VideoXpert™
Software Development Kit
VERSION 3.0

PELCO

January 27, 2021

Contents

List of Figures

Overview & Guidelines

The VideoXpert SDK is designed to be a resource-oriented architecture that developers can use to communicate with VideoXpert systems. The SDK consists of a set of logical objects, each modeled as an interface to provide clients with access to system information. The full set of resources defined by this SDK are provided in the Interfaces section.

An SDK client begins interacting with an IVxSystem by logging in to the system. Clients can retrieve additional resources using the methods provided by this resource. For instance, in order to determine what devices exist on a system, an SDK client would execute the `GetDevices` method found in the IVxSystem interface. This would provide the client with a VxCollection containing all of the IVxDevice resources present on the system.

The following sections contain the specifications, constraints, and guidelines that apply to all of the requests and responses that occur in the VideoXpert SDK.

1.1 Imperative Terms

The key words "MUST", "SHALL", "MUST NOT", "SHALL NOT", "REQUIRED", "OPTIONAL", "SHOULD", "SHOULD NOT", "RECOMMENDED", and "MAY" in this document, and all other documents that comprise the specifications of the VideoXpert SDK, are to be interpreted as described in "Key words for use in RFCs to Indicate Requirement Levels" [RFC 2119].

1.2 Licensing

The VideoXpert SDK requires a license to be present on the system it is connecting to. If a license has not been added to the system, a 90 day grace period will begin for that system. The SDK will be able to connect and interact with the system normally during this period. If the system remains unlicensed once the grace period expires the SDK will no longer be able to communicate with the system until a valid license has been added to it.

License verification occurs during the login process by verifying that the license associated with the VxLoginInfo `licenseKey` is present and active on the system. The `licenseKey` MUST be populated with a license key string obtained from Pelco. The SDK will not attempt to connect to a system until this field has been populated with a valid key.

There are VxResult values specific to licensing (see Result Values - Licensing (SDK)) that MAY be returned by the Global `VxSystemLogin` method. If the system is not licensed and the grace period is active `kSdkLicenseGracePeriodActive` will be returned. If the system remains unlicensed after the 90 day grace period `kSdkLicenseGracePeriodExpired` will be returned. These values SHOULD be used to inform the user that a license is required. The `graceLicenseExpirationTime` field provided by IVxSystem SHOULD also be used when informing the user that the grace period is in use.

If the grace period is expires while the SDK is in use it will send out a VxInternalEvent containing the `kGraceLicenseExpired eventType` to any subscribers and then log out of the system. At this point the SDK will no longer be able to communicate with the system until a valid license has been added to it.

1.3 Collection Filtering

Often a client may require only a subset of the resources within a collection (see Collections). When this is the case, a client SHOULD filter the request using VxCollectionFilterItems in order to reduce the load on the server, client, and network. For example, IVxSystem `GetClips` may return a collection containing thousands of IVxClip making it very inefficient for a client to retrieve the entire set and then filter out the subset of data it needs on its end. When available, the allowed VxCollectionFilterItem are listed in the Collection Filters column of the Interface tables.

Clients MUST NOT send duplicate or unavailable VxCollectionFilterItem; these will be ignored by the server if sent.

1.4 Advanced Filtering

An advanced filter may be requested by clients on available resources by utilizing the `kAdvancedQuery` filter. The availability of an advanced filter on a resource is indicated by the `kAdvancedQuery` being present in the Collection Filters column of the Interface tables. The format of an advanced filter is specified using the ABNF syntax description notation defined in [RFC 2234]:

```
CONNECTIVE  = "and" / "or"
EXPRESSION  = BEXPRESSION / UEXPRESSION ; binary or unary expression
BEXPRESSION = "(" FIELDS " " OP " " VALUES ")" ; must be encoded
UEXPRESSION = "(" FIELDS " " UOP ")" ; must be encoded
FIELDS      = FIELD *("," FIELD)
OP          = "eq" / "ne" / "gt" / "gte" / "lt" / "lte" / "likei"
UOP         = "exists" / "nexists"
VALUES      = "'" VALUE "'" *(",'" VALUE "'")
FIELD       = ; case-sensitive string representing the field name
VALUE       = ; case-sensitive string representing the field value
```

