

DIRECTV SHEF Public Beta Command Set

Published by



DIRECTV®

DTV-MD-0359

(Rev. 1.1.H)

December 15, 2010



REVISION HISTORY			
Revision	Date of Issue	Author	Scope
1.0	September 15, 2010	S.C. J. G.	Updated with SHEF spec v1.1DraftE
1.1.H	December 15, 2010	S.C. J. G.	Updated with SHEF spec v1.1DraftH

Public Beta
Subject to Change



Table Of Contents

<u>Section</u>	<u>Page</u>
1 Introduction	6
1.1 Disclaimer	6
1.2 Scope.....	6
1.3 Feedback	6
2 Set-top Box HTTP Exported Functionality (SHEF).....	7
2.1 Introduction.....	7
2.2 Network Security Settings	8
2.3 SHEF Opt In.....	8
2.4 Supported Functionality	9
2.4.1 DVR Functionality	9
2.4.2 TV Functionality	9
2.4.3 REMOTE Functionality	10
2.4.4 INFO Functionality	10
2.5 SHEF HTTP Status Codes	10
2.6 Limitations.....	11
2.7 Protocol Definition	11
3 Interface Functions	12
3.1 DVR Functionality	12
3.1.1 Get Playlist	12
3.1.1.1 Get Playlist Request Command.....	12
3.1.1.2 Get Playlist JSON Response.....	12
3.1.1.3 Get Playlist Sample Response	14
3.1.2 Play.....	16
3.1.2.1 Play Request Command.....	16
3.1.2.2 Play JSON Response	16
3.1.2.3 Play Sample Response.....	17
3.2 TV Functionality	17
3.2.1 Get Tuned.....	17
3.2.1.1 Get Tuned Request Command.....	17



3.2.1.2	Get Tuned JSON Response	17
3.2.1.3	Get Tuned Sample Response	19
3.2.2	Get Program Info.....	20
3.2.2.1	Get Program Info Request Command	20
3.2.2.2	Get Program Info JSON Response	20
3.2.2.3	Get Program Info Sample Response.....	20
3.2.3	Tune.....	20
3.2.3.1	Tune Request Command	20
3.2.3.2	Tune JSON Response	21
3.2.3.3	Tune Sample Response.....	21
3.3	Remote Keys Functionality	22
3.3.1	Remote Keys	22
3.3.1.1	Remote Keys Request Command	22
3.3.1.2	Remote Key JSON Response	22
3.3.1.3	Remote Key Sample Response	22
3.4	Info Functionality.....	23
3.4.1	Get Version	23
3.4.1.1	Get Version Request Command	23
3.4.1.2	Get Version JSON Response.....	23
3.4.1.3	Get Version Sample Response	23
3.4.2	Get Options	24
3.4.2.1	Get Options Request Command	24
3.4.2.2	Get Options Response	24
3.4.2.3	Get Options Sample Response	24



