DIRECTV SHEF Command Set

Published by

DIRECTV

DTV-MD-0359

(Rev. 1.3.C)

October 17, 2011

DIRECTV, Inc.

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
1.2.A	January 27, 2011	S.C. J. G.	Updated with SHEF spec v1.2DraftA
1.2.C	February 28, 2011	S.C. J. G.	Updated with SHEF spec v1.2DraftC
1.2.D	May 10, 2011	S.C. J. G.	Updated with SHEF spec v1.2DraftD: - Added notes about deprecating Get Playlist and Play; moved to separate section of document - Added web address of the latest SHEF command set documentation Updates to commands throughout
1.3.C	October 17, 2011	S.C.	Updated with SHEF spec v1.3DraftC: - Deprecated Get Playlist and Play; moved to separate section of document - Minor updates/clarifications to commands throughout - Added clientAddr fields for HR34 support - Added getLocations for HR34 support Removed “Beta” references

Table Of Contents

Section ***Page***

Table of Figures

List of Tables

1 Introduction

1.1 Terms of Use

1.1.1 Terms of Use and Legal Disclaimer

Except as otherwise provided in a separate written agreement signed by DIRECTV, these Terms of Use represent the entire understanding between you and DIRECTV regarding your limited use of this DIRECTV SHEF Command Set and the corresponding Set-Top Box HTTP Exported Functionality (“SHEF API specification”). DIRECTV is providing you access to the SHEF API specification subject to these Terms of Use. By accessing this SHEF API specification, you signify your acceptance of these Terms of Use and your acceptance of the obligations and Legal Disclaimer provided herein. If you do not agree to these Terms of Use, then do not use the SHEF API specification and you do not have any rights to use the SHEF API specification.

DIRECTV makes no representations or warranties of any kind, express or implied, that use of SHEF API specification or the technologies described in these SHEF API specification will not infringe any patents, copyrights, trade mark or other intellectual property rights of third parties. Nothing in this SHEF API 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. 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.

This SHEF API 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. This SHEF API specification is provided “as is” and the user of these specifications assumes any and all risks associated with the use of these specifications.

DIRECTV reserves the right, at any time, to terminate, discontinue, revise, modify, alter, update, or remove portions of the SHEF API specification and these Terms of Use, and DIRECTV expects to make revision, modifications, alteration and updates from time to time and you should periodically check to see that you are using the most current version of the SHEF API specification. If you are dissatisfied with these Terms of Use or the SHEF API specifications, your sole and exclusive remedy is to discontinue your use of the specifications.

If you provide DIRECTV with any comments or suggestions concerning the SHEF API specification ("Feedback"), you hereby agree that all Feedback is provided on a non-proprietary and non-confidential basis, and grant DIRECTV a non-exclusive, fully transferable, worldwide, perpetual, irrevocable license, with the right to sublicense through multiple levels of sublicensees, to incorporate, disclose, and use without limitation all Feedback for any purpose. You also grant to DIRECTV to modify, revise, update, change, alter and otherwise create derivative works of the Feedback and to fully exploit any and all rights to the Feedback.

You hereby agree to indemnify, defend and hold DIRECTV, and its officers, directors, employees, agents, licensors and licensees (collectively, the "Indemnified Parties") harmless from and against any and all liability and costs incurred by the Indemnified Parties in connection with any claim arising out of your use or misuse of the SHEF API specification, including without limitation, any and all reasonable attorneys' fees and court costs. DIRECTV reserves the right, at its own expense, to assume the exclusive defense and control of any claim subject to your indemnification obligation to the Indemnified Parties. You shall participate and cooperate as reasonably required in the defense of any such claim subject to your indemnification obligations.

1.1.2 Personal and Non-Commercial Use Only

Unless otherwise specified in writing by DIRECTV, the SHEF API specification is available for your individual, personal and non-commercial use only. You may not use, copy, distribute, transmit, reproduce, publish, license, create derivative works from, transfer or sell any information, products or services obtained through the SHEF API specification. Unless otherwise specified in writing by DIRECTV, other than only with the DIRECTV service, you may not use the SHEF API specification and any functionality provided thereby, directly or indirectly, with or in connection with any other multi-channel video distribution service, including but not limited to satellite, cable, broadcast, telecommunications, telephone, internet, wireless, or other video distribution system.