where

- ***CONNECTIVE*** operators have the following precedence: `"and"` > `"or"`.
- ***FIELDS*** is a comma-separated list of case-sensitive field names to search. Only fields listed in the Collection Filters, that are fields of the resource being queried, may be used. Non-listed and/or non-attribute fields will be ignored. Field names in advanced queries use the snake_case format of the matching VxCollectionFilterItem. For example, IVxDataSource provides the `kManualRecording` VxCollectionFilterItem ; which matches the IVxDataSource `isManuallyRecording` field. Therefore this field can be used in an advanced query using the format `manual_recording`.
- ***OP*** is used to specify the manner of comparison between the given fields and values. The operator name itself is case-sensitive. Depending on the type of field(s) being queried, MUST be one of:
  - **numeric, boolean, or string fields**
    - **eq**: returns results where any of the specified numeric, boolean, or string fields are equal to any of the specified values.
    - **likei**: returns results where any of the specified numeric, boolean, or string fields contain any of the specified values, case-insensitive. Not applicable to DateTime fields.

    - **ne**: returns results where none of the specified *numeric, boolean, or string* fields are equal to any of the specified values.
  - **numeric or DateTime fields**
    - **gt**: returns results where any of the specified *numeric* fields are greater than any of the specified values.
    - **gte**: returns results where any of the specified *numeric* fields are greater than or equal to any of the specified values.
    - **lt**: returns results where any of the specified *numeric* fields are less than any of the specified values.
    - **lte**: returns results where any of the specified *numeric* fields are less than or equal to any of the specified values.
- ***UOP*** is used to specify a unary operation on a set of fields. The operator name itself is case-sensitive. MAY be used on any field types and MUST be one of:
  - **exists**: returns results where any of the specified fields have a non-null value.
  - **nexists**: returns results where any of the specified fields have a null value.
- ***VALUES*** is a comma-separated list of case-sensitive values to search. Each value is enclosed in single quotes; single quote characters within the value MUST be escaped by using two single quote characters (''). If multiple fields are supplied, they MUST be of the same type, either numeric or string.

Examples:

- Search for resources where "name" contains "hallway":

```
(name likei 'hallway')
```

- Search for resources where "name" or "desc" contains "101" or "102":

```
(name,desc likei '101','102')
```

- Search for resources where "number" is less than "101" and "number" is greater than "75":

```
(number lt '101')and(number gt '75')
```

- Search for resources where "number" is less than "50" or "number" is greater than "100":

```
(number lt '50')or(number gt '100')
```

- Search for resources where "sourceUserName" is set (non-null):

```
(source_user_name exists)
```

1.5 Counting

A client MAY request a limit on the size of a returned VxCollection `collection` resource by providing a `kCount` VxCollectionFilterItem with an integer value equal to the maximum number of items to return. If the `kCount` VxCollectionFilterItem is *not* specified, this indicates the client would like the entire collection. Clients should be aware that the server MAY return *less* than the requested maximum at its discretion. Servers MUST NOT return more than the requested maximum. The `kCount` MUST be an integer greater than or equal to `0`.

1.6 System Events

An IVxEvent is generated on the VideoXpert system whenever an IVxSituation occurs. The IVxSituation specifies how corresponding events shall be generated and handled when the situation is detected. A client may subscribe to these events using the IVxSystem `StartNotifications` method or retrieve past events using the `GetEvents` method.

By default, a client will only receive events for situations that have been configured to notify the currently logged in user. This configuration is based on which roles are present in the IVxNotification settings for each IVxSituation. If the IVxRole of the currently logged in user has been defined in an IVxNotification for a IVxSituation, the client will be notified when the situation occurs. The IVxSituation `shouldNotify` field must also be set to `true` in order for any notification to be sent for it.

Alternatively, a client may specify which situations to receive using the overloaded version of the `StartNotifications` method. Here, a client may specify the specific situation(s) to receive notifications for. These notifications are in addition to notifications received per IVxSituation configuration. And will be received regardless of the IVxSituation configuration. User notifications can be disabled by setting `userNotification` to `false`, which when combined with a list of situation(s) will result in notifications only being sent for the specified situation(s).

A client may also create custom situations and generate a new events on the system. Custom situations are referred to as "external" situations and use the format `external/<company>/<event>` for their `type`. An external IVxSituation can be added using the IVxSystem `AddSituation` method. These situations can then be used to generate new events using the IVxSystem `InsertEvent` method. Additional information related to the event may be added to the `properties` of the VxNewEvent, which will be present in the IVxEvent that is generated.

