# VOD Interface Part 10b

# Recorder Status Interface

# History

| Version | Status | Date | Author | Description |
|---|---|---|---|---|
| 1.0 | Proposed | 2013-Feb-21 | Kevin Farrington | |

## Document Statuses

This document can have one of the following statuses:

| | |
|---|---|
| **DRAFT** | Document is draft. |
| **PROPOSAL** | Document is ready for review. |
| **RELEASED** | Document has been reviewed and released by SeaChange development. |

## Document Version Numbering

This document has unique version numbers to unique states of the document. The version number is expressed as a *major.minor* number pair. These numbers which start at zero, are assigned in increasing order and correspond to new developments of the document. Whereby the minor number is used to convey maintenance type of changes and the major one conveys new document releases that has been gone thru a review-release cycle.

## Document Change Procedure

Changes to this specification will result in a new version of this document. When this document has either the DRAFT or, PROPOSAL status, change requests must be sent to the author. In case the document has RELEASED status, change requests must be submitted to the *Change Control Board* (CCB) of SeaChange development.

SeaChange®

# Contents

# 1 Introduction

## 1.1 Purpose of the document

To record linear programming, Video On Demand (VOD) systems must discover and maintain information about video recording devices in a deployment. This information consists of device capabilities, topology information and current status. The purpose of this document is to describe a generic recorder status interface enabling for use by video recorder vendors to integrate their devices with the VOD recording system.

## 1.2 Scope

This document deals only with the status reporting interface between a video recorder related to the VOD infrastructure. The recorder control interface is described separately in the IF-3d specification. Other interfaces for the recorder, such as the linear broad feeds or connectivity to storage are out of scope for this interface.

## 1.3 Changes

The interfaces in this document are specified as accurately as possible at the time of the writing. Any changes to the interfaces must be mutually agreed between SeaChange and the customer. SeaChange keeps the right to improve the interfaces. Suppliers using the interfaces will be notified of such changes.

## 1.4 Conventions

The following conventions are applicable in this document:

- The word *shall* is used to indicate mandatory requirements strictly to be followed and from which no deviation is permitted *(shall* equals *is required to)*.
- The word *should* is used to indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others; or that a certain course of action is preferred but not necessarily required; or that (in the negative form) a certain course of action is deprecated but not prohibited *(should* equals *is recommended that)*.
- The word *must* is used only to describe unavoidable situations.
- The word *will* is only used in statements of fact.
- The word *may* is used to indicate a course of action permissible within certain limits *(may* equals *is permitted to)*.
- The word *can* is used for statements of possibility and capability, whether material, physical, or causal *(can* equals *is able to)*.

## 1.5 Standards

Date time formats between the systems are ALWAYS in UTC time and W3C (ISO 8601 profile) formatting, e.g.: *2004-11-05T13:15:30Z*, unless specifically stated otherwise.

This way time discontinuities can be avoided (i.e. daylight savings). Note that all interfacing systems must decode/encode the date time to the correct local time.

## 1.6 References

This section lists the references made in this document.

### 1.6.1 Normative References

[HTTP] R. Fielding et al., Hypertext Transfer Protocol - HTTP 1.1, June 1999, http://tools.ietf.org/html/rfc2616

[IF-3d] VOD Interface Part 3d - Content propagation interface, Version 1.5, January 16, 2013.

[IF-5d] VOD Interface Part 5d - STB - Traxis.SRM - VideoServer , Version 1.4, January 10, 2013.

### 1.6.2 Informative References

[REST] Representational State Transfer – Architectural Styles and the Design of Network-based Software Architectures – Roy Fielding

http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

## 1.7 Open Issues

# 2 Introduction

The recorder status interface specifies the discovery and status interface used by video recorders to describe their existence, status and capabilities to the VOD system.

Video recorders periodically send status information, in the form of recorder status messages, to the VOD system over HTTP protocol connections. Each status message is an XML-encoded document with an XML Schema described syntax as described in Appendix B.

The Recorder Status interface relies on a idealized model of a video recorder as shown in Figure 1.