Table of Figures

Figure 2-1: SHEF Data Flow..... 7

Figure 2-2: SHEF External Device Settings..... 8

Figure 2-3: SHEF External Device Home Network Settings..... 9

List of Tables

Table 2-1 HTTP Status Codes 10

Table 2-2: STB Http Exported Functionality Request Syntax 11

Table 3-1: Get Playlist Request Fields Definition 12

Table 3-2: Get Playlist JSON Response Fields Definition..... 13

Table 3-3: Play Request Fields Definition..... 16

Table 3-4: Play JSON Response Definition..... 16

Table 3-5: Get Tuned Request Definition 17

Table 3-6: Get Tuned JSON Response Definition..... 17

Table 3-7: Get Program Info Request Definition..... 20

Table 3-8: Tune JSON Request Definition..... 20

Table 3-9: Tune JSON Response Definition 21

Table 3-10: Remote Key Request Fields Definition 22

Table 3-11: Remote Key JSON Response Definition 22

Table 3-12: Get Version Request Fields Definition 23

Table 3-13: Get Version JSON Response Definition 23

Table 3-14: Get Options Request Fields Definition..... 24

Table 3-15: Get Options JSON Response Definition 24



1 Introduction

1.1 Disclaimer

DIRECTV makes no representations or warranties, express or implied, that use of the technologies described in this specification will not infringe patents, copyrights, or other intellectual property rights of third parties. Nothing in this specification should be construed as granting permission to use any of the technologies described. Anyone planning to make use of technology covered by the intellectual property rights of others should first obtain permission from the holder(s) of the rights. This specification is subject to change without notice. DIRECTV does not accept any responsibility whatsoever for any damages or liability, direct or consequential, which may result from use of this specification or any related discussions. These specifications are provided “as is” and the user of these specifications assumes any and all risks associated with the use of these specifications. DIRECTV expressly disclaims any and all representations or warranties, express or implied, regarding the specifications, including without limitation any warranty as to merchantability, fitness for a particular purpose, non-interruption of use, or non-infringement.

1.2 Scope

This document provides information on the Public Beta release of the DIRECTV Set-top box (STB) HTTP Exported Functionality (SHEF) as an aid for installers and system integrators. This document is relevant to DIRECTV high-definition set-top box models H21, HR20 and newer. Other models are not supported.

1.3 Feedback

DIRECTV requests and welcomes your feedback and suggestions on this Public Beta release of the Set-top box HTTP Exported Functionality (SHEF). DIRECTV will evaluate these comments and may incorporate them in future versions of SHEF.

Please email feedback to custominstallsupport@directv.com



2 Set-top Box HTTP Exported Functionality (SHEF)

2.1 Introduction

This section specifies the interface and protocol definition between the Web applications and the DIRECTV Set-top box HTTP Exported Functionality (SHEF). SHEF is supported on DIRECTV STB models H21 and newer models as well as HR20 and newer models.

The general function of the SHEF interface is to accept HTTP requests from Web applications, process them, and return any data or status responses that result from the request.

The interface between Web applications and the STB allows the Web applications to request services from the STB components of the DIRECTV set-top box. The Web applications send a HTTP request to a HTTP daemon running in the STB containing a service type and any necessary parameters required to complete the request. The STB responds with results formatted in HTTP and JSON (JavaScript Object Notation) compatible streams.

Important Note: DIRECTV STBs may support Network Connections via an Ethernet Port and/or a Coax Connection. It is critical that only one of these interfaces be used and that both network types not be connected simultaneously on a given STB. Use of the Coax Connection is preferred. A DIRECTV Ethernet to Coax Adapter (DECA) may be used to bridge the Coax Network to the Home Ethernet Network.

Figure 2-1: SHEF Data Flow shows the data flow between the Web application and through the SHEF Server.

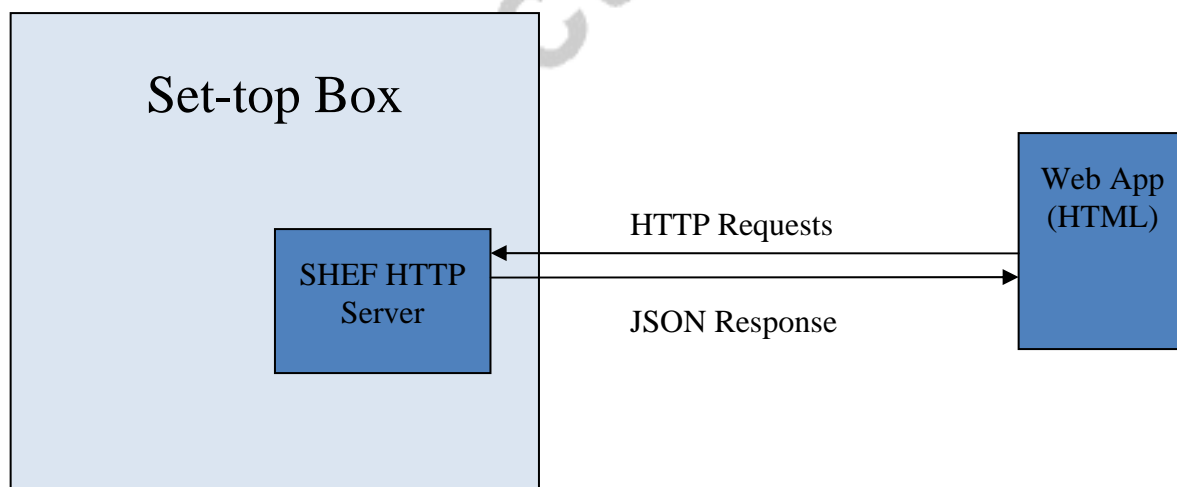


Figure 2-1: SHEF Data Flow

Services will be requested via the HTTP/1.1 protocol as specified in RFC 2616 (see <http://tools.ietf.org/html/rfc2616>). The format of these requests will utilize the 'GET' method requests.



