER"}
```

2.2 DeviceData

Description Used to retrieve device data

Method Get

Syntax /DeviceData

The parameters are:

Parameter	Description
!DeviceName	Name of Device to be referenced by method.
!DeviceCmd	Command to be performed.
!DeviceType	Filter to be used when retrieving a list of elements.
!DeviceValue	A numeric value. Required when using the "Set" DeviceCmd for Timers.
!DeviceColor	A 6 Digit Hex value representing desire RGB Color.
?Token	Required string expression authorizing access HALws.

DeviceCmd Values

String	Description
"GetNames"	Retrieves a list of elements.
"GetDetails"	Retrieves fields related to the device referenced by DeviceName.

DeviceType Values

String	Description
"All"	Retrieves a list of all elements.
"Appl"	Retrieves a list of Appliance elements.
"Light"	Retrieves a list of Lighting elements.
"Sensor"	Retrieves a list of Sensor elements.
"Video"	Retrieves a list of Video elements.
"Custom"	Retrieves a list of Custom elements.
"Theatre"	Retrieves a list of Theater / IR elements.

Response Values

Responses
<pre>{"Status":"Error", "Description":"Missing DeviceCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid DeviceCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid DeviceType"}</pre>
<pre>{"Status":"Error", "Description":"Missing DeviceName"}</pre>
<pre>{"Status":"Error", "Description":"HAL Server is not loaded"}</pre>

Example Commands and Responses

Get the names of all appliances.

```
/DeviceData!DeviceCmd=GetNames!DeviceType=Appl?Token=a1j278kw72b4cd2s
{"DeviceElements": [ {"DeviceName":"BOATHOUSE BOAT HEATER"},
{"DeviceName":"HOTWATER HEATER"}, {"DeviceName":"LIVINGROOM LEFT
RECPTICAL"}, {"DeviceName":"LIVINGROOM RIGHT RECPTICAL"},
{"DeviceName":"MASTER BEDROOM MATTRESS WARMER"} ] }
```

Get the name of all Theatre devices.

```
/DeviceData!DeviceCmd=GetNames!DeviceType=Theatre?Token=a1j278kw72b4cd2s
{"DeviceElements": [ {"DeviceName":"BASEMENT DIRECT TV"},
{"DeviceName":"BASEMENT TV"}, {"DeviceName":"FAMILYROOM HD DVR"},
{"DeviceName":"FAMILYROOM PLAYSTATION"}, {"DeviceName":"FAMILYROOM
STEREO"}, {"DeviceName":"FAMILYROOM TV"}, {"DeviceName":"MASTER BEDROOM
TV"}, {"DeviceName":"MASTER BEDROOM DVD"} ] }
```

Get the name of all sensors.

```
/DeviceData!DeviceCmd=GetNames!DeviceType=Sensor?Token=a1j278kw72b4cd2s
{"DeviceElements": [ {"DeviceName":"BACKDECK MOTION SENSOR"},
{"DeviceName":"BACKYARD MOTION SENSOR"}, {"DeviceName":"BASEMENT BAR
MOTION SENSOR"}, {"DeviceName":"BASEMENT DECK MOTION SENSOR"},
{"DeviceName":"BASEMENT DOOR SENSOR"}, {"DeviceName":"BASEMENT FIREPLACE
MOTION SENSOR"}, {"DeviceName":"BASEMENT OFFICE MOTION DETECTOR"},
{"DeviceName":"BASEMENT SMOKE DETECTOR"}, {"DeviceName":"DINING ROOM
MOTION SENSOR"}, {"DeviceName":"DRIVEWAY ENTERANCE MOTION SENSOR"},
{"DeviceName":"DRIVEWAY EXIT MOTION SENSOR"}, {"DeviceName":"FRONT DOOR
SENSOR"}, {"DeviceName":"FRONT SIDEWALK MOTION SENSOR"},
{"DeviceName":"FRONT YARD MOTION SENSOR"}, {"DeviceName":"GARAGE DOOR
SENSOR"}, {"DeviceName":"GARAGE MOTION SENSOR"}, {"DeviceName":"KITCHEN
MOTION SENSOR"}, {"DeviceName":"LIVINGROOM MOTION SENSOR"},
{"DeviceName":"UPSTAIRS MOTION SENSOR"}, {"DeviceName":"UPSTAIRS SMOKE
DETECTOR"}, {"DeviceName":"WATER TANK MOISTURE SENSOR"} ] }
```

Get the name of all lights.

```
/DeviceData!DeviceCmd=GetNames!DeviceType=Light?Token= a1j278kw72b4cd2s
{"DeviceElements": [ {"DeviceName":"BACK SIDEWALK LIGHTS"},
{"DeviceName":"BACK YARD LIGHTS"}, {"DeviceName":"BAR LIGHTS"},
{"DeviceName":"BASEMENT BATHROOM LIGHTS"}, {"DeviceName":"BASEMENT DOOR
LIGHT"}, {"DeviceName":"BASEMENT FIREPLACE LIGHTS"},
{"DeviceName":"BASEMENT OFFICE LIGHTS"}, {"DeviceName":"BASEMENT PORCH
LIGHTS"}, {"DeviceName":"BASEMENT STEP LIGHTS"}, {"DeviceName":"BOATHOUSE
LIGHTS"}, {"DeviceName":"DINING ROOM LIGHTS"}, {"DeviceName":"DRIVEWAY
LIGHTS"}, {"DeviceName":"FAMILYROOM FIREPLACE LIGHTS"},
{"DeviceName":"FOYER LIGHTS"}, {"DeviceName":"FRONT PORCH LIGHTS"},
{"DeviceName":"GARAGE LIGHTS"}, {"DeviceName":"KITCHEN LIGHTS"},
{"DeviceName":"KITCHEN SINK LIGHTS"}, {"DeviceName":"LIVINGROOM FIREPLACE
LIGHTS"}, {"DeviceName":"LIVINGROOM LIGHTS"}, {"DeviceName":"MASTER
BATHROOM LIGHTS"}, {"DeviceName":"MASTER BEDROOM LIGHTS"},
{"DeviceName":"MASTER BEDROOM DRESSER LAMP"}, {"DeviceName":"MASTER
BEDROOM SCONCE"}, {"DeviceName":"MASTER BEDROOM WINDOW LAMP"},
{"DeviceName":"SIDE YARD LIGHTS"}, {"DeviceName":"STAIRWAY LIGHTS"},
{"DeviceName":"UPSTAIRS GUESTROOM LIGHTS"}, {"DeviceName":"UPSTAIRS
HALLWAY LIGHTS"} ] }
```


3.0 Mode Area

3.1 ModeService

Description Used to set or get the current house Mode

Method Get

Syntax /ModeService

The parameters are:

Parameter	Description
!ModeName	Name of Mode to be referenced by method.
?Token	Required string expression authorizing access HALws.

ModeCmd Values

String	Description
"Set"	The mode referenced by ModeName is set to be the current Mode.
"Get"	Request the currently set Mode.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing ModeCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid ModeCmd"}</pre>
<pre>{"Status":"Error", "Description":"Missing ModeName"}</pre>
<pre>{"Status":"Error", "Description":"Mode not found", "ModeName":"<name>"}</pre>

Example Commands and Responses

Set the house mode.

```
/ModeService!ModeCmd=Set!ModeName=Away?Token=a1j278kw72b4cd2s
{"Status":"OK", "ModeCmd":"SET", "ModeName":"AWAY"}
```

Get the current house mode.

```
/ModeService!ModeCmd=Get?Token=a1j278kw72b4cd2s
{"ModeName":"AWAY", "ModeCmd":"GET"}
```

3.2 ModeData

Description Used to retrieve Mode data

Method Get

Syntax /ModeData

The parameters are:

Parameter	Description
!ModeCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

ModeCmd Values

String	Description
"GetNames"	Retrieves a list of elements.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing ModeCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid ModeCmd"}</pre>

Example Commands and Responses

Get the name of all house modes.