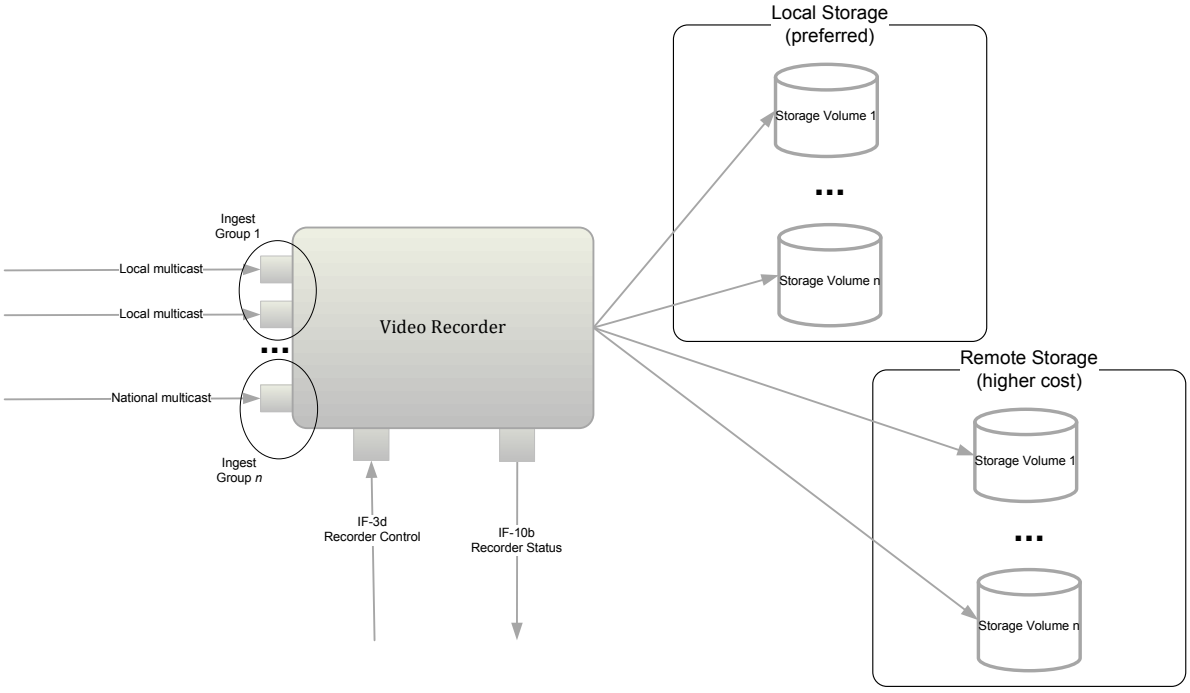


**FIGURE 1 Video Recorder Model**

The VOD system views a video recorder as a device that is capable of receiving linear streams and creating recordings, timed portions of the stream saved to long-term storage for later on-demand play back.

The linear streams may be received over one or more network interfaces; if multiple network interfaces are supported, there may be multiple distinct networks connected to the interfaces. The recorder must manage the bandwidth and ingest capacity over these interfaces, ensuring that resources are not overcommitted.

Recordings are controlled over the IF-3d interface which is used to schedule individual recordings, determine the status of a recording and delete recordings. Recordings are always identified by a back-office defined unique content identifier. When asked to initiate a recording, the recorder will commit ingest and storage resources to the recording and prepare to start recording. The recording start time and end time are passed to the recorder; the recorder is expected to manage the recording process by starting and stopping at the appropriate times. A recorder must be time synchronized with an authoritative time source, typically over a

protocol such as NTP. At any time, the back-office may attempt to adjust the recording schedule for a recording - for example, to set the end time to "now" to prematurely terminate a recording.

Recordings are constructed by accessing a linear stream over one of the ingest network interfaces. To ensure that a particular linear stream is accessible to a video recorder, the recorder's inputs are placed into ingest groups. All members of an ingest group have access to the same linear streams. An ingest group may span individual recorders and may include multiple network interfaces on a single recorder.