2.2 Network Security Settings

Before using SHEF review appropriate home network security options such as password protection and encryption.

2.3 SHEF Opt In

In order to use SHEF in your network, it must be enabled in the set-top box by navigating to the “External Device” settings screen (Menu->System Setup->Whole-Home->External Device) screen. Once on this screen select “Allow” for “External Access” as shown in Figure 2-2. Also turn on the other options as desired. Read and acknowledge the caution as shown in Figure 2-3.

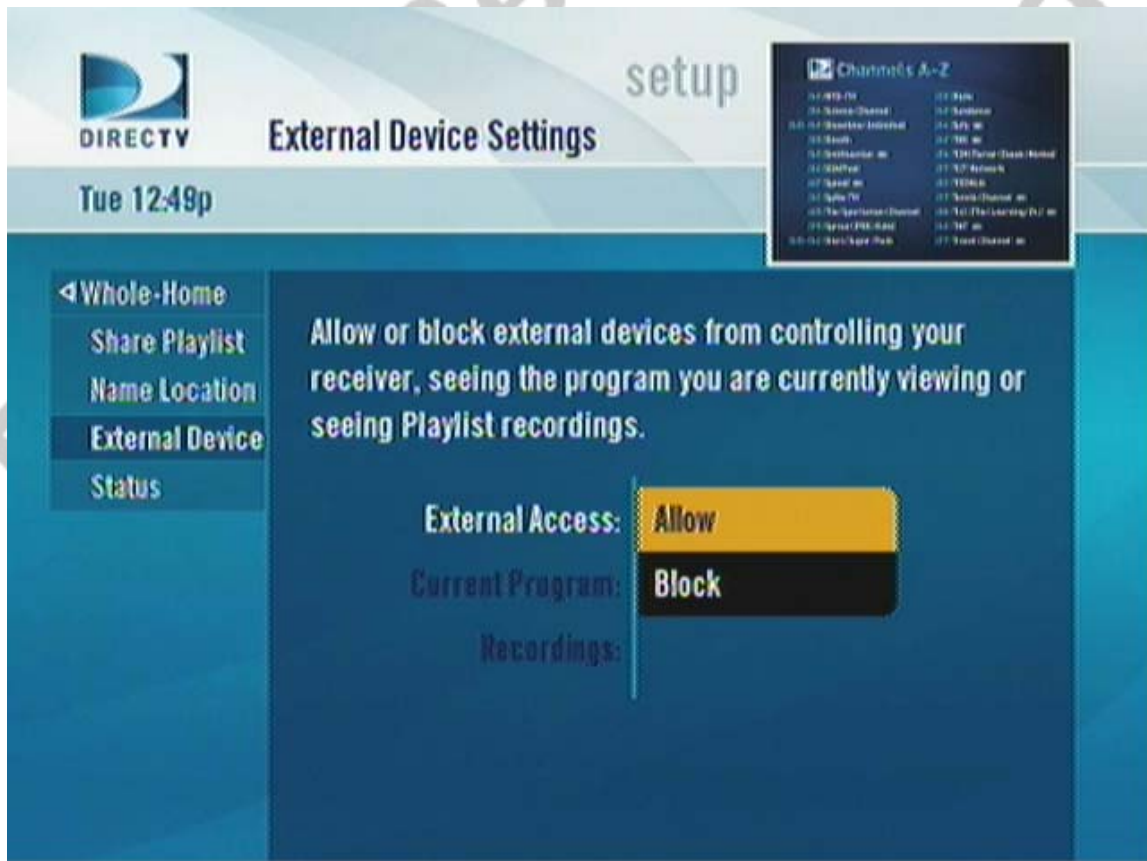


Figure 2-2: SHEF External Device Settings

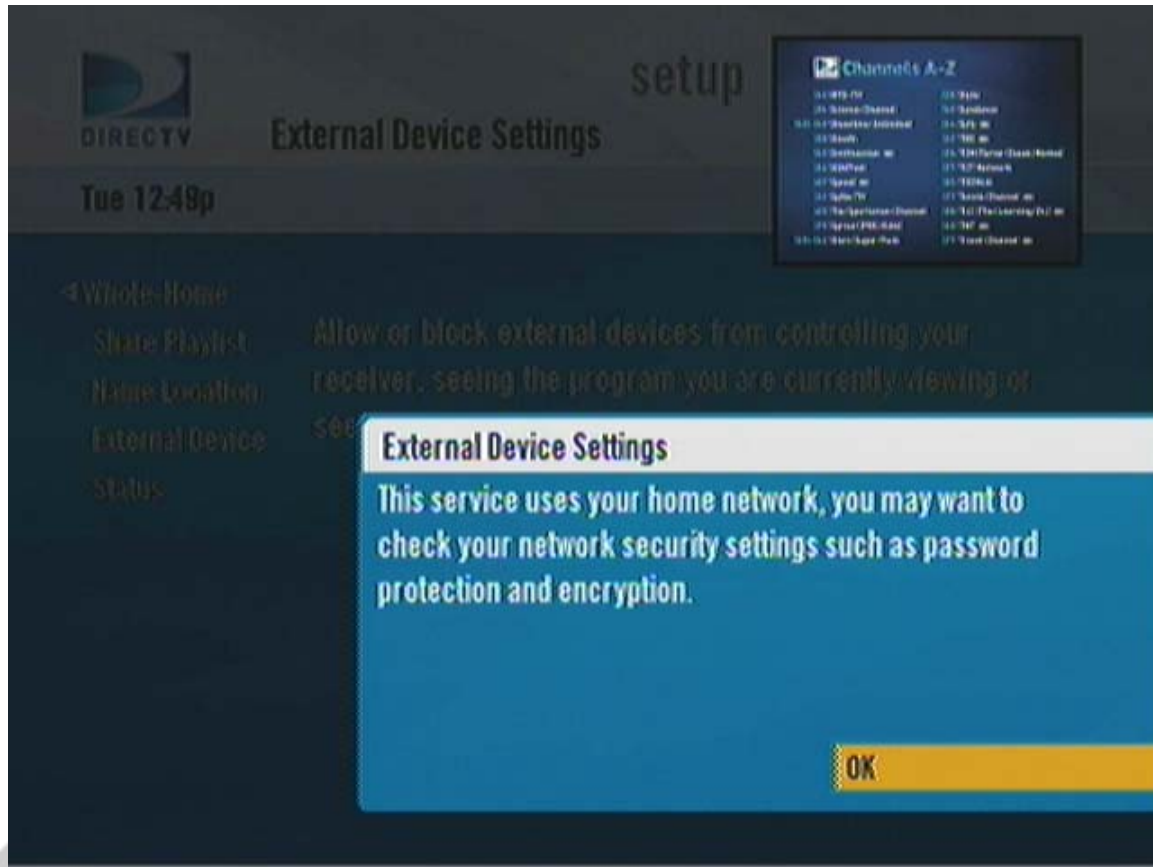


Figure 2-3: SHEF External Device Home Network Settings

2.4 Supported Functionality

The middleware core supports several requests for service from the SHEF. These requests may be made externally from networked or wireless devices, or internally via interactive applications.

SHEF requests include the following major functionality:

2.4.1 DVR Functionality

- **Get Playlist** – Return a list of recordings in the Playlist.
- **Play** – Play a recorded program in the Playlist.

2.4.2 TV Functionality

- **Get Tuned** – Return detailed information about the currently viewed program, either live or recorded.
- **Get Program Info** – Return detailed information about a program for a specified time in the query.
- **Tune** – Tune to a specified channel.



2.4.3 REMOTE Functionality

- **Simulate Remote Key** – Create KEY_PRESSED and KEY_RELEASED events.

2.4.4 INFO Functionality

- **Get Version** – Return the version of current implementations
- **Get Options** – Return a list of the available features (this list).

2.5 SHEF HTTP Status Codes

The following is a list of HTTP Status Codes that SHEF may return in the HTTP response. Additional HTTP Status codes not stated in this list may be added for future use.