1.7 Internal Events

In addition to system events, clients may also subscribe to internal events; which are sent from the VideoXpert SDK itself. Currently, the SDK will send internal events under two circumstances: the connection to the system was lost/restored or a system/grace license has expired. A client may subscribe to these events using the IVxSystem `StartInternalNotifications` method.

The process for receiving internal events is the same as system events. However, the notification received by the client will contain a VxInternalEvent instead of an IVxEvent. The reason for the event will be specified by the VxInternalEventType set in the `eventType` field. These events SHOULD be used to perform the appropriate actions based on the VxInternalEventType. For example, the `kGraceLicenseExpired` event should be used to inform the client that the license grace period has expired and return the client to a pre-login state (if applicable).

1.8 Media Retrieval

Media data retrieval is accomplished using the RTSP control protocol [RFC 2326] or MJPEG (in which a IVxDataSession acts as the control protocol). Both live and recorded media are accessed using the same method; clients do not need to know where the media is located to retrieve it.

The high level steps a client will perform (see diagram) are:

1. Get the IVxDataSource producing the desired media.

2. Select the IVxDataInterface from the IVxDataSource `dataInterfaces` attribute that has the desired interface.
3. Use the VxStreamProtocol and the `dataEndpoint` attribute on the IVxDataInterface to control the media stream. The server will determine whether it should pull the media from a live source or a recorded source based on the control requests.

Every IVxDevice that produces media data contains one or more IVxDataSource. These IVxDataSources represent various types of media that the IVxDevice can produce (e.g. audio, video, or metadata). An IVxDataSource provides a list of IVxDataInterfaces, each of which provides a particular protocol used to transmit and control the data produced by the IVxDevice.

A media session is initiated using the VxStreamProtocol defined by the IVxDataInterface. This session is represented by the server as a IVxDataSession. An IVxDataSession is automatically created by the server for every RTSP session or explicitly requested for an MJPEG session.

When a client closes the stream, the server will automatically delete the associated IVxDataSession.

1.9 RTSP Streaming

Initiating and controlling an RTSP stream from the server is accomplished using the RTSP control protocol [RFC 2326]. Due to the distributed architecture of VideoXpert, the initial stream request is used by the server to determine the location of the requested media (see sequence diagram). The server will return "spoofed" responses during these initial requests. The information in these responses should NOT be used to set up the media client. The server will redirect the client to the URI of the requested media in the Location header of the PLAY response. This URI may then be used to initialize and control the stream. These responses will also contain the actual stream information and may be used to set up the media client.

Understanding RTSP Commands:

- **`OPTIONS`**: "What RTSP commands do you support?"
- **`DESCRIBE`**: "Tell me about which streams/tracks are available at the given stream URL, typically in SDP format"
- **`SETUP`**: "Please set up the requested stream according to the info I specify in the Transport header. I understand that you may choose to stream in a different way then I requested, so I will await your response to find out how you will stream. The SETUP response will also include a Session header which will include a session ID along with an optional session timeout value (in seconds)."
  - protocol (e.g. RTP)
  - transport method (e.g. unicast or multicast)
  - ports to receive data (e.g. 6780-6781, where first port is to receive RTP data, and second port is to receive RTCP data)
- **`PLAY`**: "Please stream all streams that I have previously SETUP to the locations that you told me you would stream them in the respective SETUP responses. I will pass the session ID value that I received in the SETUP response to reference the applicable session in the Session header."
- **`GET_PARAMETER`**: "Please keep the session alive for the session ID specified in the Session header. This session ID value was obtained in the earlier SETUP response."
- **`TEARDOWN`**: "Please tear either entire session or the individual track (based on the provided URI) for the session ID specified in the Session header. This session ID value was obtained in the earlier SETUP response."

1.10 MJPEG Streaming

MJPEG streams are controlled using an IVxDataSession that is created by an IVxDataSource (see Media Retrieval for details). Once created, JPEG frames may be retrieved from the server through HTTP GET requests to the data endpoint specified in the `jpegUri` field (see sequence diagram).