```
/ModeData!ModeCmd=GetNames!Token=a1j278kw72b4cd2s
```

```
{"ModeElements": [ {"ModeName":"AWAY"}, {"ModeName":"DAY"},
{"ModeName":"ENTERTAINING"}, {"ModeName":"JUST ME"}, {"ModeName":"NIGHT"},
{"ModeName":"NORMAL"} ] }
```

4.0 Scene Area

4.1 SceneService

Description Used to set a Scene

Method Get

Syntax /SceneService

The parameters are:

Parameter	Description
!SceneName	Name of Scene to be referenced by method.
?Token	Required string expression authorizing access HALws.

SceneCmd Values

String	Description
"Set"	The Scene referenced by SceneName is set to be the current Scene.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing SceneCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid SceneCmd"}</pre>
<pre>{"Status":"Error", "Description":"Missing SceneName"}</pre>
<pre>{"Status":"Error", "Description":"Scene not found", "SceneName":"<name>"}</pre>

Example Commands and Responses

Set a scene.

```
/SceneService!SceneCmd=Set!SceneName=Dinner%20Scene?Token=a1j278kw72b4cd2s
{"Status":"OK", "SceneCmd":"SET", "SceneName":"DINNER SCENE"}
```

4.2 SceneData

Description Used to retrieve Scene data

Method Get

Syntax /SceneData

The parameters are:

Parameter	Description
!SceneCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

SceneCmd Values

String	Description
"GetNames"	Retrieves a list of elements.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing SceneCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid SceneCmd"}</pre>

Example Commands and Responses

Get the name of all scenes.

```
/SceneData!SceneCmd=GetNames!Token=a1j278kw72b4cd2s
```

```
{"SceneElements": [ {"SceneName":"DINNER SCENE"}, {"SceneName":"PARTY SCENE"}, {"SceneName":"MOVIE SCENE"} ] }
```

5.0 Macro Area

5.1 MacroService

Description Used to set a Macro

Method Get

Syntax /MacroService

The parameters are:

Parameter	Description
!MacroName	Name of Macro to be referenced by method.
?Token	Required string expression authorizing access HALws.

MacroCmd Values

String	Description
"Set"	The Macro referenced by MacroName is executed.

Response Values

Response
<pre>{"Status": "Error", "Description": "Missing MacroCmd"}</pre>
<pre>{"Status": "Error", "Description": "Invalid MacroCmd"}</pre>
<pre>{"Status": "Error", "Description": "Missing MacroName"}</pre>
<pre>{"Status": "Error", "Description": "Macro not found", "MacroName": "<name>"}</pre>

Example Commands and Responses

Set a macro to run.

```
/MacroService!MacroCmd=Set!MacroName=Going%20To%20Bed?Token=a1j278kw72b4cd
2s
```

```
{"Status": "OK", "MacroCmd": "SET", "MacroName": "GOING TO BED"}
```

5.2 MacroData

Description Used to retrieve Macro data

Method Get

Syntax /MacroData

The parameters are:

Parameter	Description
!MacroCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

MacroCmd Values

String	Description
"GetNames"	Retrieves a list of elements.

Response Values

Response
<code>{"Status":"Error", "Description":"Missing MacroCmd"}</code>
<code>{"Status":"Error", "Description":"Invalid MacroCmd"}</code>

Example Commands and Responses

Get the name of all macros.

```
/MacroData!MacroCmd=GetNames!Token=a1j278kw72b4cd2s
```

```
{
  "MacroElements": [
    {"MacroName":"ADJUST THE BED"},
    {"MacroName":"CLOSE GARAGE DOOR"},
    {"MacroName":"FAMILYLYROOM TV ABC"},
    {"MacroName":"FAMILYROOM TV CBS"},
    {"MacroName":"FAMILYROOM TV ESPN"},
    {"MacroName":"FAMILYROOM TV OFF"},
    {"MacroName":"FAMILYROOM TV ON"},
    {"MacroName":"FAMILYROOM TV VOLUME DOWN"},
    {"MacroName":"FAMILYROOM TV VOLUME UP"},
    {"MacroName":"GOING TO BED"},
    {"MacroName":"HEAD ALL THE WAY DOWN"},
    {"MacroName":"HEAD ALL THE WAY UP"},
    {"MacroName":"HEAD DOWN"},
    {"MacroName":"HEAD DOWN MORE"},
    {"MacroName":"HEAD UP"},
    {"MacroName":"HEAD UP MORE"},
    {"MacroName":"OPEN BEDROOM DOOR"},
    {"MacroName":"OPEN GARAGE DOOR"},
    {"MacroName":"RESET THE BED"}
  ]
}
```

6.0 Group Area

6.1 GroupService

Description Used to set a Group of devices to a state.

Method Get

Syntax /GroupService

The parameters are:

Parameter	Description
!GroupCmd	Command to be performed.
!GroupName	Name of Group to be referenced by method.
?Token	Required string expression authorizing access HALws.

GroupCmd Values

String	Description
"On"	The Group referenced by GroupName is set to be set to On.
"Off"	The Group referenced by GroupName is set to be set to Off.

Response Values

Response
<pre>{"Status": "Error", "Description": "Missing GroupCmd"}</pre>
<pre>{"Status": "Error", "Description": "Invalid GroupCmd"}</pre>
<pre>{"Status": "Error", "Description": "Missing GroupName"}</pre>
<pre>{"Status": "Error", "Description": "Group not found", "GroupName": "<name>"}</pre>

Example Commands and Responses

Set a group of devices to a state.

```
/GroupService!GroupCmd=On!GroupName=Outside?Token=a1j278kw72b4cd2s
{"Status": "OK", "GroupCmd": "OFF", "GroupName": "OUTSIDE"}
```

6.2 GroupData

Description Used to retrieve Group data

Method Get

Syntax /GroupData

The parameters are:

Parameter	Description
!GroupCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

GroupCmd Values

String	Description
"GetNames"	Retrieves a list of elements.

Response Values

Response
<code>{"Status":"Error", "Description":"Missing GroupCmd"}</code>
<code>{"Status":"Error", "Description":"Invalid GroupCmd"}</code>

Example Commands and Responses

Get the name of all groups.

```
/GroupData!GroupCmd=GetNames!Token=a1j278kw72b4cd2s
```

```
{
  "GroupElements": [
    {"GroupName": "APPLIANCES"}, {"GroupName": "BACK"},
    {"GroupName": "BACK YARD LIGHTS"}, {"GroupName": "BASEMENT"},
    {"GroupName": "CHRISTMAS LIGHTS"}, {"GroupName": "DOCK LIGHTS"},
    {"GroupName": "FRONT"}, {"GroupName": "INSIDE"}, {"GroupName": "KITCHEN LIGHTS"},
    {"GroupName": "LOFT AREA"}, {"GroupName": "MAIN FLOOR"},
    {"GroupName": "MASTER BEDROOM"}, {"GroupName": "OUTSIDE"},
    {"GroupName": "SIDE YARD"}, {"GroupName": "UPSTAIRS"} ] }

```


7.0 HVAC Area

7.1 HVACService

Description Used to control or retrieve the state of a thermostat.

Method Get

Syntax /HVACService

The parameters are:

Parameter	Description
!HVACName	Name of HVAC to be referenced by method.
!HVACZone	Zone of HVAC to be referenced by method.
!HeatSpValue	Heating Setpoint as a numeric value between 0 and 99
!CoolSpValue	Cooling Setpoint as a numeric value between 0 and 99
!FanMode	Fan mode as a string value defined below.
!HVACMode	HVAC mode as a string value defined below.
!HVACCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

FanMode Values

String	Description
"On"	The HVAC referenced by FanMode is to be set to On.
"Auto"	The HVAC referenced by FanMode is to be set to Auto.

HVACModeValues

String	Description
"Cool"	The HVAC referenced by HVACName is to be set to On.
"Heat"	The HVAC referenced by HVACName is to be set to Off.
"Auto"	The HVAC referenced by HVACName is to be Dimmed.
"Off"	The HVAC referenced by HVACName is to be set to True.
"EHeat"	The HVAC referenced by HVACName is to be set to False.

HVACModeValues

String	Description
"Set"	The HVAC referenced byHVACName is to be set to On.
"Get"	Request the status of the HVAC referenced by HVACName.

Response Values

Response
<code>{"Status":"Error", "Description":"Missing HVACName"}</code>
<code>{"Status":"Error", "Description":"Invalid HeatSpValue"}</code>
<code>{"Status":"Error", "Description":"Invalid CoolSpValue"}</code>
<code>{"Status":"Error", "Description":"Invalid FanMode"}</code>
<code>{"Status":"Error", "Description":"Invalid HVACMode"}</code>
<code>{"Status":"OK"}</code>
<code>{"Status":"Error", "Decription":"No thermostat data returned", "HVACCmd":"GET", "HVACName":"MAIN THERMOSTAT", "HVACZone":"2"}</code>

Example Commands and Responses

Set the heating set point, fan mode and thermostat mode of a zone.