Any other use is prohibited, including copying, reproducing, republishing, uploading, posting, transmitting, editing or distributing SHEF API specification in any way by any means unless you have prior written permission from DIRECTV.

You shall not collect, use, disclose, rate, analyze, sort or otherwise mine any data or other information about DIRECTV customers, personal information and/or any usage or viewing information, specific or aggregated. You shall respect the privacy of any data or information regarding a DIRECTV customer, viewing habits, and recorded programs and comply with all applicable laws associated with such customer data and/or information.

All elements of the SHEF API specification are protected by copyright, moral rights, trademark and other laws relating to the protection of intellectual property.

1.2 Scope

This document provides information on 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 Updates

The most up-to-date version of this document can be found at:
http://satinstalltraining.com/homeautomation.html

Users of the SHEF command set are encouraged to check regularly for updates.

1.4 Feedback

DIRECTV requests and welcomes your feedback and suggestions on this 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 DIRECTV STB. 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. More information on networking DIRECTV STBs can be found at http://satinstalltraining.com or by contacting DIRECTV per the instructions in section 1.4.

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

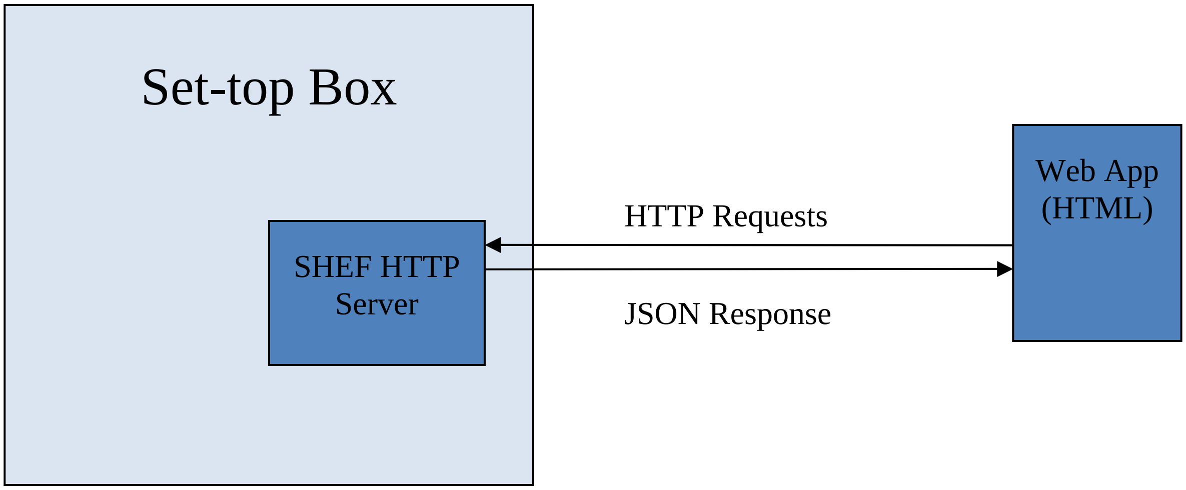


Figure 2-1: SHEF Data Flow

2.2 HTTP Implementation

2.2.1 SHEF (HTTP) Request

SHEF requests following the HTTP 1.0 or 1.1 specifications are accepted. The SHEF server expects requests to be UTF-8 encoded, with the exception of the file input part in a HTTP POST.

Version is a SHEF specific HTTP header that denotes the SHEF version of a request.

2.2.1.1 Get

The SHEF server implements a very basic HTTP GET method that ignores all HTTP request headers in the processing of SHEF requests, with the exception of the **Cookie** HTTP request header. Only a subset of the SHEF commands uses this header.

2.2.1.2 Post

The SHEF server's HTTP POST implementation only supports the form entity. No other entity types are supported. The **Content-Length** and **Content-Type** HTTP request headers must be present when a SHEF client initiates a HTTP POST SHEF request.

2.2.2 SHEF (HTTP) Response

SHEF responses are also UTF-8 encoded, with the exception of a SHEF response that is a file resource.

2.2.3 Forms

The SHEF server accepts form submissions via HTTP POST with the multipart/form-data content-type, as defined in RFC 2388. Refer to RFC 2388 for further details on the format definition.