Table 2-1 HTTP Status Codes

HTTP Status Code	Name	Description
200	OK.	Denotes that the HTTP request and response has succeeded. A successful SHEF HTTP response returns this code.
400	Bad Request.	The request contains malformed syntax. The request should not be resent.
403	Forbidden.	The server understood the request but is refusing to fulfill it.
409	Conflict.	The request could not be completed due to a conflict with resources. The user might be able to resolve the conflict and resubmit the request.
500	Internal Server Error	The server encountered an unexpected condition. The request cannot be fulfilled.
503	Service Unavailable.	The server is currently unable to handle the request due to a temporary overloading of the server. This is a temporary condition that should be resolved after some delay.
505	HTTP Version Not Supported.	The server does not support the HTTP protocol version of the request message.

In summary, only successful SHEF HTTP responses will return a status code = 200. All error SHEF HTTP responses will return an error HTTP status codes as defined by the HTTP specification (<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>). The range for the error HTTP status codes is 300 to 505.



2.6 Limitations

The SHEF will not queue requests. SHEF response times will provide sufficient responsiveness to be acceptable to a person controlling the STB.

2.7 Protocol Definition

Table 2-2: STB Http Exported Functionality Request Syntax

STB Http Exported Functionality request syntax
<p><code>http://127.0.0.1:8080/info/function?param1=<i>value</i> [&param2=<i>value</i>]</code></p> <p>In the interest of efficiency, using the well known hard coded address of the loopback adapter when possible is preferred over the equally well known name of the adapter. So for instance:</p> <p><code>http://127.0.0.1:8080/dvr/play?uniqueId=<i>num</i></code></p> <p>Would be preferred over:</p> <p><code>http://localhost:8080/dvr/play?uniqueId=<i>num</i></code></p> <p>Future expansion may require specification of an IP address outside of the local box. The STB will not maintain a list of hostnames nor be expected to do DNS lookups to resolve hostnames, but shall respond to any request that arrives.</p> <p>* <i>Italic</i> words are string variables representing the actual data. See STB Http Exported Functionality request Fields Definition for definitions of the available requests.</p>



3 Interface Functions

3.1 DVR Functionality

3.1.1 Get Playlist

When an app sends the initial `getPlaylist` command, SHEF will return a JSON response and a cookie is sent back in the HTTP header response. SHEF caches the playlist and its associated cookie. For subsequent `getPlaylist` calls, the app will use this cookie to get access to the same playlist instance

3.1.1.1 Get Playlist Request Command

If the app wants to retrieve a new instance of the playlist, it should not send a cookie in the HTTP header request. If the app wants to retrieve an existing instance of the playlist, it is required that a cookie is sent in the HTTP header request for that playlist instance.

The STB expects the cookie in the following format in the HTTP request header:

Cookie: `plSession=[alpha numeric string]`

Table 3-1: Get Playlist Request Fields Definition

Field	Description	Value	Default Value
Get Playlist – <code>http://STBIP:port/dvr/getPlaylist?[start=num][&max=num][&type=string]</code>			
<code>start</code>	Index (starting at zero) of first recorded item	1...count of recorded item	0
<code>max</code>	Maximum number of playlist items to return	1...25	25
<code>type</code>	Type of playlist items to return: all (all types of recordings) system (list of pushed recordings) user (list of user recordings)	Fixed value filter string	user

3.1.1.2 Get Playlist JSON Response

The following is the format of the STB's HTTP response to set a cookie:

Set-Cookie: `plSession=[alpha numeric string]; expires=[Day of Week], DD-MMM-YYYY HH:MM:SS Timezone; path=/dvr/getPlaylist; domain=[STB's IP]`