```
/HVACService!HVACCmd=Set!HVACzone=2!HeatSpValue=64!FanMode=Auto!HVACMode=Heat?Token=a1j278kw72b4cd2s
```

```
{"Status":"Ok", "HVACCmd":"SET", "HVACName":"MAIN THERMOSTAT", "HVACZone":"2", "HeatSpValue":"64", "FanMode":"Auto", "HVACMode":"HEAT"}
```

Set the cooling set point of a zone by thermostat name.

```
/HVACService!HVACCmd=Set!HVACName=UPSTAIRS%20THERMOSTAT~1!CoolSpValue=71?Token=a1j278kw72b4cd2s
```

```
{"Status":"Ok", "HVACCmd":"SET", "HVACName":"UPSTAIRS THERMOSTAT",  
"HVACZone":"1", "CoolSpValue":"71"}
```

Get the HVAC status for a zone.

```
/HVACService!HVACCmd=GetStatus!HVACzone=2?Token=a1j278kw72b4cd2s
```

```
{"Status":"Ok", "HVACCmd":"GETSTATUS", "HVACName":"MAIN THERMOSTAT",  
"HVACZone":"2", "eTemp":"71", "HeatTemp":"64", "CoolTemp":"84",  
"HvacMode":"Heating", "Fan":"Fan Auto"}
```

7.2 HVACData

Description Used to retrieve HVAC data

Method Get

Syntax /HVACData

The parameters are:

Parameter	Description
!HVACCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

HVACCmd Values

String	Description
"GetNames"	Retrieves a list of elements.
"GetZones"	Retrieves a list of elements.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing HVACCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid HVACCmd"}</pre>

Example Commands and Responses

Get the names of all thermostats.

```
/HVACData!HVACCmd=GetNames!Token=a1j278kw72b4cd2s
```

```
{"HVACElements": [ {"HVACName":"UPSTAIR THERMOSTAT"}, {"HVACName":"MAIN FLOOR THERMOSTAT"}, {"HVACName":"BASEMENT THERMOSTAT"} ] }
```

Get the zones of all thermostats.

```
/HVACData!HVACCmd=GetZones!Token=a1j278kw72b4cd2s
```

```
{"HVACElements": [ {"HVACStat": [ {"HVACName":"UPSTAIR THERMOSTAT", "HVACZone":"1" } ] }, {"HVACStat": [ {"HVACName":"MAIN FLOOR THERMOSTAT", "HVACZone":"2" } ] }, {"HVACStat": [ {"HVACName":"BASEMENT THERMOSTAT", "HVACZone":"3" } ] } ] }
```

8.0 Infrared Area

8.1 IrService

Description Used to control or retrieve the state of a device

Method Get

Syntax /IrService

The parameters are:

Parameter	Description
!IrCmd	Command to be performed.
!IrDevice	A string value specifying the device to be controlled.
!IrButton	A string specifying the button to be sent.
!IrRepeat	A numeric value to specify how many times to send the IrButton.
?Token	Required string expression authorizing access HALws.

IrCmd Values

String	Description
"Set"	Request an action to be performed by !DeviceAction.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing IrCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid IrCmd"}</pre>
<pre>{"Status":"Error", "Description":"Missing IrDevice"}</pre>
<pre>{"Status":"Error", "Description":"Missing IrButton"}</pre>
<pre>{"Status":"Error", "Description":"Invalid IrRepeat"}</pre>

Example Commands and Responses

Set the IR device button.

```
/IrService!IrCmd=Set!IrDevice=BASEMENT%20DVR!IrButton=LAST%20CHANNEL?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "IrCmd":"SET", "IrDevice":"BASEMENT DVR", "IRButton":"LAST CHANNEL"}
```

Set the IR device button to repeat X times.

```
/IrService!IrCmd=Set!IrDevice=BASEMENT%20TV!IrButton=VOLUME%20DOWN!IrRepeat=3?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "IrCmd":"SET", "IrDevice":"BASEMENT TV", "IRButton":"VOLUME DOWN", "IrRepeat":"3"}
```

8.2 IrData

Description Used to retrieve Ir data

Method Get

Syntax /IrData

The parameters are:

Parameter	Description
!IrName	Name of Ir to be referenced by method.
!IrCmd	Command to be performed.
!IrType	Filter to be used when retrieving a list of elements.
!IrValue	A numeric value. Required when using the "Set" IrCmd for Timers.
!IrColor	A 6 Digit Hex value representing desire RGB Color.
?Token	Required string expression authorizing access HALws.

IrCmd Values

String	Description
"GetNames"	Retrieves a list of elements.
"GetDetails"	Retrieves fields related to the Ir referenced by IrDevice.

DeviceType Values

String	Description
"All"	Retrieves a list of all elements.
"Appl"	Retrieves a list of Appliance elements.
"Light"	Retrieves a list of Lighting elements.
"Sensor"	Retrieves a list of Sensor elements.
"Video"	Retrieves a list of Video elements.
"Custom"	Retrieves a list of Custom elements.
"Theatre"	Retrieves a list of Theater / IR elements.

Response Values

Responses
<pre>{"Status":"Error", "Description":"Missing IrCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid IrCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid IrType"}</pre>
<pre>{"Status":"Error", "Description":"Missing IrName"}</pre>

Example Commands and Responses

Get the IR all devices.

```
/IrData!IRCmd=GetNames?Token=a1j278kw72b4cd2s
```

```
{"IrElements": [ {"IrName":"BASEMENT DIRECT TV"}, {"IrName":"BASEMENT DVR"}, {"IrName":"BASEMENT TV"}, {"IrName":"BEDROOM TV TWO"}, {"IrName":"FAMILYROOM HD DVR"}, {"IrName":"FAMILYROOM PLAYSTATION"}, {"IrName":"FAMILYROOM STEREO"}, {"IrName":"FAMILYROOM TV"}, {"IrName":"LIVINGROOM TV"}, {"IrName":"MASTER BEDROOM DVD"} ] }
```

Get the IR buttons for a device.

```
/IrData!IRCmd=GetButtons!IrDevice=BASEMENT DVR?Token= a1j278kw72b4cd2s
```

```
{"IrButtons": [ {"IrButton":"ADJUST UP"}, {"IrButton":"AUX, RGB"}, {"IrButton":"AV, TUNER"}, {"IrButton":"BACK"}, {"IrButton":"CHANNEL DOWN"}, {"IrButton":"CHANNEL ENTER"}, {"IrButton":"CHANNEL UP"}, {"IrButton":"DIGIT 0"}, {"IrButton":"DIGIT 1"}, {"IrButton":"DIGIT 2"}, {"IrButton":"DIGIT 3"}, {"IrButton":"DIGIT 4"}, {"IrButton":"DIGIT 5"}, {"IrButton":"DIGIT 6"}, {"IrButton":"DIGIT 7"}, {"IrButton":"DIGIT 8"}, {"IrButton":"DIGIT 9"}, {"IrButton":"EXIT, CANCEL"}, {"IrButton":"EXTERNAL ANTENNA"}, {"IrButton":"FAST FORWARD"}, {"IrButton":"FAVORITE"}, {"IrButton":"GUIDE"}, {"IrButton":"LAST CHANNEL"}, {"IrButton":"MENU"} ] }
```