MJPEG streams use a pull transport method to deliver frames to the client. Which means the client MUST send a new HTTP GET request for each JPEG frame it wishes to receive. For every request, clients MUST perform authentication. Clients MUST send an HTTP cookie [RFC 6265] containing the current authentication token for the session. The current authorization token can be retrieved by using the IVxDataSource `GetAuthToken` method, which SHOULD only be used for the initial HTTP GET request.

In response to a successful request, the server MAY return an HTTP cookie containing an updated authentication token. The server will expire a token after a period of time has elapsed and will provide the remaining token expiration time in the cookie Max-Age attribute. Clients MUST always use the latest token received as the server MAY return a new authentication token in response to any request, invalidating the previous token.

Example cookie request:

```
Cookie: auth_token=b2eb-a24-6fd3
```

Example cookie response:

```
Set-Cookie: auth_token=a6eb-a14-2ad1; Max-Age=7427
```

Additionally, the server MAY return a custom `X-Resource-Timestamp` HTTP header in the response. When present, this header will contain a DateTime value representing the absolute temporal position of the requested JPEG frame.

Example timestamp response:

```
X-Resource-Timestamp: 2018-10-23T16:41:00.579-07:00
```

If there is currently no data, but more will become available, the server will return a `403 (Forbidden)/EdgeOfStream` response. This response will typically occur when requests are being sent too frequently. If there is no remaining data, the server will return a `403 (Forbidden)/EndOfStream` response.

The IVxDataSession MAY timeout if the client does not perform a request or call the `RefreshSession` method within 30 seconds. If a `404 (Not Found)` is received when interacting with this IVxDataSession, it is safe to assume the session has timed out; a new IVxDataSession will need to be created from the IVxDataSource.

1.11 Best Practices

Tips, recommendations, and principles of good practice for building high quality applications with the VideoXpert SDK. Following these points will help to maximize the performance of both client and server

applications while reducing development overhead.

Request only the data you need. Make use of available filters to limit the data returned by SDK requests to only the data you require.

Do not continue sending requests if you are receiving error messages. If you are receiving error messages, stop sending requests, analyze the error messages, and take remedial action.

Primitives

2.1 Basic Types

Type	Value	Description	Examples
Boolean	boolean	A logical truth value, either true or false.	true
ColorArgb	number	An integer that defines an ARGB (Alpha, Red, Green, Blue) color. Range is from 0 to 0xFFFFFFFF	4278255360
DateTime	string	Representation of date and time using the Gregorian calendar. The format used is [RFC 3339].	“1985-04-12T23:20:50.52Z” “1996-12-19T16:39:57-08:00” “1990-12-31T23:59:59Z”
EmailAddress	string	An email address conforming to the `addr-spec` in [RFC 5322, Section 3.4.1]. *Note: Visit `http://emailregex.com` for tips.*	“jdoe@example.com”
Float	number	A fractional number represented as a decimal value.	3.5 7
Host	string	Hostname [RFC 1123], IPv4 address, or IPv6 address.	“pool.ntp.org” “10.221.220.110” “fe80::218:8bff:fe6d:957b”
Integer	number	A whole number that may be positive or negative.	-8 42
IP	string	IPv4 or IPv6 address represented in the standard notation (see below).	“10.221.220.110” “fe80::218:8bff:fe6d:957b”
IPv6	string	IPv6 address in hexadecimal form that consists of eight double-bytes separated by colons. The address representation may be shortened by abbreviating a number of consecutive zeros to a double colon, which is allowed once in any single IPv6 address [RFC 4291]. IPv6 literals within URIs are distinguished by enclosing the IP literal within square brackets [RFC 3986].	“fe80::218:8bff:fe6d:957b”
SASLString	string	A set of Unicode characters, according to the SASLprep profile [RFC 4013] of stringprep [RFC 3454] (see [RFC 4013, Section 2.3] for a list of prohibited characters), that are UTF-8 encoded. Case-sensitive unless otherwise specified. Summary: All valid JSON strings minus control, non-printable, and non-ASCII space characters.	“Hello”
String	string	A sequence of characters.	“Hello”
Time	string	A time subset of the DateTime representation; represents a fixed point in time. The format used is the `full-time` portion of the [RFC 3339, Section 5.6] ABNF.	“23:20:50.52Z” “16:39:57-08:00” “23:59:59Z”