**Table 3-2: Get Playlist JSON Response Fields Definition**

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			
hasChanged	boolean	An indicator that the user's playlist has added or removed recordings	✓
itemsReturned	number	number of items returned in this getPlayList call	
totalNumItems	number	Total number of items in the playlist	
type	string	All, system, or user	
playList [
{		//grouped events	
groupIndex	number	number from 0..n to uniquely identify the group	
groupTitle	string	title of the grouped events	
isRecording	boolean	whether any of the grouped events are currently being recorded	
numItems	number	number of items in the group	
numNewItems	number	number of unviewed items in the group	
recordings [
{			
contentId	string	unique identifier for this movie or program	✓
programId	number	program object id	
uniqueId	string	unique identifier of the event	
materialID	number	material ID only available for VOD and push titles	✓
startTime	number	recorded start time in seconds of the event in UTC	
duration	number	duration in seconds of the event	
major	number	major channel number	
minor	number	minor channel number	
callsign	string	call letter of the channel	
isOffAir	boolean	whether the event is an ATSC event	
isVod	boolean	whether the event is a Video-On-Demand event	
isPpv	boolean	whether the event is a Pay-Per-View event	
isPurchased	boolean	only available if isPpv returns true; whether the event is purchased	✓
isRecording	boolean	whether the event is currently being recorded	
rating	string	rating of the event	



title	string	title of the program	
keepUntilFull	boolean	true: the event will be deleted when the disk is full and more space is needed; false: the event will not be deleted unless user chooses to do so	
isViewed	boolean	whether the event has been viewed	
expiration	string	expiration day & time of the event	
isPartial	boolean	whether the event is a partially recorded	
crid	string	only available if the event is a crid recording; content reference identifier service name	✓
offset	number	number of seconds from the scheduled start time of a recording	
url	string	URL to play out content	✓
}} ...			
} ...			
]			

3.1.1.3 Get Playlist Sample Response

```
{
  "hasChanged": false,
  "itemsReturned": 25,
  "playList": [
    {
      "groupIndex": 0
      "groupTitle": "GIADA",
      "isRecording": false,
      "numItems": 2,
      "numNewItems": 2,
      "recordings": [
        {
          "callsign": "COOK",
          "contentId": "1 1 8A901 98",
          "duration": 1800,
          "expiration": "0",
          "isOffAir": false,
          "isPartial": false,
          "isPpv": false,
          "isRecording": false,
```



```
"isViewed": false,
"isVod": false,
"keepUntilFull": true,
"major": 232,
"minor": 65535,
"offset": 0,
"programId": "3894732",
"rating": "No Rating",
"startTime": 1280863800,
"title": "Everyday Italian",
"uniqueId": "5715372633822084281",
},
{
  "callsign": "FOODHD",
  "contentId": "1 1 1093FC 23",
  "duration": 245,
  "expiration": "0",
  "isOffAir": false,
  "isPartial": true,
  "isPpv": false,
  "isRecording": false,
  "isViewed": false,
  "isVod": false,
  "keepUntilFull": true,
  "major": 231,
  "minor": 65535,
  "offset": 0,
  "programId": "5415938",
  "rating": "No Rating",
  "startTime": 1280853000,
  "title": "Giada at Home",
  "uniqueId": "8715790354843114519",
  "url": "http://192.168.1.101:9000/playback?id=avmedia-0x78f4b52645331c17"
}
]
},...
```



```

],
"status": {
  "code": 200,
  "msg": "OK.",
  "query": "/dvr/getPlayList"
}
"totalNumItems": 30,
"type": "user"
}

```

3.1.2 Play

3.1.2.1 Play Request Command

Table 3-3: Play Request Fields Definition

Field	Description	Value	Default Value
Play – http://STBIP:port/dvr/play?uniqueId=num[&playFrom=string][&offset=num]			
uniqueId	Begin playback of recorded program where <i>uniqueId</i> specifies a unique identifier returned by a call to Get Playlist.	Number returned from Get Playlist	
playFrom	Where to begin playback. One of the following strings: start (beginning of recording) resume (previous point playback was stopped or beginning if program has not previously been played back) offset (begin a number of seconds from the beginning of recording)	Fixed value operation string	resume
offset	Number of seconds from beginning of recording to start at. Note: Offset is only used when the PlayFrom field is set to “offset”.	0...number of seconds in the program.	0

3.1.2.2 Play JSON Response

Table 3-4: Play JSON Response Definition

Name	Type	Description	Optional
status {			



Name	Type	Description	Optional
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			

3.1.2.3 Play Sample Response

```
{ "status": {
  "code": 200,
  "msg": "OK",
  "query": "/dvr/play?uniqueId=2873298685953728822"
}}
```