```
(AUDIO)"}, {"IrButton":"MENU DOWN, ADJUST DOWN"}, {"IrButton":"MENU LEFT"}, {"IrButton":"MENU RIGHT"}, {"IrButton":"MENU SELECT"}, {"IrButton":"MENU UP, ADJUST UP"}, {"IrButton":"MUTE"}, {"IrButton":"OFF, POWER ON/OFF"}, {"IrButton":"ON, POWER ON/OFF"}, {"IrButton":"PAUSE"}, {"IrButton":"PLAY"}, {"IrButton":"POWER ON/OFF"}, {"IrButton":"RECORD"}, {"IrButton":"REWIND"}, {"IrButton":"SAP, CC"}, {"IrButton":"SLOW"}, {"IrButton":"STOP"}, {"IrButton":"SURROUND ON/OFF"}, {"IrButton":"TV, COMPONENT"}, {"IrButton":"TV/VIDEO"}, {"IrButton":"VOLUME DOWN"}, {"IrButton":"VOLUME UP"} ] }
```

9.0 HAL Area

9.1 HALService

Description Used to perform various actions or retrieve system information.

Method Get

Syntax /HALService

The parameters are:

Parameter	Description
!HALCmd	Command to be performed.
!HALAction	A string value specifying the device to be controlled.
!HALParam	A string value specifying the data to be used by the HALAction.
?Token	Required string expression authorizing access HALws.

HALCmd Values

String	Description
"Set"	Request an action to be performed by !HALAction.
"GetSummary"	Get the general status of the various interfaces.

HALCmd Values

String	Description
"TTS"	Send string as defined by HALParam to Text To Speech engine.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing HALCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid HALCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid HALAction"}</pre>
<pre>{"Status":"Error", "Description":"Missing HALParam"}</pre>

Example Commands and Responses

Set the TTS text to be spoken.

```
/HALservice!HALCmd=Set!HALAction=TTS!HALparam=Hello%20Everyone,%20How%20are%20you?Token= a1j278kw72b4cd2s
```

```
{"Status":"OK", "HALCmd":"SET", "HALAction":"TTS"}
```

Get the a system summary.

```
/HALService!HALCmd=GetSummary?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "HALCmd":"GETSUMMARY", "Sunrise":"7:22:00 AM",
"Sunset":"7:12:00 PM", "CurrentMode":"Day", "TimeModeSet":"3/12/2016
7:22:36 PM", "CurrentTemp":"71F", "SecurityStatus":"Disarmed",
"NewVoiceMsgs":"0", "LastCall":"01/28/16 7:24 PM, Chesapeake VA, (888)
555-1212", "KwNow":"1.224", "SpeechIn":"Turn The Basement Fireplace Lights
Off", "SpeechOut":"Turn The Basement Fireplace Lights Off",
"LastSensorEvent":"Front Door Motion Sensor", "LastDeviceEvent":"Basement
Fireplace Lights Off", "LastEvent":"Listen State Changed: On:listen:mic"}
```

10.0 Security Area

10.1 SecurityService

Description Used to control or retrieve the state of the security system.

Method Get

Syntax /SecurityService

The parameters are:

Parameter	Description
!SecurityZone	A numeric value between 0 and 256
!SecurityMode	Security mode as a string value defined below.
!SecurityCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

SecurityCmd Values

String	Description
"Set"	The action referenced by SecurityAction is to be set.
"GetStatus"	Retrieve the status of the security system.
"Bypass"	Not implemented yet. SecurityZone= 0 represents all zones.
"Restore"	Not implemented yet. SecurityZone= 0 represents all zones.

SecurityModeValues

String	Description
"Day"	The Security System mode is to be set to Day.
"Night"	The Security System mode is to be set to Night.
"Away"	The Security System mode is to be set to Away.
"Vacation"	The Security System mode is to be set to Vacation.
"Day Instant"	The Security System mode is to be set to Day Instant.
"Night Delay"	The Security System mode is to be set to Night Instant.
"Stay"	The Security System mode is to be set to Stay.
"Home"	The Security System mode is to be set to Home.
"Night Instant"	The Security System mode is to be set to Night Instant.
"Stay Instant"	The Security System mode is to be set to Stay Instant.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing SecurityCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid SecurityCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid SecurityAction"}</pre>
<pre>{"Status":"Error", "Description":"Missing SecurityMode"}</pre>
<pre>{"Status":"Error", "Description":"Invalid SecurityMode"}</pre>
<pre>{"Status":"Error", "Description":"No thermostat data returned", "SecurityCmd":"GET", "SecurityName":"MAIN THERMOSTAT", "SecurityZone":"2"}</pre>

Example Commands and Responses

Arm the security system in Away mode.

```
/SecurityService!SecurityCmd=Set!SecurityAction=Arm!SecurityMode=Away?Token=a1j278kw72b4cd2s
```

```
{"Status":"Ok", "SecurityCmd":"SET", "SecurityAction":"ARM", "SecurityMode":"AWAY"}
```

Get the status of the security system.

```
/SecurityService!SecurityCmd=GetStatus?Token=a1j278kw72b4cd2s
```

```
{"Status":"DISARMED and NOT READY", "SecurityCmd":"GETSTATUS"}
```


10.2 SecurityData

Description Used to retrieve Security data

Method Get

Syntax /SecurityData

The parameters are:

Parameter	Description
!SecurityCmd	Command to be performed.
!SecurityZone	A numeric value between 0 and 256.
?Token	Required string expression authorizing access HALws.

SecurityCmd Values

String	Description
"GetNames"	Retrieves a list of zone ids and names.
"GetZones"	Retrieves a list of zone ids, names and status.
"GetZone"	Retrieves status a zone specified by SecurityZone.

Response Values

Response
<pre>{"Status":"Error", "Description":"Missing SecurityCmd"}</pre>
<pre>{"Status":"Error", "Description":"Invalid SecurityCmd"}</pre>

Example Commands and Responses

Get the status of a security zone.

```
/SecurityData!SecurityCmd=GetZone!SecurityZone=32!Token=alj278kw72b4cd2s
{"SecurityZone": [ {"SecurityZoneId":"32", "SecurityZoneName":"BASEMENT
BAR MOTION SENSOR", "SecurityZoneStatus":"CLOSED"} ] }
```

Get the status of all security zones.

```
/SecurityData!SecurityCmd=GetZones!Token=alj278kw72b4cd2s
{"SecurityZoneStatus": [ {"SecurityZoneId":"1", "SecurityZoneName":"NONE",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"2",
"SecurityZoneName":"BOATHOUSE DOOR SENSOR",
"SecurityZoneStatus":"BYPASSED"}, {"SecurityZoneId":"3",
"SecurityZoneName":"BOATHOUSE GARAGE DOOR SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"4",
"SecurityZoneName":"BOATHOUSE MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"5",
"SecurityZoneName":"HVAC WATER SENSOR", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"6", "SecurityZoneName":"BASEMENT OFFICE MOTION
DETECTOR", "SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"7",
"SecurityZoneName":"NONE", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"8", "SecurityZoneName":"NONE",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"9",
"SecurityZoneName":"NONE", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"10", "SecurityZoneName":"MISCELLANEOUS DOOR SENSOR",
"SecurityZoneStatus":"BYPASSED"}, {"SecurityZoneId":"11",
"SecurityZoneName":"KITCHEN MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"12",
"SecurityZoneName":"KITCHEN DOOR SENSOR", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"13", "SecurityZoneName":"DINING ROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"14",
"SecurityZoneName":"LIVINGROOM LEFT DOOR SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"15",
"SecurityZoneName":"LIVINGROOM RIGHT DOOR SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"16",
```