The form controls supported are:

- Input elements
  - text
  - password (no encryption support)
  - checkbox (single and multiple selections)
  - radio buttons
  - file
  - hidden
- Select element (single and multiple selections)
- Text area element

Forms submitted using the default content type of application/x-www-form-url encoded is not supported.

The SHEF server implementation follows RFC 2388, with some minor deviations. There is no support for the **Content-Transfer-Encoding** header for each part. The default UTF-8 encoding is always used for all form element values (with the exception of the file input element where no encoding is applied). The **Content-Type** header is only recognized and used if the part is for a file input element. There is also no support for the multipart/mixed format.

Currently only the following content types are supported for a file input element (although more types may be added later):

- text/*
- image/*
- image/gif
- image/jpeg
- image/png

A file shall be rejected if it exceeds 25 MB.

2.2.4 Form Example

This is a basic HTTP POST SHEF request for a form that contains one (1) text input element and one (1) select element with multiple selection support.

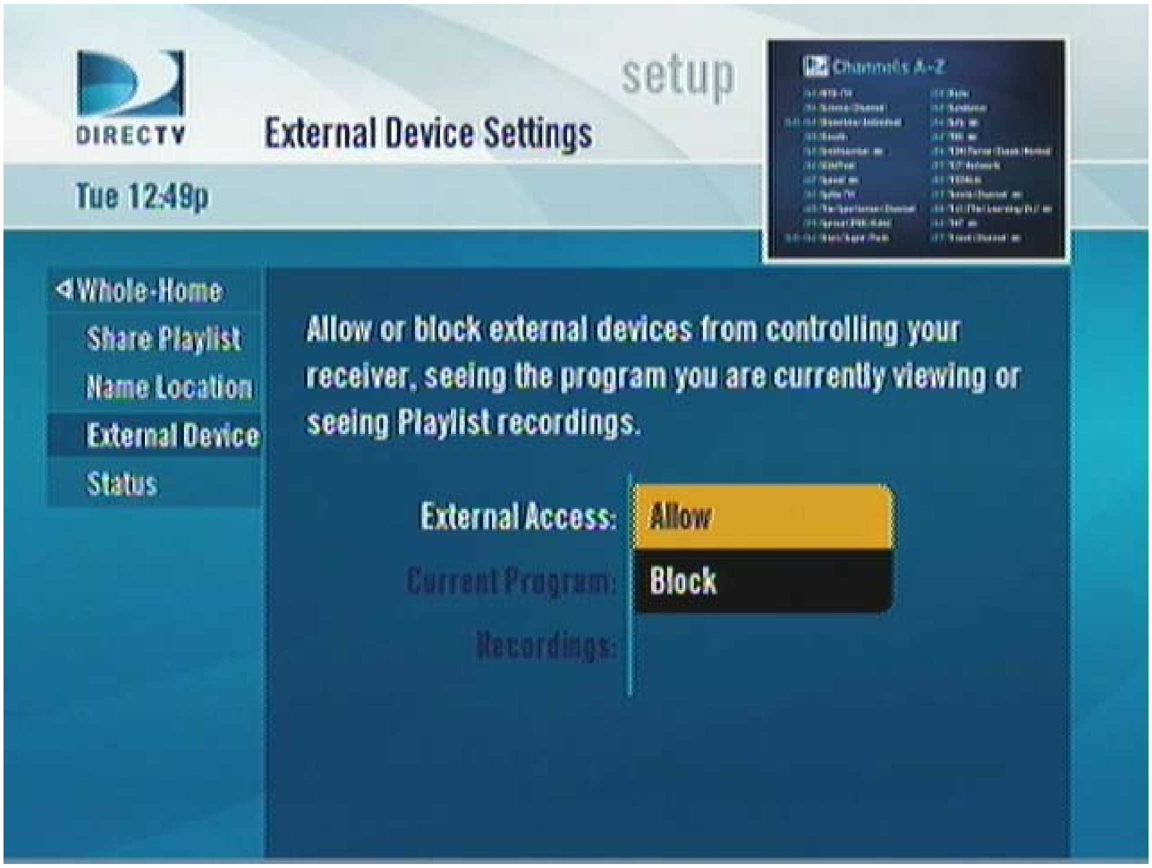
<table>
<tr><td colspan="2">POST /info/example?callback=jsonp HTTP/1.1</td></tr>
<tr><td>Host</td><td>192.168.1.125:8080</td></tr>
<tr><td>Content-Length</td><td>418</td></tr>
<tr><td>Content-Type</td><td>multipart/form-data; boundary=---------------------------5568188186258588041888136691</td></tr>
<tr><td colspan="2">-----------------------------5568188186258588041888136691[CRLF]
Content-Disposition: form-data; name="myText"[CRLF]
[CRLF]
Hello World![CRLF]
-----------------------------5568188186258588041888136691[CRLF]
Content-Disposition: form-data; name="mySelects"[CRLF]
[CRLF]
Arby’s[CRLF]
-----------------------------5568188186258588041888136691[CRLF]
Content-Disposition: form-data; name="mySelects"[CRLF]
[CRLF]
KFC[CRLF]
-----------------------------5568188186258588041888136691--[CRLF]</td></tr>
</table>