Recordings are placed on storage volumes. The back-office is responsible for choosing the placement of recordings onto storage volumes based on network topology to the customer's devices. A recorder may have the ability to write to multiple storage volumes. In order to support different network topologies for the recorder's connectivity to storage, the recorder may provide a "cost" factor indicating the relative expense of it using a particular volume. For example, a recorder may have the ability to store content locally or on a more expensive regional library. The recorder may tell the back-office that the regional library has a higher cost and should only be used when it is not viable to use the local store.

Some recorders have additional capabilities such as the ability to transcode recorded content, package recorded content for adaptive bit rate playback, and integrate with DRM systems to encrypt the content. The Recording Status Interface enables recorders to describe these capabilities to the VOD system.

# 3 Recorder Status interface

The Recorder Status interface is used by video recorders to provide information to the VOD back-office to enable it to make informed decisions when choosing a video recorder to handle a recording task.

## 3.1 Interface binding

The interface binding uses HTTP POST requests periodically initiated by video recorders conveying status messages in an XML document conveyed in the request's entity body.

### 3.1.1 HTTP Protocol Usage

All participants in this messaging interface must support HTTP 1.1 as defined in [HTTP]**Error! Reference source not found.**. The VOD back-office will act in the HTTP server role accepting connections from Video Recorder clients.

All protocol messages conveyed in HTTP entity bodies must use the **application/xml** type in the HTTP *Content-Type* header. All responses that include a body must specify the HTTP *Content-Length* header, unless chunked transfer coding is being used. The use of the server dropping the underlying TCP connection to indicate the end of a message, as was required for HTTP 1.0, shall not be used.

All protocol messages conveyed over this interface shall be encoded with the UTF-8 character set.

### 3.1.2 Response handling

The response on a HTTP request is an HTTP status codes according to the “Success” or “Client Error” classes (i.e. status codes 2xx, 4xx or 5xx) of status codes.

The following HTTP status codes in a HTTP response are used in the online reporting interface:

| Code | Definition | Description |
|---|---|---|
| 200 | OK | The request has been successfully executed and optionally an XML structure is returned with the requested information |
| 302 | Redirect | The server has redirected the request to an alternate server URL. The new URL is conveyed in the HTTP *Location* header. |
| 400 | Bad Request | The request contains invalid syntax or cannot be fulfilled |
| 403 | Forbidden | The server understood the request but is refusing to fulfil it. |
| 404 | Not Found | The resource of the request does not exist in the database |
| 500 | Internal Server Error | The request caused an internal server error |

## 3.2 Protocol Messages

### 3.2.1 HTTP Endpoint

All messages supported by the Recorder Status Interface are sent in HTTP POST requests to a single URL. The syntax of the URL is:

http[s]://<host>:<port>/<prefix path>/RecorderStatus

where:

host - is the DNS name or numeric address of the VOD system. Any numeric IPv6 addresses must be enclosed in square brackets (e.g. [::1]) per RFC-2732.

port - the listening endpoint on the VOD system. (e.g. 80 or 8080)

prefix path - an implementation defined path to the recording status interface endpoint. This is deployment specific and must be configurable on the client.

### 3.2.2 RecorderStatusUpdate Request

The RecorderStatusUpdate message is used to define or update the VOD system's view of a video recorder. The Video Recorder should compose and post this message to the VOD system:

- When a recorder completes its boot/initialization to let the VOD system know that it is in service.
- When significant changes occur to the status of the recorder, such as going out of service, losing connectivity to a storage volume, topology changes, etc.
  Periodically to updated the VOD system with the status of the recorder. This reporting period shall not be more frequent than every 30 seconds.

The following figure presents the structure of the *RecorderStatusUpdate* request.