```
"SecurityZoneName":"LIVINGROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"17",
"SecurityZoneName":"POWDER ROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"18",
"SecurityZoneName":"MASTER BEDROOM DOOR SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"19",
"SecurityZoneName":"MASTER BEDROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"20",
"SecurityZoneName":"MASTER BATHROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"21",
"SecurityZoneName":"MAIN LEVEL SMOKE DETECTOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"22",
"SecurityZoneName":"FRONT DOOR SENSOR", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"23", "SecurityZoneName":"UPSTAIRS MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"24",
"SecurityZoneName":"UPSTAIRS SMOKE DETECTOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"25",
"SecurityZoneName":"PLAYROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"26",
"SecurityZoneName":"UPSTAIRS FRONT BEDROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"27",
"SecurityZoneName":"UPSTAIRS BATHROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"28",
"SecurityZoneName":"UPSTAIRS BACK BEDROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"29",
"SecurityZoneName":"BASEMENT SMOKE DETECTOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"30",
"SecurityZoneName":"GARAGE MOTION SENSOR", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"31", "SecurityZoneName":"BASEMENT DOOR SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"32",
"SecurityZoneName":"BASEMENT BAR MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"33",
"SecurityZoneName":"BASEMENT FIREPLACE MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"34",
"SecurityZoneName":"BASEMENT GUEST BEDROOM DOOR SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"35",
"SecurityZoneName":"BASEMENT GUEST BEDROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"36",
"SecurityZoneName":"BASEMENT GUEST BATHROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"37",
"SecurityZoneName":"WELL WATER TANK MOISTURE SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"38",
"SecurityZoneName":"GARAGE DOOR SENSOR", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"39", "SecurityZoneName":"LAUNDRY ROOM DOOR SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"40",
"SecurityZoneName":"LAUNDRY ROOM MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"41",
"SecurityZoneName":"JACUZZI COVER SENSOR", "SecurityZoneStatus":"CLOSED"},
{"SecurityZoneId":"42", "SecurityZoneName":"JACUZZI MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"43",
"SecurityZoneName":"DRIVEWAY ENTERANCE MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"44",
"SecurityZoneName":"SIDE YARD MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"45",
"SecurityZoneName":"DRIVEWAY EXIT MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"46",
"SecurityZoneName":"GARAGE SCREEN DOOR SENSOR",
"SecurityZoneStatus":"BYPASSED"}, {"SecurityZoneId":"47",
"SecurityZoneName":"BACKDECK MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"48",
"SecurityZoneName":"BOATRAMP MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"49",
```

```
"SecurityZoneName":"PORTABLE MOTION SENSOR",
"SecurityZoneStatus":"CLOSED"}, {"SecurityZoneId":"50",
"SecurityZoneName":"GAZEBO MOTION SENSOR", "SecurityZoneStatus":"CLOSED"}
] }
```

Get the id and names of all security zones.

```
/SecurityData!SecurityCmd=GetZones!Token=a1j278kw72b4cd2s
```

```
{"SecurityZones": [ {"SecurityZoneId":"1", "SecurityZoneName":"NONE"},
{"SecurityZoneId":"2", "SecurityZoneName":"Boathouse Door Sensor"},
{"SecurityZoneId":"3", "SecurityZoneName":"Boathouse Garage Door Sensor"},
{"SecurityZoneId":"4", "SecurityZoneName":"Boathouse Motion Sensor"},
{"SecurityZoneId":"5", "SecurityZoneName":"Hvac Water Sensor"},
{"SecurityZoneId":"6", "SecurityZoneName":"Basement Office Motion
Detector"}, {"SecurityZoneId":"7", "SecurityZoneName":"NONE"},
{"SecurityZoneId":"8", "SecurityZoneName":"NONE"}, {"SecurityZoneId":"9",
"SecurityZoneName":"NONE"}, {"SecurityZoneId":"10",
"SecurityZoneName":"Miscellaneous Door Sensor"}, {"SecurityZoneId":"11",
"SecurityZoneName":"Kitchen Motion Sensor"}, {"SecurityZoneId":"12",
"SecurityZoneName":"Kitchen Door Sensor"}, {"SecurityZoneId":"13",
"SecurityZoneName":"Dining Room Motion Sensor"}, {"SecurityZoneId":"14",
"SecurityZoneName":"Livingroom Left Door Sensor"}, {"SecurityZoneId":"15",
"SecurityZoneName":"Livingroom Right Door Sensor"},
{"SecurityZoneId":"16", "SecurityZoneName":"Livingroom Motion Sensor"},
{"SecurityZoneId":"17", "SecurityZoneName":"Powder Room Motion Sensor"},
{"SecurityZoneId":"18", "SecurityZoneName":"Master Bedroom Door Sensor"},
{"SecurityZoneId":"19", "SecurityZoneName":"Master Bedroom Motion
Sensor"}, {"SecurityZoneId":"20", "SecurityZoneName":"Master Bathroom
Motion Sensor"}, {"SecurityZoneId":"21", "SecurityZoneName":"Main Level
Smoke Detector"}, {"SecurityZoneId":"22", "SecurityZoneName":"Front Door
Sensor"}, {"SecurityZoneId":"23", "SecurityZoneName":"Upstairs Motion
Sensor"}, {"SecurityZoneId":"24", "SecurityZoneName":"Upstairs Smoke
Detector"}, {"SecurityZoneId":"25", "SecurityZoneName":"Playroom Motion
Sensor"}, {"SecurityZoneId":"26", "SecurityZoneName":"Upstairs Front
Bedroom Motion Sensor"}, {"SecurityZoneId":"27",
"SecurityZoneName":"Upstairs Bathroom Motion Sensor"},
{"SecurityZoneId":"28", "SecurityZoneName":"Upstairs Back Bedroom Motion
Sensor"}, {"SecurityZoneId":"29", "SecurityZoneName":"Basement Smoke
Detector"}, {"SecurityZoneId":"30", "SecurityZoneName":"Garage Motion
Sensor"}, {"SecurityZoneId":"31", "SecurityZoneName":"Basement Door
Sensor"}, {"SecurityZoneId":"32", "SecurityZoneName":"Basement Bar Motion
Sensor"}, {"SecurityZoneId":"33", "SecurityZoneName":"Basement Fireplace
Motion Sensor"}, {"SecurityZoneId":"34", "SecurityZoneName":"Basement
Guest Bedroom Door Sensor"}, {"SecurityZoneId":"35",
"SecurityZoneName":"Basement Guest Bedroom Motion Sensor"},
{"SecurityZoneId":"36", "SecurityZoneName":"Basement Guest Bathroom Motion
Sensor"}, {"SecurityZoneId":"37", "SecurityZoneName":"Well Water Tank
Moisture Sensor"}, {"SecurityZoneId":"38", "SecurityZoneName":"Garage Door
Sensor"}, {"SecurityZoneId":"39", "SecurityZoneName":"Laundry Room Door
Sensor"}, {"SecurityZoneId":"40", "SecurityZoneName":"Laundry Room Motion
Sensor"}, {"SecurityZoneId":"41", "SecurityZoneName":"Jacuzzi Cover
Sensor"}, {"SecurityZoneId":"42", "SecurityZoneName":"Jacuzzi Motion
Sensor"}, {"SecurityZoneId":"43", "SecurityZoneName":"Driveway Entrance
Motion Sensor"}, {"SecurityZoneId":"44", "SecurityZoneName":"Side Yard
Motion Sensor"}, {"SecurityZoneId":"45", "SecurityZoneName":"Driveway Exit
Motion Sensor"}, {"SecurityZoneId":"46", "SecurityZoneName":"Garage Screen
Door Sensor"}, {"SecurityZoneId":"47", "SecurityZoneName":"Backdeck Motion
Sensor"}, {"SecurityZoneId":"48", "SecurityZoneName":"Boatramp Motion
Sensor"}, {"SecurityZoneId":"49", "SecurityZoneName":"Portable Motion
Sensor"}, {"SecurityZoneId":"50", "SecurityZoneName":"Gazebo Motion
Sensor"} ] }
```

11.0 Telephone Area

11.1 TelephoneService

Description Used to manage telephony services.

Method Get

Syntax /TelephoneService

The parameters are:

Parameter	Description
!TelephoneCmd	Command to be performed.
!TelephoneValue	A string value specifying the number to be dialed or state of answering machine.
!MailboxNumber	A numeric value to specify the mailbox to use for a TelephoneCmd.
!MailboxName	A string value to specify the mailbox to use for a TelephoneCmd.
!MsgDate	A date of format DD/MM/YY to use for a TelephoneCmd.
!MsgTime	A time of format HH:MM to use for a TelephoneCmd.
?Token	Required string expression authorizing access HALws.

TelephoneCmd Values

String	Description
"DeleteAllMsg"	Delete all voice messages in the specified Mailbox.
"DeleteMsg"	Delete message specified by MsgData and MsgTime.
"NewMsg"	Mark message specified by MsgData and MsgTime as New.
"SaveMsg"	Mark message specified by MsgData and MsgTime as Saved.
"Rings"	Set the number of rings before Answering Machine answers.
"AnsweringMachine"	Set the answering machine on or off based on TelephoneValue.
"Dial"	Dial the number specified by TelephoneValue.
"Hangup"	Hangup the telephone line.
"SpeakerPhone"	Turn speakerphone on.
"Answer"	Answer the telephone line.
"GetStatus"	Get the status of the modem.

Response Values

Response
<code>{"Status": "Error", "Description": "Invalid TelephoneCmd"}</code>
<code>{"Status": "Error", "Description": "Missing TelephoneCmd"}</code>
<code>{"Status": "Error", "Description": "Invalid TelephoneValue"}</code>
<code>{"Status": "Error", "Description": "Missing TelephoneValue"}</code>
<code>{"Status": "Error", "Description": "Invalid MailboxNumber"}</code>
<code>{"Status": "Error", "Description": "Missing MailboxNumber"}</code>
<code>{"Status": "Error", "Description": "Missing MailboxName"}</code>
<code>{"Status": "Error", "Description": "Invalid MsgDate"}</code>
<code>{"Status": "Error", "Description": "Missing MsgDate"}</code>
<code>{"Status": "Error", "Description": "Invalid MsgTime"}</code>
<code>{"Status": "Error", "Description": "Missing MsgTime"}</code>
<code>{"Status": "Error", "TelephoneCmd": "NEWMSG"}</code>
<code>{"Status": "Error", "TelephoneCmd": "SAVEMSG"}</code>
<code>{"Status": "Error", "TelephoneCmd": "DELETEALLMSGS"}</code>

Example Commands and Responses

Get the status of the Modem.