2.3 Network Security Settings

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

2.4 SHEF Opt In

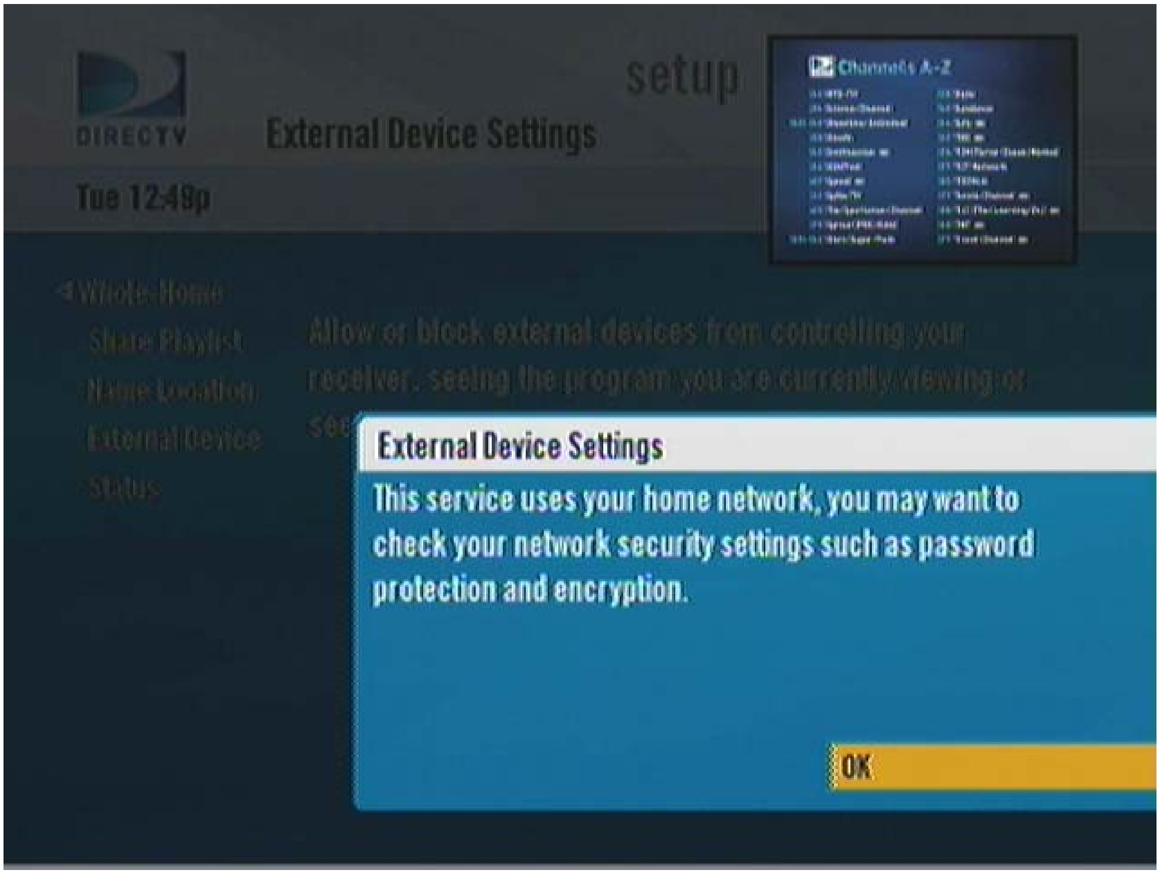
In order to use SHEF in your network, it must be enabled in the set-top box by navigating to the “External Device” settings screen (Menu->System Setup->Whole-Home->External Device) screen. Once on this screen select “Allow” for “External Access” as shown in Figure 2-2. Also turn on the other options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



Allow or block external devices from controlling your receiver, seeing the program you are currently viewing or seeing Playlist recordings.