3.2 TV Functionality

3.2.1 Get Tuned

3.2.1.1 Get Tuned Request Command

Table 3-5: Get Tuned Request Definition

Field	Description	Value	Default Value
Get Tuned - http://STBIP:port/tv/getTuned			
None	No fields are defined for this request.	N/A	N/A

3.2.1.2 Get Tuned JSON Response

Table 3-6: Get Tuned JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			
stationId	number	unique identifier for the channel	
contentId	string	unique identifier for this movie or program	✓
programId	number	program object id	
materialID	number	material ID	✓



Name	Type	Description	Optional
		only available for VOD and push titles	
startTime	number	start time in seconds of a live event or when a recording happens in UTC time	
duration	number	duration in seconds of a live event or actual duration of a recording	
major	number	major channel number	
minor	number	minor channel number	
callsign	string	call letter of a channel	
isOffAir	boolean	whether the event is an ATSC event	
isVod	boolean	whether the event is a Video-On-Demand event	
isPpv	boolean	whether the event is a Pay-Per-View event	
isPurchased	boolean	only available if isPpv returns true; whether the event is purchased	✓
isRecording	boolean	whether the event is currently being recorded	
rating	string	rating of the event	
isPclocked	number	whether the parental control is enabled; 1: locked, 2: temporarily unlocked, 3: unlocked	
title	string	title of the program	
uniqueId	string	only available if the event is a recording; unique identifier of the event	✓
keepUntilFull	boolean	only available if the event is a recording; true: the event will be deleted when the disk is full and more space is needed; false: the event will not be deleted unless user chooses to do so	✓
isViewed	boolean	only available if the event is a recording; whether the event has been viewed	✓
expiration	string	only available if the event is a recording; expiration day & time of the recording	✓
recType	number	only available if the event is a recording; type of recordings; 1: manual recording; 2: findby recording; 3: regular recording	✓
findbyWord	string	only available if the event is recorded based on search criteria (recType=2); words used to search recordings	✓
isPartial	boolean	only available if the event is a recording; whether the event is partially recorded	✓
priority	string	only available if the event is a series recording; priority order of the event; format: n of total priorities	✓
music {		only available if the event is a music channel; music sound track information	✓



Name	Type	Description	Optional
by	string	artist of the music	
cd	string	name of the cd	
title	string	title of the song	
}			
offset	number	number of seconds from the scheduled start time of a recording	

3.2.1.3 Get Tuned Sample Response

```
{
  "callsign": "FOODHD",
  "contentId": "1 1 89237 37",
  "duration": 1791,
  "expiration": "0",
  "isOffAir": false,
  "isPartial": false,
  "isPclocked": 1,
  "isPpv": false,
  "isRecording": false,
  "isViewed": true,
  "isVod": false,
  "keepUntilFull": true,
  "major": 231,
  "minor": 65535,
  "offset": 263,
  "programId": "4405732",
  "rating": "No Rating",
  "recType": 3,
  "startTime": 1278342008,
  "stationId": 3900976,
  "status": {
    "code": 200,
    "msg": "OK.",
    "query": "/tv/getTuned"
  },
  "title": "Tyler's Ultimate",
  "uniqueId": "6728716739474078694"
```



}

3.2.2 Get Program Info

3.2.2.1 Get Program Info Request Command

Table 3-7: Get Program Info Request Definition

Field	Description	Value	Default Value
Get Program Info - http://STBIP:port/tv/getProgInfo?major=num[&minor=num][&time=num]			
major	Major number of channel to tune to.	1...9999	
minor	Minor number of channel to tune to. Note: 65535 is used for no minor number	0...999, 65535	65535
time	Time of the program to query Note: Program returned must be a live TV program and the program's end time must be greater than the specified query time.	seconds since epoch time	current time since epoch time

3.2.2.2 Get Program Info JSON Response

Refer to Table 3-6: Get Tuned JSON Response Definition.