```
/TelephoneService!TelephoneCmd=GetStatus?Token=a1j278kw72b4cd2s
```

```
{"Status": "DIALING", "TelephoneCmd": "GETSTATUS"}
```

Delete all messages in the specified mailbox. If no mailbox given, main mailbox used.

```
/TelephoneService!TelephoneCmd>DeleteAllMsgs?Token=a1j278kw72b4cd2s
```

```
{"Status": "OK", "TelephoneCmd": "DELETEALLMSGS"}
```

Set a message status to be new.

```
/TelephoneService!TelephoneCmd=NewMsg!MsgDate=11/09/15!MsgTime=09%3a06?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"NEWMMSG"}
```

Set a message status to be saved.

```
/TelephoneService!TelephoneCmd=SaveMsg!MsgDate=11/09/15!MsgTime=09%3a06?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"SAVEMSG"}
```

Delete a message.

```
/TelephoneService!TelephoneCmd>DeleteMsg!MsgDate=01/27/16!MsgTime=11:51?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"DELETEMMSG"}
```

Set the ring count for the answering machine.

```
/TelephoneService!TelephoneCmd=Rings!TelephoneValue=6?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"RINGS", "TelephoneValue":"6"}
```

Set the answering machine on.

```
/TelephoneService!TelephoneCmd=AnsweringMachine!TelephoneValue=On?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"ANSWERINGMACHINE", "TelephoneValue":"ON"}
```

Dial a phone number.

```
/TelephoneService!TelephoneCmd=Dial!TelephoneValue=999-555-1212?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"DIAL", "TelephoneValue":"999-555-1212"}
```

Hangup the telephone line.

```
/TelephoneService!TelephoneCmd=Hangup?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"HANGUP"}
```

Enable the modem's speakerphone.

```
/TelephoneService!TelephoneCmd=SpeakerPhone?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"SPEAKERPHONE"}
```

Answer the telephone line.

```
/TelephoneService!TelephoneCmd=Answer?Token=a1j278kw72b4cd2s
```

```
{"Status":"OK", "TelephoneCmd":"ANSWER"}
```

Get the status of the Modem.

```
/TelephoneService!TelephoneCmd=GetStatus?Token=a1j278kw72b4cd2s
```

```
{"Status":"OFFHOOK", "TelephoneCmd":"GETSTATUS"}
```

Get the status of the Modem.

```
/TelephoneService!TelephoneCmd=GetStatus?Token=a1j278kw72b4cd2s
```

```
{"Status":"RINGING", "TelephoneCmd":"GETSTATUS"}
```

Get the status of the Modem.

```
/TelephoneService!TelephoneCmd=GetStatus?Token=a1j278kw72b4cd2s
```

```
{"Status":"ONHOOK", "TelephoneCmd":"GETSTATUS"}
```

11.2 TelephoneData

Description Used to retrieve Telephone data

Method Get

Syntax /TelephoneData

The parameters are:

Parameter	Description
!TelephoneCmd	Command to be performed.
!MailboxNumber	A numeric value to specify the mailbox to use for a TelephoneCmd.
!MailboxName	A string value to specify the mailbox to use for a TelephoneCmd.
?Token	Required string expression authorizing access HALws.

TelephoneCmd Values

String	Description
"GetMailboxes"	Retrieves a list of mailbox names and numbers.
"GetCallsIn"	Retrieves a list of calls in.
"GetCallsOut"	Retrieves a list of calls out.
"GetBlockList"	Retrieves the block caller list.
"GetDirectory"	Retrieves a list of the directory.
"GetNewMsgs"	Retrieves a list of new messages for the specified Mailbox Number or Name.
"GetSavedMsgs"	Retrieves a list of saved messages for the specified Mailbox Number or Name.

Response Values

Responses

```
{ "Status": "Error", "Description": "Invalid TelephoneCmd" }
{ "Status": "Error", "Description": "Missing TelephoneCmd" }
{ "Status": "Error", "Description": "Invalid MailboxNumber" }
{ "Status": "Error", "Description": "Missing MailboxNumber" }
{ "Status": "Error", "Description": "Missing MailboxName" }
```

Example Commands and Responses

Get mailbox names, numbers and message counts.

```
/TelephoneData!TelephoneCmd=GetMailBoxes?Token=a1j278kw72b4cd2s
{ "TelephoneMailboxes": [ { "MailboxName": "TIM", "Number": "0",
"NewMsgs": "0", "SavedMsgs": "265" }, { "MailboxName": "ANONYMOUS",
"Number": "1", "NewMsgs": "0", "SavedMsgs": "0" }, { "MailboxName": "UNKNOWN
CALLER", "Number": "2", "NewMsgs": "0", "SavedMsgs": "2" },
{ "MailboxName": "TEST", "Number": "3", "NewMsgs": "0", "SavedMsgs": "2" },
{ "MailboxName": "INTERCEPTED", "Number": "4", "NewMsgs": "0",
"SavedMsgs": "0" } ] }
```

Get the calls in list.

```
/TelephoneData!TelephoneCmd=GetCallsIn?Token=a1j278kw72b4cd2s
{ "TelephoneCallsIn": { "DATE": "01/28/16", "TIME": "19:24", "NUMBER": "(1999)
555-1212", "NAME": "Chesapeake VA" }, { "DATE": "01/28/16", "TIME": "18:30",
"NUMBER": "(1855) 442-5435", "NAME": "+18554425435" }, { "DATE": "01/28/16",
"TIME": "18:14", "NUMBER": "(1999) 555-1212", "NAME": "Vivian Shriver at
home" }, { "DATE": "01/28/16", "TIME": "15:16", "NUMBER": "(1999) 555-1212",
"NAME": "BECKER ERIC" }, { "DATE": "01/28/16", "TIME": "12:00",
"NUMBER": "Unknown", "NAME": "Unknown" }, { "DATE": "01/28/16", "TIME": "10:24",
"NUMBER": "(1855) 442-5435", "NAME": "+18554425435" }, { "DATE": "01/27/16",
"TIME": "17:41", "NUMBER": "(1999) 555-1212", "NAME": "DIRECTV" },
{ "DATE": "01/27/16", "TIME": "15:55", "NUMBER": "(1999) 555-1212",
"NAME": "WIRELESS CALLER" }, { "DATE": "01/27/16", "TIME": "11:51",
"NUMBER": "(1999) 555-1212", "NAME": "Unavailable" } ] }
```

Get the calls out list.