External Access: Allow
Block
Current Program:
Recordings:

Figure 2-2: SHEF External Device Settings



Allow or block external devices from controlling your receiver, seeing the program you are currently viewing or see

External Device Settings

This service uses your home network, you may want to check your network security settings such as password protection and encryption.

Figure 2-3: SHEF External Device Home Network Settings

2.5 Supported Functionality

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

SHEF requests include the following major functionality:

2.5.1 DVR Functionality

- **Get Playlist –** Return a list of recordings in the Playlist. **Important note: This command is deprecated and will cease to function starting in SHEF version 1.3 (second half of 2011). The description for this command has been moved to section 4: Interface Functions – Deprecated Commands.**
- **Play –** Play a recorded program in the Playlist. **Important note: This command is deprecated and will cease to function starting in SHEF version 1.3 (second half of 2011). The description for this command has been moved to section 4: Interface Functions – Deprecated Commands.**

2.5.2 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.

2.5.3 REMOTE Functionality

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

2.5.4 INFO Functionality

- **Get Version –** Return the version of current implementations
- **Get Options –** Return a list of the available features (this list).
- **Mode –** Return the mode (active, standby) of the STB.
- **GetLocations –** Return an array of location name and client address. Note: Added for HR34.

2.6 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.
304	Not Modified.	No content has changed. No response body is included in the HTTP response.
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.

HTTP Status Code	Name	Description
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.

2.7 Limitations

SHEF provides sufficient responsiveness to be acceptable to a person controlling the STB. Sub-second responses are desired, but millisecond response time is not required.

SHEF is single-threaded and will not directly queue requests. This will provide some degree of serialization, but the execution of requests in a particular order is not guaranteed as http does not guarantee requests will arrive in any particular order.

2.8 Protocol Definition

Table 2-2: STB Http Exported Functionality Request Syntax

STB Http Exported Functionality request syntax
http://127.0.0.1:8080/info/function?param1=*value*[&param2=*value*] In the interest of efficiency, using the well known hard coded IP address when possible is preferred over the equally well known name of the adapter. So for instance: **http://127.0.0.1:8080/dvr/play?uniqueId=*num*** Would be preferred over: **http://localhost:8080/dvr/play?uniqueId=*num*** Future expansion may require specification of an IP address outside of the STB. 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. * *Italic* words are string variables representing the actual data. See STB Http Exported Functionality request Fields Definition for definitions of the available requests.

3 Interface Functions

3.1 TV Functionality

3.1.1 Get Tuned

3.1.1.1 Get Tuned Request Command

Table 3-1: Get Tuned Request Definition

Field	Description	Value	Default Value
Get Tuned - http://STBIP:port/tv/getTuned?[clientAddr=string][&videoWindow=string]			
clientAddr	Identifies the server or client. Note: Added for HR34.	0 for server and mac address (hexstring without colons) for clients	0
videoWindow	Identifies the whether primary or secondary window. Note: Added for HR34.	Primary or secondary window	primary

3.1.1.2 Get Tuned JSON Response

Table 3-2: 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	
commandResult	number	result of the command request 0: success, non zero: error	
msg	string	OK: success, otherwise an error message is returned	
}			
stationId	number	unique identifier for the channel	
programId	number	program object id	
materialId	number	material ID 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	

Name	Type	Description	Optional
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	
date	string	release year or first released date of the event for single event or first air date for series	✓
title	string	title of the program	
episodeTitle	string	title of the episode	✓
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	✓
expiryTime	number	expiration day & time of the event in UTC seconds	✓
recType	number	only available if the event is a recording; type of recordings; 1: manual recording 2: findby recording 3: regular recording 4. recurring manual	✓
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;	✓

Name	Type	Description	Optional
		format: n of total priorities	
music {		only available if the event is a music channel music sound track information	
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 or program start time of a live tv or live buffer	