3.2.2.3 Get Program Info Sample Response

Refer to 3.2.1.3 Get Tuned Sample Response

3.2.3 Tune

3.2.3.1 Tune Request Command

Table 3-8: Tune JSON Request Definition

Field	Description	Value	Default Value
Tune - http://STBIP:port/tv/tune?major=num[&minor=num]			
major	Major number of channel to tune to.	1...9999	
minor	Minor number of channel to tune to. Note: 65535 is used for no minor number	0...999, 65535	65535



3.2.3.2 Tune JSON Response

Table 3-9: Tune JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			

3.2.3.3 Tune Sample Response

```
{
  "status": {
    "code": 200,
    "msg": "OK",
    "query": "/tv/tune?major=508"
  }
}
```



3.3 Remote Keys Functionality

3.3.1 Remote Keys

3.3.1.1 Remote Keys Request Command

Table 3-10: Remote Key Request Fields Definition

Field	Description	Value	Default Value
Remote Key - http://STBIP:port/remote/processKey?key=string[&hold=string]			
key	Name of the key to be simulated. Names accepted include: power, poweron, poweroff, format, pause, rew, replay, stop, advance, ffwd, record, play, guide, active, list, exit, back, menu, info, up, down, left, right, select, red, green, yellow, blue, chanup, chandown, prev, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, dash, enter	Fixed value remote control key name strings.	
hold	Simulate key being pressed, released, or both keyUp (simulate key being released only) keyDown (simulate key being pressed only) keyPress (simulate both press and release)	Fixed value command strings.	keyPress

3.3.1.2 Remote Key JSON Response

Table 3-11: Remote Key JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			
key	string	name of the simulated key	
Hold	string	holding status of the simulated key: keyUp, keyDown, or keyPress	

3.3.1.3 Remote Key Sample Response

```
{
  "hold": "keyPress",
}
```



```

"key": "info",
"status": {
  "code": 200,
  "msg": "OK",
  "query": "/remote/processKey?key=info&hold=keyPress"
}
}

```

3.4 Info Functionality

3.4.1 Get Version

3.4.1.1 Get Version Request Command

Table 3-12: Get Version Request Fields Definition

Field	Description	Value	Default Value
Get Version - http://STBIP:port/info/getVersion			
None	No fields are defined for this request.	N/A	N/A

3.4.1.2 Get Version JSON Response

Table 3-13: Get Version JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			
systemTime	number	System time in secs in UTC	
version	string	version of current implementation	

3.4.1.3 Get Version Sample Response

```

{
  "status": {
    "code": 200,
    "msg": "OK",

```



```

    "query": "/info/getVersion"
  },
  "systemTime": 1281625203,
  "version": "1.1"
}

```

3.4.2 Get Options

3.4.2.1 Get Options Request Command

Table 3-14: Get Options Request Fields Definition

Field	Description	Value	Default Value
Get Options - http://STBIP:port/info/getOptions			
None	No fields are defined for this request.	N/A	N/A

3.4.2.2 Get Options Response

The response is an HTML page that returns the list of available commands and its description.

Table 3-15: Get Options JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			
command	string	command string	
description	string	description of the command	
parameters [
{			
isRequired	boolean	whether the parameter is required	
name	string	name of the parameter	
type	string	type of the parameter	
} ...			
]			

3.4.2.3 Get Options Sample Response

```

"options": [

```




```
{
  "command": "/dvr/getPlayList",
  "description": "List of playable programs. Warning: This
command may change or be disabled in the future.",
  "parameters": [
    {
      "isRequired": false,
      "name": "wrapper",
      "type": "string"
    },
    {
      "isRequired": false,
      "name": "start",
      "type": "int"
    },
    {
      "isRequired": false,
      "name": "type",
      "type": "string"
    },
    {
      "isRequired": false,
      "name": "callback",
      "type": "string"
    },
    {
      "isRequired": false,
      "name": "max",
      "type": "int"
    }
  ]
},
{
  "command": "/dvr/play",
  "description": "Play a program. Warning: This command may
change or be disabled in the future.",
  "parameters": [
    {
      "isRequired": false,
      "name": "wrapper",
      "type": "string"
    },
    {
      "isRequired": false,
      "name": "offset",
      "type": "long"
    }
  ]
}
```



```
    },
    {
      "isRequired": false,
      "name": "callback",
      "type": "string"
    },
    {
      "isRequired": false,
      "name": "playFrom",
      "type": "string"
    },
    {
      "isRequired": true,
      "name": "uniqueId",
      "type": "long"
    }
  ]
},
...
],
"status": {
  "code": 200,
  "msg": "OK.",
  "query": "/info/getOptions?callback=jsonp"
}
}
```