```
/TelephoneData!TelephoneCmd=GetCallsOut?Token=a1j278kw72b4cd2s
```

```
{"TelephoneCallsOut":{"DATE":"03/01/16", "TIME":"12:14", "NUMBER":"855-442-5435", "NAME":"TIM SHRIVER"}, {"DATE":"01/28/16", "TIME":"17:50", "NUMBER":"1 999 555-1212", "NAME":"Local Dial"}, {"DATE":"01/28/16", "TIME":"11:53", "NUMBER":"1 999 555-1212", "NAME":"Local Dial"}, {"DATE":"01/28/16", "TIME":"11:26", "NUMBER":"1 999 555-1212", "NAME":"Local Dial"}, {"DATE":"01/27/16", "TIME":"21:47", "NUMBER":"1 999 555-1212", "NAME":"Local Dial"} ] }
```

Get block caller list.

```
/TelephoneData!TelephoneCmd=GetBlockList?Token=a1j278kw72b4cd2s
```

```
{"TelephoneBlockList":{"LASTNAME":"","FIRSTNAME":"","COMPANY":"Unavailable Marketer", "STREET":"","CITY":"","STATE":"","ZIP":"","PHONE":"(866) 999-1212", "WORKPHONE":"","CELLPHONE":""} ] }
```

Get the directory listing.

```
/TelephoneData!TelephoneCmd=GetDirectory?Token=a1j278kw72b4cd2s
```

```
{"TelephoneDirectory":{"LASTNAME":"","FIRSTNAME":"","COMPANY":"HAL", "STREET":"","CITY":"","STATE":"","ZIP":"","PHONE":"855 442-5435", "WORKPHONE":"","CELLPHONE":""}, {"LASTNAME":"Snyder", "FIRSTNAME":"George", "COMPANY":"","STREET":"","CITY":"","STATE":"MD", "ZIP":"","PHONE":"1 999 555 1212", "WORKPHONE":"","CELLPHONE":""}, {"LASTNAME":"Stephens", "FIRSTNAME":"SUSAN", "COMPANY":"","STREET":"","CITY":"","STATE":"MD", "ZIP":"","PHONE":"","WORKPHONE":"","CELLPHONE":"999 555 1212"}, {"LASTNAME":"Stevenson", "FIRSTNAME":"Bill", "COMPANY":"","STREET":"","CITY":"","STATE":"TX", "ZIP":"","PHONE":"","WORKPHONE":"","CELLPHONE":""}, {"LASTNAME":"Walrath", "FIRSTNAME":"Doug", "COMPANY":"","STREET":"","CITY":"","STATE":"MD", "ZIP":"","PHONE":"","WORKPHONE":"1 999 555 1212", "CELLPHONE":""} ] }
```

Get list of new messages in a specified mailbox.

```
/TelephoneData!TelephoneCmd=GetNewMsgs!MailboxNumber=0?Token=a1j278kw72b4cd2s
```

```
{"TelephoneNewMsgs": {"DATE":"11/09/15", "TIME":"09:06", "NUMBER":"0", "CIDNAME":"Silzer Daryl", "CIDNUMBER":"(999) 555-1212", "LENGTH":"14.0", "FILE":"91061710.wav"} ] }
```

Get list of saved messages in a specified mailbox.

```
/TelephoneData!TelephoneCmd=GetSavedMsgs!MailboxNumber=2?Token=a1j278kw72b4cd2s
```

```
{"TelephoneSavedMsgs": {"DATE":"01/04/16", "TIME":"18:40", "NUMBER":"0", "CIDNAME":"Neary Dennis", "CIDNUMBER":"(999) 555-1212", "LENGTH":"14.0", "FILE":"274503783.wav"}, {"DATE":"01/02/16", "TIME":"19:37", "NUMBER":"0", "CIDNAME":"Cell Phone Va", "CIDNUMBER":"(999) 555-1212", "LENGTH":"19.3", "FILE":"105131839.wav"}, {"DATE":"12/13/15", "TIME":"18:03", "NUMBER":"0", "CIDNAME":"Unknown", "CIDNUMBER":"Unknown", "LENGTH":"19.8", "FILE":"262186591.wav"}, {"DATE":"12/08/15", "TIME":"15:48", "NUMBER":"0", "CIDNAME":"Jakubowski Paul", "CIDNUMBER":"(999) 555-1212", "LENGTH":"25.0", "FILE":"333628246.wav"}, {"DATE":"12/01/15", "TIME":"13:21", "NUMBER":"0", "CIDNAME":"Unknown", "CIDNUMBER":"Unknown", "LENGTH":"17.5", "FILE":"14995900.wav"}, {"DATE":"11/09/15", "TIME":"09:06", "NUMBER":"0", "CIDNAME":"Silzer Daryl", "CIDNUMBER":"(999) 555-1212", "LENGTH":"14.0", "FILE":"91061710.wav"}, {"DATE":"10/24/15", "TIME":"18:11", "NUMBER":"0", "CIDNAME":"Pena Tammy", "CIDNUMBER":"(999) 555-1212", "LENGTH":"53.0", "FILE":"20833676.wav"} ] }
```

11.3 VoiceMsg

Description Used to retrieve Telephone data

Method Get

Syntax /VoiceMsg

The parameters are:

Parameter	Description
Filename	Voice message file to be downloaded.
?Token	Required string expression authorizing access HALws.

Response Values

Responses

```
{"Status":"Error", "Description":"Missing Filename"}
```

Example Commands and Responses

Get the voice message in a wav format.

```
/VoiceMsg/5209385.wav?Token=a1j278kw72b4cd2s
```

Get the voice message in a mp3 format.

```
/VoiceMsg/5209385.mp3?Token=a1j278kw72b4cd2s
```


12.0 Digital Music Center Area

12.1 DMCSERVICE

Description Used to manage DMC functions.

Method Get

Syntax /DMCSERVICE

The parameters are:

Parameter	Description
!DMCCmd	Command to be performed.
!DMCSong	Song to be played.
!DMCListType	Playlist type.
!DMCPlayList	Playlist name.
!DMCStationName	Radio station name.
!DMCStationURL	Radio station URL.
!DMCValue	Used for misc commands such as volume.
?Token	Required string expression authorizing access HALws.

DMCCmd Values

String	Description
"PlaySong"	Play the specified song.
"PlayStation"	Play the specified radio station.
"PlayUrl"	Play the specified URL stream.
"Play"	Play the current source.
"Stop"	Stop the current source.
"Pause"	Pause the current source.
"Next"	Next Song in playlist.
"Prev"	Previous song in playlist.
"Shuffle"	Toggle shuffle state.
"Repeat"	Toggle repeat mode.
"Mute"	Mute volume.
"Unmute"	Unmute volume.
"VolUp"	Raise volume.
"VolDown"	Lower volume.
"Volume"	Set the volume to a value between 0 and 100 as specified n DMCValue.
"Media"	Switch from Radio to Media.
"Radio"	Switch from Media to Radio.
"Location"	Set to audio output location.

DMCListType Values

String
"Custom"
"Artist"
"Album"
"Genre"
"Year"
"Advanced"
"All Songs"

Response Values

Response
<code>{"Status": "Error", "Description": "Invalid DMCCmd"}</code>
<code>{"Status": "Error", "Description": "Missing DMCCmd"}</code>
<code>{"Status": "Error", "Description": "Missing DMCSong"}</code>
<code>{"Status": "Error", "Description": "Missing DMCListType"}</code>
<code>{"Status": "Error", "Description": "Invalid DMCListType"}</code>
<code>{"Status": "Error", "Description": "Missing DMCStationName"}</code>
<code>{"Status": "Error", "Description": "Missing DMCStationURL"}</code>

Example Commands and Responses

Play specified song by album.

```
/DMCservice!DMCCmd=PlaySong!DMCSong=Ice Cream!DMCPlaylist=Freedom
Sessions!DMCListType=Album?Token=a1j278kw72b4cd2s
{"Status":"Ok", "DMCcmd":"PLAYSONG"}
```

Play specified song by artist.

```
/DMCservice!DMCCmd=PlaySong!DMCSong=Ice Cream!DMCPlaylist=Sarah
McLachlan!DMCListType=Artist?Token=a1j278kw72b4cd2s
{"Status":"Ok", "DMCcmd":"PLAYSONG"}
```

Pause the current source.