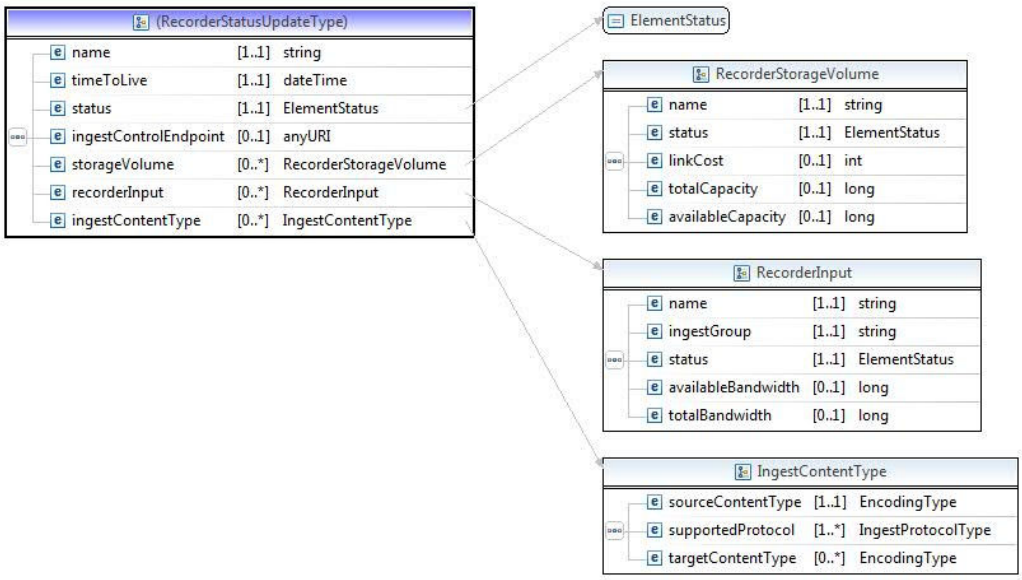

**Figure 2: RecorderStatusUpdate message**

| Field | Description | Req. |
|---|---|---|
| name | The recorder name. The recorder name must uniquely identify each recorder in the deployment.<br>Maximum of 128 characters. | Yes |
| timeToLive | The time to live field defines an expiration date for the information in this message. If an additional update is not received before this message expires, the VOD system shall assume that the recorder is going out of service and has transitioned to the "draining" status. | Yes |
| status | The status of the recorder. This field may contain one of the following values:<br>➢ **up** - indicating that the recorder is currently in service and accepting new requests.<br>➢ **draining** - the recorder is in the process of going out of service. No new requests should be sent to the recorder, but any active recordings will complete.<br>➢ **offline** - the recorder is out of service. New recording tasks will not be accepted and any in-process recording tasks have been terminated. | Yes |
| ingestControlEndpoint | The HTTP URL used to send recording control requests to this recorder over the IF-3d interface.<br>This field is not required but must be provided when the status field indicates that the recorder is in the "up" state. | No |
| storageVolume | A list of RecorderStorageVolume elements describing the storage volume connectivity of the recorder. | No |
| recorderInput | A list of RecorderInput elements describing each recorder input. | No |
| ingestContentType | A list of IngestContentType elements describing the types of content that the recorder can handle. | No |

The RecorderStorageVolume element has the following structure:

| Field | Description | Req. |
|---|---|---|
| name | The name of the storage volume. Storage volume names are unique deployment-wide and identify the volume in recording (IF-3d) and playback (IF-5d) operations.<br>Maximum of 128 characters. | Yes |
| status | The status of the storage volume's connectivity to this recorder. This field may contain one of the following values:<br>➢ **up** - indicating that the recorder is currently able to access and use the storage volume to store new recordings.<br>➢ **draining** - the recorder will not use the storage volume for any new recording requests, but any recording requests that have already been accepted and are in-progress are able to complete.<br>➢ **offline** - the recorder is no longer able to access the storage volume. | Yes |
| linkCost | An non-negative, signed 32-bit integer value representing the relative cost of using the storage volume for recording. A value of 0 represents a the lowest cost and is the default value if this field is not specified. | No |
| totalCapacity | The total storage capacity of the storage volume, in bytes. | No |
| availableCapacity | The current available capacity of the storage volume, in bytes. | No |