Type	Value	Description	Examples
TimeOfDay	string	A time subset of the DateTime representation; represents a relative time of day (relative to the server's local time). The format used is the `partial-time` portion of the [RFC 3339, Section 5.6] ABNF.	"21:20:05.42" "09:00:00" "23:59:59"
UPN	string	A User Principal Name has two String parts: the UPN prefix (a user name) and a UPN suffix (a domain name). The parts are joined together by the at symbol (`@`) to make the complete UPN. The UPN prefix alone MAY be provided (e.g. `jdoe`) in order to use the default domain (e.g. `LOCAL` for VideoXpert). If only the suffix is omitted (e.g. `jdoe@`), this indicates that no domain shall be used.	"JDoe@Example" "JDoe@" "JDoe"
URI	string	A HTTP link to a web resource. This can be either a full or a relative path. The format used is [RFC 3986].	"/system/devices" "http://10.221.1.80:8080/system"

2.2 Enumerations

Type	Values	Description
VxAccessPointType	kUnknown	An error or unknown value was returned.
	kDoor	A door.
	kDoorWithReader	A door with a reader.
	kGate	A gate.
VxAccessStatus	kUnknown	An error or unknown value was returned.
	kClosed	The access point is closed.
	kFaulted	The access point has faulted.
	kForced	The access point has been forced.
	kLocked	The access point is locked.
	kOpened	The access point is open.
	kPropped	The access point has been propped.
	kUnlocked	The access point is unlocked.
VxAckState	kUnknown	An error or unknown value was returned.
	kAckNeeded	The event needs acknowledgement.
	kAcked	The event is acknowledged.
	kAutoAcked	The event is auto-acknowledged.
	kNoAckNeeded	No event acknowledgement is needed.
	kSilenced	The event has been silenced.
VxAlarmInputType	kUnknown	An error or unknown value was returned.
	kAlarm	The alarm input is a generic alarm.
	kFire	The alarm input is a fire alarm.
VxAlarmState	kUnknown	An error or unknown value was returned.
	kActive	The alarm input is active.
	kInactive	The alarm input is inactive.
VxAnalyticBehaviorType	kUnknown	An error or unknown value was returned.
	kObjectLineCounter	Object line counting analytic.
	kObjectInZone	Object detected in zone analytic.
	kObjectWrongWay	Object wrong way analytic.

Type	Values	Description
VxAnalyticObjectType	kUnknown	An error or unknown value was returned.
	kPerson	A person object type.
	kVehicle	A vehicle object type.
VxAuthProtocol	kUnknown	An error or unknown value was returned.
	kMD5	MD5 authentication protocol.
	kNone	No authentication protocol.
	kSHA	SHA authentication protocol.
VxBackupStatus	kUnknown	An error or unknown value was returned.
	kFailed	Backup has failed.
	kHalted	Backup was halted.
	kInProgress	Backup is in progress.
	kSuccessful	Backup was successful.
VxBackupStatusReason	kUnknown	An error or unknown value was returned.
	kBackupStorageFull	Backup storage is full.
	kBackupStorageUnauthenticated	Backup storage unauthenticated.
	kBackupStorageUnavailable	Backup storage is unavailable.
	kShadowCopyCreationFailure	Shadow copy creation failed.
	kShadowCopyMountFailure	Shadow copy mount failed.
VxBookmarkRetentionPolicy	kUnknown	An error or unknown value was returned.
	kDeleteAfterLimit	Unlocked bookmarks are automatically deleted after the retention limit.
	kDeleteAfterRecording	Unlocked bookmarks are automatically deleted after their corresponding recordings are deleted.
	kNeverDelete	Unlocked bookmarks are not automatically deleted.
VxCellInputMode	kUnknown	An error or unknown value was returned.
	kDigitalPtz	Digital PTZ mode.
	kNavigation	Navigation mode.
	kPlayback	Playback mode.
	kPtz	PTZ mode.
VxCellLayout	k1x1	A 1x1 monitor layout.
	k1x2	A 1x2 monitor layout.
	k2x1	A 2x1 monitor layout.
	k2x2	A 2x2 monitor layout.
	k2x3	A 2x3 monitor layout.
	k3x2	A 3x2 monitor layout.
	k3x3	A 3x3 monitor layout.
	k4x3	A 4x3 monitor layout.
	k4x4	A 4x4 monitor layout.
	k5x5	A 5x5 monitor layout.
	k1plus12	A 1 plus 12 monitor layout.
	k2plus8	A 2 plus 8 monitor layout.
	k3plus4	A 3 plus 4 monitor layout.
	k1plus5	A 1 plus 5 monitor layout.
	k1plus7	A 1 plus 7 monitor layout.