```
DMCservice!DMCCmd=Pause?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"PAUSE"}
```

Play the current source.

```
DMCservice!DMCCmd=Play?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"PLAY"}
```

Stop the current source.

```
DMCservice!DMCCmd=Stop?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"STOP"}
```

Mute the current source.

```
DMCservice!DMCCmd=Mute?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"MUTE"}
```

Unmute the current source.

```
DMCservice!DMCCmd=Unmute?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"UNMUTE"}
```

Switch to radio mode.

```
DMCservice!DMCCmd=Radio?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"RADIO"}
```

Set the volume.

```
DMCservice!DMCCmd=Volume!DMCValue=75?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"VOLUME"}
```

Play a radio station.

```
/DMCservice!DMCCmd=PlayStation!DMCStationName=100%20Hitz?Token=a1j278kw72b
4cd2s
{"Status":"OK", "DMCcmd":"PLAYSTAION"}
```

Play a radio station URL.

```
/DMCservice!DMCCmd=Set!DMCAction=PlayUrl!DMCStationURL=http%3a%2f%2f69.4.2
34.186%3a9140!DMCStationName=100 Hitz?Token=a1j278kw72b4cd2s
{"Status":"OK", "DMCcmd":"PLAYURL"}
```

12.2 DMCDData

Description Used to retrieve DMC data

Method Get

Syntax /DMCDData

The parameters are:

Parameter	Description
!DMCCmd	Command to be performed.
!DMCSong	Song to be played.
!DMCListType	Playlist type.
!DMCPlaylist	Playlist name.
!DMCStationName	Radio station name.
!DMCStationURL	Radio station URL.
!DMCValue	Used for misc commands such as volume.
?Token	Required string expression authorizing access HALws.

TelephoneCmd Values

String	Description
"GetPlaylist"	Retrieves a list of mailbox names and numbers.
"GetStationList"	Retrieves a list of calls in.
"GetSongList"	Retrieves a list of calls out.
"GetNextSong"	Retrieves the block caller list.
"GetPrevSong"	Retrieves a list of the directory.
"GetCurrentSong"	Retrieves a list of new messages for the specified Mailbox Number or Name.
"GetCurrentPlaylist"	Retrieves a list of saved messages for the specified Mailbox Number or Name.
"GetLocation"	Retrieves a list of calls in.
"GetLocations"	Retrieves a list of calls out.
"GetVolume"	Retrieves the block caller list.
"GetMute"	Retrieves a list of the directory.
"GetStatus"	Retrieves a list of calls in.

Response Values

Responses
<pre>{"Status":"Error", "Description":"Invalid DMCCmd"}</pre>
<pre>{"Status":"Error", "Description":"Missing DMCCmd"}</pre>
<pre>{"Status":"Error", "Description":"Missing DMCSong"}</pre>
<pre>{"Status":"Error", "Description":"Missing DMCListType"}</pre>
<pre>{"Status":"Error", "Description":"Invalid DMCListType"}</pre>
<pre>{"Status":"Error", "Description":"Missing DMCStationName"}</pre>

Example Commands and Responses

Get play lists by album.

```
/DMCDdata!DMCCmd=GetPlaylist!DMCPlaylist=Album?Token=a1j278kw72b4cd2s
```

```
{"DMCPlaylist":"ALBUM", "DMCPlayLists": [ {"DMCList":"Freedom Sessions"}, {"DMCList":"Fumbling Towards Ecstasy"}, {"DMCList":"Solace"}, {"DMCList":"Surfacing"} ] }
```

Get play lists by artist.

```
/DMCDdata!DMCCmd=GetPlaylist!DMCPlaylist=Artist?Token=a1j278kw72b4cd2s
```

```
{"DMCPlaylist":"ARTIST", "DMCPlayLists": [ {"DMCList":"Freedom Sessions"}, {"DMCList":"Fumbling Towards Ecstasy"}, {"DMCList":"Sarah McLachlan"}, {"DMCList":"Solace"} ] }
```

Get play lists by year.

```
/DMCdata!DMCCmd=GetPlayList!DMCPlayList=Year?Token=alj278kw72b4cd2s  
{ "DMCPlayList": "YEAR", "DMCPlayLists": [ { "DMCList": "Nineties" } ] }
```

Get play lists by Genre.

```
/DMCdata!DMCCmd=GetPlayList!DMCPlayList=Genre?Token=alj278kw72b4cd2s  
{ "DMCPlayList": "GENRE", "DMCPlayLists": [ { "DMCList": "Blues" }, { "DMCList": "None" } ] }
```

Get station list.

```
/DMCdata!DMCCmd=GetStationList?Token=alj278kw72b4cd2s  
{ "DMCStations": [ { "DMCStation": "100 HITZ",  
"DMCStationURL": "HTTP://69.4.234.186:9140" }, { "DMCStation": "KISS FM",  
"DMCStationURL": "HTTP://80.86.106.136:80" } ] }
```

13.0 Speech Area

13.1 SpeechService

Description Used to Post a wav file to be processed by HAL for recognition.

Method Post

Syntax /SpeechService

The parameters are:

Parameter	Description
!SpeechCmd	Name of Mode to be referenced by method.
?Token	Required string expression authorizing access HALws.

SpeechCmd Values

String	Description
"Process"	The command to be performed by the speech service.

Response Values

Response
<pre>{"Status": "Error", "Description": "Missing SpeechCmd"}</pre>
<pre>{"Status": "Error", "Description": "Invalid SpeechCmd"}</pre>
<pre>{"Status": "Error", "Description": "Missing FileName"}</pre>

Notes

This service utilizes the HTTP Post Method to send a wav file to the Speech Service for processing. The wav file audio format should be PCM 16 bit at a sampling rate of 11025.

Example Commands and Responses

Post a wav file to the Speech Service to be processed.

```
/SpeechService!SpeechCmd=Process?Token=a1j278kw72b4cd2s
```

```
{"Status": "OK", "SpeechCmd": "PROCESS", "FileName": "MYAUDIO.WAV"}
```

13.2 SpeechData

Description Used to retrieve Speech data

Method Get

Syntax /SpeechData

The parameters are:

Parameter	Description
!SpeechCmd	Command to be performed.
?Token	Required string expression authorizing access HALws.

SpeechCmd Values

String	Description
"Get"	Retrieves most recent speech information.

Response Values

Response
<pre>{"Status": "Error", "Description": "Missing SpeechCmd"}</pre>
<pre>{"Status": "Error", "Description": "Invalid SpeechCmd"}</pre>

Example Commands and Responses

Multiple examples of getting speech information.

```
/SpeechData!SpeechCmd=Get!Token=alj278kw72b4cd2s
{"SpeechFileName": "LastAudio125604.mp3", "Speech": "Unable To Recognize
You.", "TTS": "Unable to recognize you.", "TimeIn": "3/22/2016 12:56:04
PM"}

{"SpeechFileName": "LastAudio125702.mp3", "Speech": "What Time Is It",
"TTS": "The Time Is 12:57 Pm.", "TimeIn": "3/22/2016 12:57:02 PM"}

{"SpeechFileName": "LastAudio125819.mp3", "Speech": "Hello Sally", "TTS":
"How May I Help You.", "TimeIn": "3/22/2016 12:58:19 PM"}

{"SpeechFileName": "LastAudio125846.mp3", "Speech": "When Is Sunset",
"TTS": "The Sunset For Today Is At 07:21 Pm.", "TimeIn": "3/22/2016
12:58:46 PM"}

{"SpeechFileName": "LastAudio125902.mp3", "Speech": "Thank You", "TTS":
"You're Welcome.", "TimeIn": "3/22/2016 12:59:02 PM"}
```

13.3 SpeechLastAudio

Description Used to retrieve a specific Audiofile of a TTS that was made by HAL.

Method Get

Syntax /LastAudio

The parameters are:

Parameter	Description
Timestamp	HHMMSS.wav
?Token	Required string expression authorizing access HALws.

Response Values

Response

```
{"Status":"Error", "Description":"Missing Filename"}
```

Example Commands and Responses

Get a specific TTS that was made by HAL

```
/LastAudio125902.wav?Token=a1j278kw72b4cd2s
```