The RecorderInput element has the following structure:

| Field | Description | Req. |
|---|---|---|
| name | The name of the recorder input. The recorder input name is uniquely identifies the network input within the recorder, but is not unique across recorders.<br>Maximum of 128 characters. | Yes |
| ingestGroup | The name of the ingest group that this network interface is a member of. The ingest group name uniquely identifies the group deployment-wide and is used to identify the linear streams that are accessible to the recorder's input.<br>Maximum of 128 characters. | Yes |

| Field | Description | Req. |
|---|---|---|
| status | The status of the recorder's input. This field may contain one of the following values:<br>➢ **up** - indicating that the input is currently available to receive and record linear streams.<br>➢ **draining** - the recorder input is being gracefully take out of service. No new recordings will be accepted over the interface, but in-progress recordings will be able to drain gracefully.<br>➢ **offline** - the recorder is not able to use this input. | Yes |
| totalBandwidth | The total ingest bandwidth of the input, in bits/second. | No |
| availableBandwidth | The currently available bandwidth of the input, in bits/second. | No |

The IngestContentType element has the following structure:

| Field | Description | Req. |
|---|---|---|
| sourceContentType | The type of encoding type supported by this recorder. | Yes |
| supportedProtocol | A list of one or more protocols that can be used to ingest the conent type. Value values include:<br>➢ **ftp** - an FTP copy from a source volume.<br>➢ **fxp** - a server-to-server FTP transfer controlled.<br>➢ **cifs** - ingest from a CIFS file share<br>➢ **nfs** - copy from an NFS file share<br>➢ **nfs link** - a special case of NFS where the intest content and recorded output reside on the same origin server. NFS links are used to "copy" the content.<br>➢ **multicast** - ingest by tuning an MPEG-2 SPTS multicast. | Yes |
| targetContentType | A list of encoding types that the recorder is capable of generating from the source. This is used to convey various transcoding, packaging and DRM capabilities of the recorder. | No |

# Appendix A. Recorder status usage examples

This appendix contains some examples how the recorder status interface is used.

## 3.3 Simple recorder status update

```
POST <prefix path>/RecorderStatus HTTP/1.1
Content-Type: application/xml;charset=utf-8
Content-Length: 2064

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RecorderStatusUpdate xmlns="http://seachange.tv/adrenalin/recorderStatus"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <name>Acton.EngLab.Recorder23</name>
    <timeToLive>2012-02-23T17:41:05.484Z</timeToLive>
    <status>up</status>

     <ingestControlEndpoint>http://[2013:0222:1001::1017]:8080/ContentPropagationInterf
     ace</ingestControlEndpoint>
    <storageVolume>
        <name>Acton.EngLab.RecordingVolume.6</name>
        <status>up</status>
        <linkCost>0</linkCost>
        <totalCapacity>21685429525298</totalCapacity>
        <availableCapacity>10842714762649</availableCapacity>
    </storageVolume>
    <storageVolume>
        <name>Acton.EngLab.RecordingLibrary.1</name>
        <status>up</status>
        <linkCost>100</linkCost>
    </storageVolume>
    <recorderInput>
        <name>eth0</name>
        <ingestGroup>Acton.EngLab.Local</ingestGroup>
        <status>up</status>
        <availableBandwidth>10000000000</availableBandwidth>
        <totalBandwidth>0</totalBandwidth>
    </recorderInput>
    <recorderInput>
        <name>eth1</name>
        <ingestGroup>Acton.EngLab.Local</ingestGroup>
        <status>up</status>
        <availableBandwidth>10000000000</availableBandwidth>
        <totalBandwidth>0</totalBandwidth>
    </recorderInput>
    <recorderInput>
```

```
        <name>eth2</name>
        <ingestGroup>Acton.EngLab.National</ingestGroup>
        <status>up</status>
        <availableBandwidth>10000000000</availableBandwidth>
        <totalBandwidth>0</totalBandwidth>
    </recorderInput>
    <ingestContentType>
        <sourceContentType>mpeg2</sourceContentType>
        <supportedProtocol>multicast</supportedProtocol>
    </ingestContentType>
    <ingestContentType>
        <sourceContentType>mpeg4</sourceContentType>
        <supportedProtocol>multicast</supportedProtocol>
        <targetContentType>mpeg4-hls</targetContentType>
        <targetContentType>mpeg4-hls-aes</targetContentType>
    </ingestContentType>
</RecorderStatusUpdate>
```