3.1.1.3 Get Tuned Sample Response

```
{
  "callsign": "FOODHD",
  "date": "20070324",
  "duration": 1791,
  "episodeTitle": "Spaghetti and Clam Sauce",
  "expiration": "0",
  "expiryTime": 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,
    "commandResult": 0,
    "msg": "OK.",
    "query": "/tv/getTuned"
  },
  "title": "Tyler's Ultimate",
  "uniqueId": "6728716739474078694"
}
```

3.1.2 Get Program Info

3.1.2.1 Get Program Info Request Command

Table 3-3: Get Program Info Request Definition

Field	Description	Value	Default Value
Get Program Info - http://STBIP:port/tv/getProgInfo?major=num[&minor=num][&time=num][&clientAddr=string]			
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 up to 2 hours from current time	current time since epoch time
clientAddr	Identifies the server or client. Note: Added for HR34.	0 for server and mac address (hexstring without colons) for clients	0

3.1.2.2 Get Program Info JSON Response

Refer to Table 3-2: Get Tuned JSON Response Definition.

3.1.2.3 Get Program Info Sample Response

Refer to 3.1.1.3 Get Tuned Sample Response

3.1.3 Tune

3.1.3.1 Tune Request Command

Table 3-4: Tune JSON Request Definition

Field	Description	Value	Default Value

Field	Description	Value	Default Value
Tune - http://STBIP:port/tv/tune?major=num[&minor=num][&clientAddr=string]			
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
clientAddr	Identifies the server or client. Note: Added for HR34.	0 for server and mac address (hexstring without colons) for clients	0

3.1.3.2 Tune JSON Response

Table 3-5: Tune JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
commandResult	number	result of the command request 0: success, non zero: error	
msg	string	OK: success, otherwise an error message is returned	
}			

3.1.3.3 Tune Sample Response

```
{
  "status": {
    "code": 200,
    "commandResult": 0,
    "msg": "OK",
    "query": "/tv/tune?major=508"
  }
}
```

In the case of conflict:

```
{
"status": {
  "code": 500,
  "commandResult": 1,
  "msg": " Request conflict.",
  "query": "/tv/tune?major=5"
}
```

3.2 Remote Keys Functionality

3.2.1 Remote Keys

3.2.1.1 Remote Keys Request Command

Table 3-6: Remote Key Request Fields Definition

Field	Description	Value	Default Value
Remote Key - http://STBIP:port/remote/processKey?key=string[&hold=string][&clientAddr=string]			
key	Name of the key to be simulated. Supported keys: 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
clientAddr	Identifies the server or client. Note: Added for HR34.	0 for server and mac address (hexstring without colons) for clients	0

3.2.1.2 Remote Key JSON Response

Table 3-7: 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	
commandResult	number	result of the command request 0: success, non zero: error	
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	

3.2.1.3 Remote Key Sample Response

```
{
  "hold": "keyPress",
  "key": "info",
  "status": {
    "code": 200,
    "commandResult": 0,
    "msg": "OK",
    "query": "/remote/processKey?key=info&hold=keyPress"
  }
}
```

3.3 Info Functionality

3.3.1 Get Version

3.3.1.1 Get Version Request Command

Table 3-8: 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

3.3.1.2 Get Version JSON Response

Table 3-9: 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	
commandResult	number	result of the command request 0: success, non zero: error	
msg	string	OK: success, otherwise an error message is returned	
}			
accessCardId	string	Access card ID (xxxx-xxxx-xxxx)	

receiverId	string	Receiver ID (xxxx xxxx xxxx)	
stbSoftwareVersion	string	Current STB software version in hex string	
systemTime	number	System time in secs in UTC	
version	string	version of current implementation	

3.3.1.3 Get Version Sample Response

```
{
  "accessCardId": "0021-1495-6572",
  "receiverId": "0288 7745 5858",
  "status": {
    "code": 200,
    "commandResult": 0,
    "msg": "OK",
    "query": "/info/getVersion"
  },
  "stbSoftwareVersion": "0x4ed7",
  “systemTime”: 1281625203,
  "version": "1.2"
}
```

3.3.2 Get Options

3.3.2.1 Get Options Request Command

Table 3-10: 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

3.3.2.2 Get Options Response

The response returns the list of available commands and its description.