Type	Values	Description
	k12plus1	A 12 plus 1 monitor layout.
	k8plus2	A 8 plus 2 monitor layout.
	k1plus1plus4	A 1 plus 1 plus 4 monitor layout.
	k1plus4tall	A 1 plus 4 tall monitor layout.
	k1plus4wide	A 1 plus 4 wide monitor layout.
	kMonitorWall	A monitor wall layout.
VxCollectionFilterItem	kNone	No filter.
	kCount	The maximum number of items to return per page.
	kDataSourceId	Filter by data source id.
	kSearchStartTime	Filter by start time.
	kSearchEndTime	Filter by end time.
	kName	Filter by name.
	kId	Filter by id.
	kStart	The start index.
	kType	Filter by type.
	kSituationType	Filter by situation type.
	kUnassigned	True to return only items that are not assigned.
	kDataSourceNumber	Filter by data source number.
	kDescription	Filter by the description value.
	kModifiedSince	Filter by time since last modified.
	kTime	Filter by the time value.
	kEndTime	Filter by the end time value.
	kStartTime	Filter by the start time value.
	kEvent	Filter by the event value.
	kFramerate	Filter by the framerate value.
	kClientType	Filter by the client type value.
	kOwned	True to return only owned items.
	kOwner	Filter by owner.
	kUsername	Filter by the username value.
	kAllTags	Filter tags by name.
	kCapturing	True to return only items that are capturing.
	kIp	Filter by IP.
	kNumber	Filter by number.
	kRecording	True to return only items that are recording.
	kState	Filter by the state value.
	kCommissioned	True to return only items that are commissioned.
	kModel	Filter by model.
	kSerial	Filter by serial.
	kVendor	Filter by vendor.
	kVersion	Filter by version.

Type	Values	Description
	kAckState	Filter by the ack state value.
	kAckUser	Filter by the ack user value.
	kGeneratorDeviceId	Filter by generator device id.
	kNotifies	True to return items that were sent to the client.
	kSeverity	Filter by severity.
	kSourceDeviceId	Filter by source device id.
	kSourceUserName	Filter by source username.
	kPercentComplete	Filter by the percent complete value.
	kSize	Filter by the size value.
	kStatus	Filter by status.
	kResourceId	Filter by resource id.
	kInternal	True to return items that are internal (read-only).
	kAudibleNotify	Filter by the audible notification value.
	kLog	Filter by the log value.
	kNotify	Filter by the notify value.
	kResourceType	Filter by resource type.
	kLinked	True to return only items that are linked.
	kAllPrivateTags	Filter tags by name (owned by current user).
	kManualRecording	Filter by items being manually recorded.
	kFirstName	Filter by first name.
	kLastName	Filter by last name.
	kHasProperty	Filter by items that define a specific property.
	kHasStatus	Filter by items that have a specific status.
	kDataSourceAllTags	Filter tags by name.
	kDataSourceAllPrivateTags	Filter tags by name (owned by current user).
	kDataSourceName	Filter by data source name.
	kServicePropertyId	Filter by name of the corresponding service id.
	kImageType	Filter by a image type.
	kAdvancedQuery	Filter by advanced query.
	kTagsAll	Filter by all public and private tags.
	kEnabled	True to return only items that are enabled.
	kInitiated	Filter by the intiated time value.
	kDataStorageId	Filter by data storage id.
	kTrashed	True to return only items that have been trashed.
	kDataSourceType	Filter by data source type.
	kDeviceId	Filter by device id.
	kDriverType	Filter by driver type.
	kEncoding	Filter by encoding type.
	kFolder	True to return only items that are folders.