## 3.4 Record going out of service

```
POST <prefix path>/RecorderStatus HTTP/1.1
Content-Type: application/xml;charset=utf-8
Content-Length: 355

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<RecorderStatusUpdate xmlns="http://seachange.tv/adrenalin/recorderStatus"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <name>Acton.EngLab.Recorder23</name>
    <timeToLive>2012-02-23T17:46:34.130Z</timeToLive>
    <status>draining</status>
</RecorderStatusUpdate>
```

# Appendix B. REST services XML schemas

The following XSD describes the recorder status XML schema:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="http://seachange.tv/adrenalin/recorderStatus" elementFormDefault="qualified"
 xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://seachange.tv/adrenalin/recorderStatus">

 <xs:element name="RecorderStatusUpdate">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="name" type="xs:string" maxOccurs="1" minOccurs="1" />
    <xs:element name="timeToLive" type="xs:dateTime" maxOccurs="1" minOccurs="1" />
    <xs:element name="status" type="ElementStatus" maxOccurs="1" minOccurs="1" />
    <xs:element name="ingestControlEndpoint" type="xs:anyURI" maxOccurs="1" minOccurs="0" />
    <xs:element name="storageVolume" type="RecorderStorageVolume" maxOccurs="unbounded" minOccurs="0" />
    <xs:element name="recorderInput" type="RecorderInput" maxOccurs="unbounded" minOccurs="0" />
    <xs:element name="ingestContentType" type="IngestContentType" maxOccurs="unbounded" minOccurs="0" />
   </xs:sequence>
  </xs:complexType>
 </xs:element>

 <xs:complexType name="RecorderStorageVolume">
  <xs:sequence>
   <xs:element name="name" type="xs:string" maxOccurs="1" minOccurs="1" />
   <xs:element name="status" type="ElementStatus" maxOccurs="1" minOccurs="1" />
   <xs:element name="linkCost" type="xs:int" maxOccurs="1" minOccurs="0" />
   <xs:element name="totalCapacity" type="xs:long" maxOccurs="1" minOccurs="0" />
   <xs:element name="availableCapacity" type="xs:long" maxOccurs="1" minOccurs="0" />
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="RecorderInput">
  <xs:sequence>
   <xs:element name="name" type="xs:string" maxOccurs="1" minOccurs="1" />
   <xs:element name="ingestGroup" type="xs:string" maxOccurs="1" minOccurs="1" />
   <xs:element name="status" type="ElementStatus" maxOccurs="1" minOccurs="1" />
   <xs:element name="availableBandwidth" type="xs:long" maxOccurs="1" minOccurs="0" />
   <xs:element name="totalBandwidth" type="xs:long" maxOccurs="1" minOccurs="0" />
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="IngestContentType">
  <xs:sequence>
   <xs:element name="sourceContentType" type="EncodingType" maxOccurs="1" minOccurs="1" />
   <xs:element name="supportedProtocol" type="IngestProtocolType" maxOccurs="unbounded" minOccurs="1"/>
   <xs:element name="targetContentType" type="EncodingType" maxOccurs="unbounded" minOccurs="0" />
  </xs:sequence>
 </xs:complexType>

 <xs:simpleType name="ElementStatus">
  <xs:restriction base="xs:string">
```

```
      <xs:enumeration value="up"></xs:enumeration>
      <xs:enumeration value="draining"></xs:enumeration>
      <xs:enumeration value="down"></xs:enumeration>
    </xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="EncodingType">
          <xs:restriction base="xs:string"></xs:restriction>
  </xs:simpleType>

  <xs:simpleType name="IngestProtocolType">
    <xs:restriction base="xs:string">
      <xs:enumeration value="ftp"/>
      <xs:enumeration value="fxp"/>
      <xs:enumeration value="cifs"/>
      <xs:enumeration value="nfs link"/>
      <xs:enumeration value="multicast"/>
      <xs:enumeration value="pgm"/>
      <xs:enumeration value="norm"/>
    </xs:restriction>
  </xs:simpleType>
</xs:schema>
```