Table 3-11: 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	
commandResult	number	result of the command request 0: success, non zero: error	
msg	string	OK: success, otherwise an error message is returned	
}			
command	string	command string	
deprecated	boolean	true if deprecated	✓
description	string	description of the command	
formControls [	string		
{			
isRequired	boolean	whether the parameter is required	
name	string	name of the control type	
type	string	type of the control	
} …			
]			
urlParameters [			
{			
isRequired	boolean	whether the parameter is required	
name	string	name of the parameter	
type	string	type of the parameter	
} …			
]			

3.3.2.3 Get Options Sample Response

```
{
  "options": [
    {
      "command": "/info/getLocations",
      "description": "List of available client locations. Warning: This command
may change or be disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        }
      ]
    },
    {
      "command": "/info/getVersion",
      "description": "Set-top-box and SHEF information. Warning: This command
may change or be disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        }
      ]
    },
    {
      "command": "/info/mode",
      "description": "Set-top-box mode. Warning: This command may change or be
disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "clientAddr",
          "type": "string"
        }
      ]
    },
    {
      "command": "/remote/processKey",
      "description": "Process a key request from the remote control. Warning: 
This command may change or be disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "clientAddr",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "hold",
          "type": "string"
        },
        {
          "isRequired": true,
          "name": "key",
          "type": "string"
        }
      ]
    },
    {
      "command": "/serial/processCommand",
      "description": "Process a command request from remote control. Warning:
This command may change or be disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        },
        {
          "isRequired": true,
          "name": "cmd",
          "type": "hex"
        }
      ]
    },
    {
      "command": "/tv/getProgInfo",
      "description": "Program information of specified channel at current or
```

```
specific time. Warning: This command may change or be disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "minor",
          "type": "int"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        },
        {
          "isRequired": true,
          "name": "major",
          "type": "int"
        },
        {
          "isRequired": false,
          "name": "clientAddr",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "time",
          "type": "long"
        }
      ]
    },
    {
      "command": "/tv/getTuned",
      "description": "Information about the currently viewed program. Warning:
This command may change or be disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "videoWindow",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "clientAddr",
          "type": "string"
        }
      ]
    },
    {
      "command": "/tv/tune",
      "description": "Tune to a channel. Warning: This command may change or be
disabled in the future.",
      "formControls": [],
      "urlParameters": [
        {
          "isRequired": false,
          "name": "wrapper",
          "type": "string"
        },
        {
          "isRequired": false,
          "name": "minor",
          "type": "int"
        },
        {
          "isRequired": false,
          "name": "callback",
          "type": "string"
        },
        {
          "isRequired": true,
          "name": "major",
          "type": "int"
        },
        {
          "isRequired": false,
          "name": "clientAddr",
          "type": "string"
        }
      ]
    }
  ],
  "status": {
    "code": 200,
    "commandResult": 0,
    "msg": "OK.",
    "query": "/info/getOptions"
  }
}
```

3.3.3 Mode

3.3.3.1 Mode Request Command

Table 3-12: Mode Request Fields Definition

Field	Description	Value	Default Value
Mode - http://STBIP:port/info/mode?[clientAddr=string]			
clientAddr	Identifies the server or client. Note: Added for HR34.	0 for server and mac address (hexstring without colons) for clients	0

3.3.3.2 Mode JSON Response

Table 3-13: Mode JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
commandResult	number	result of the command request 0: success, non zero: error	
msg	string	OK: success, otherwise an error message is returned	
}			
mode	number	0 – active 1 – standby	

3.3.3.3 Mode Sample Response

```
{
  "mode": 0
  "status": {
    "code": 200,
    "commandResult": 0,
    "msg": "OK",
    "query": "/info/mode"
  }
}
```

3.3.4 GetLocations

3.3.4.1 GetLocations Request Command

Table 3-14: GetLocations Request Fields Definition

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

3.3.4.2 GetLocations JSON Response

Table 3-15: GetLocations JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
code	number	Refer to Table 2-1 HTTP Status Codes	
commandResult	number	result of the command request 0: success, non zero: error	
msg	string	OK: success, otherwise an error message is returned	
}			
locations [			
{			
clientAddr	string	0 for server and mac address (hexstring without colons) for clients	
locationName	string	location name	
}]			

3.3.4.3 GetLocations Sample Response

```
  "locations": [{
    "clientAddr": "0",
    "locationName": "A"
  }],
  "status": {
    "code": 200,
    "commandResult": 0,
    "msg": "OK.",
    "query": "/info/getLocations?callback=jsonp"
  }
}
```

4 Interface Functions – Deprecated Commands

Important note: The commands listed in this section are deprecated in SHEF version 1.3 (second half of 2011) and higher. These commands will also no longer be returned from the getOptions command.

4.1 DVR Functionality

4.1.1 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.

4.1.1.1 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 4-1: Get Playlist Request Fields Definition

Field	Description	Value	Default Value
Get Playlist – http://STBIP:port/dvr/getPlayList?[start=num][&max=num][&type=string][&clientAddr=string]			
start	Index (starting at zero) of first recorded item	1…count of recorded item	0
max	Maximum number of playlist items to return	1…25	25
type	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
clientAddr	Identifies the server or client. Note: Added for HR34.	0 for server and mac address (hexstring without colons) for clients	0

4.1.1.2 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 4-2: 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	
commandResult	number	result of the command request 0: success, non zero: error	
msg	string	OK: success, otherwise an error message is returned	
}			
hasChanged	boolean	An indicator that the user’s playlist has added or removed recordings	✓
isPclocked	number	whether the parental control is enabled; 1: locked, 2: temporarily unlocked, 3: unlocked	
isRemoteDeleteEnabled	boolean	whether the recorded programs in this HD DVR are allowed to be deleted remotely	
isSharedPlayListEnabled	boolean	whether the playlist in this HD DVR is shared with other HD receivers	
itemsReturned	number	number of items returned in this 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 [			
{			
programId	number	program object id	
uniqueId	string	unique identifier of the event	
materialId	string	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	
date	string	release year or first released date of the event for single event or first air date for series	✓
episodeTitle	string	title of the episode	✓
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	
expiryTime	number	expiration day & time of the event in UTC seconds	
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	
}] …			
} …			
]			

4.1.1.3 Get Playlist Sample Response

```
{
  "hasChanged": false,
  "isPclocked": 3,
  "isRemoteDeleteEnabled": true,
  "isSharedPlayListEnabled": true,
  "itemsReturned": 25,
  "playList": [
    {
```

```
    "groupIndex": 0
    "groupTitle": "GIADA",
    "isRecording": false,
    "numItems": 2,
    "numNewItems": 2,
    "recordings": [
      {
        "callsign": "COOK",
        "date": "20090606",
        "duration": 1800,
        "episodeTitle": "Around the World",
        "expiration": "0",
        "expiryTime": 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",
        "date": "20090606",
        "duration": 245,
        "episodeTitle": "Giada's Bake Sale",
        "expiration": "0",
        "expiryTime": 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,
    "commandResult": 0,
    "msg": "OK.",
    "query": "/dvr/getPlayList"
  }
  "totalNumItems": 30,
  "type": "user"
}
```

4.1.2 Play

4.1.2.1 Play Request Command

Table 4-3: Play Request Fields Definition

Field	Description	Value	Default Value
Play – http://STBIP:port/dvr/play?uniqueId=num[&playFrom=string][&offset=num][&clientAddr=string][&udn =string]			

Field	Description	Value	Default Value
Play – http://STBIP:port/dvr/play?uniqueId=num[&playFrom=string][&offset=num][&clientAddr=string][&udn =string]			
uniqueId	Begin playback of recorded program where *uniqueId* 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
clientAddr	Identifies the server or client. Note: Added for HR34.	0 for server and mac address (hexstring without colons) for clients	0
udn	Unique device number to uniquely identify a STB to play MRV recording.	String	

4.1.2.2 Play JSON Response

Table 4-4: Play JSON Response Definition

Name	Type	Description	Optional
status {			
query	string	incoming query string	
commandResult	number	result of the command request 0: success, non zero: error	
code	number	Refer to Table 2-1 HTTP Status Codes	
msg	string	OK: success, otherwise an error message is returned	
}			

Note: For play MRV, there will be no error returned if the uniqueId, offset, or udn is invalid. If the playback for MRV recording is unsuccessful, there will be no error returned.

4.1.2.3 Play Sample Response

```
{ "status": {
  "code": 200,
  "commandResult": 0,
  "msg": "OK",
  "query": "/dvr/play?uniqueId=2873298685953728822"
}}
```