Type	Values	Description
	kGroupId	Filter by group id.
	kHasFolderTags	True to return only items that have folder tags.
	kLayerName	Filter by layer name.
	kLicenseRequired	True to return only items that require a license.
	kLocked	True to return only items that are locked.
	kParentId	Filter by parent id.
	kProvider	Filter by provider type.
	kRecordType	Filter by recording type.
VxConfigStatus	kUnknown	An error or unknown value was returned.
	kConfigured	The cluster is configured.
	kConfiguring	The cluster is being configured.
	kFailed	Cluster configuration failed.
	kUnconfigured	The cluster is unconfigured.
VxDatabaseStatus	kUnknown	An error or unknown value was returned.
	kFailed	The database has failed.
	kOk	The database is OK.
	kRebuilding	The database is rebuilding.
	kRecovering	The database is recovering.
VxDataSourceCapability	kUnknown	An error or unknown value was returned.
	kFocus	Focus capability.
	kIris	Iris capability.
	kPanTilt	Pan/tilt capability.
	kZoom	Zoom capability.
VxDataSourceType	kUnknown	An error or unknown value was returned.
	kVideo	A video data source.
	kAudio	An audio data source.
	kMetadata	An metadata data source.
VxDataStorageType	kUnknown	An error or unknown value was returned.
	kDigitalSentry	A Digital Sentry device.
	kNSM	An NSM network video recorder device.
	kVideoXpertStorage	A VideoXpert storage device.
	kEdge	An edge storage device.
VxDayOfWeek	kUnknown	An error or unknown value was returned.
	kMonday	Monday.
	kTuesday	Tuesday.
	kWednesday	Wednesday.
	kThursday	Thursday.
	kFriday	Friday.
	kSaturday	Saturday.
	kSunday	Sunday.
VxDeviceState	kUnknown	An error or unknown value was returned.
	kOffline	The device is offline.
	kOnline	The device is online.
VideoXpertStatus	kUnknown	An error or unknown value was returned.

Type	Values	Description
	kInitializing	Being prepared for use.
	kUnauthenticated	Invalid/missing credentials.
	kIdInconsistent	Device identity mismatch.
	kNsmManager	NSM5200 manager.
	kNsmMember	NSM5200 member.
	kAuthExpired	Expired credentials.
VxDeviceType	kAcc	A VideoXpert Accessory Server device.
	kAccessController	An Access Control device.
	kAllInOne	A VideoXpert all in one device (i.e. VxPro).
	kCamera	A camera device.
	kCore	A VideoXpert Core device.
	kDecoder	A decoder device.
	kEncoder	An encoder device.
	kExternal	An external device.
	kGeneric	A generic device.
	kMg	A VideoXpert MediaGateway device.
	kMonitor	A monitor device.
	kRecorder	A network storage device.
	kUi	A UI device.
	kUnknown	An error or unknown value was returned.
	kAnalyticServer	An analytic server.
	kDatabase	A VideoXpert database.
VxDrawingProvider	kUnknown	An error or unknown value was returned.
	kEsri	Esri drawing provider.
	kSerenity	Serenity drawing provider.
VxExportFormat	kUnknown	An error or unknown value was returned.
	kMkvZip	MKV file(s) contained withing a zip file.
VxExportStatus	kUnknown	An error or unknown value was returned.
	kExporting	The export is in progress.
	kFailed	The export has failed.
	kPending	The export is queued to start.
	kSuccessful	The export has completed successfully.
VxExportStatusReason	kUnknown	An error or unknown value was returned.
	kExportDataUnretrievable	The export data is unretrievable.
	kExportStorageFull	The export storage is full.
	kExportStorageUnauthenticated	The export storage is unauthenticated.
	kExportStorageUnavailable	The export storage is unavailable.
VxExportStreamStatus	kUnknown	An error or unknown value was returned.
	kFailed	The export stream preparation has failed.
	kNeedsPreparation	The export stream needs preparation.
	kPreparing	The export stream is being prepared.
	kReady	The export stream is ready.
VxExportStreamStatusReason	kUnknown	An error or unknown value was returned.
	kExportDataUnretrievable	The export data is unretrievable.
	kExportTempStorageFull	The temporary export storage is full.

Type	Values	Description
VxFileRecoveryStatus	kUnknown	An error or unknown value was returned.
	kFailed	The file recovery failed.
	kIdle	The file recovery process is idle.
	kInProgress	The file recovery is in progress.
	kSuccess	The file recovery succeeded.
VxFocusDirection	kStop	Stop focus movement.
	kFar	Focus farther.
	kNear	Focus nearer.
VxGapFillerStatus	kUnknown	An error or unknown value was returned.
	kDisabled	Gap filler is disabled.
	kFailed	Failed to fill the gap.
	kPending	Gap filling is pending.
	kUnavailable	Gap filler status is unavailable.
VxGapReason	kUnknown	An error or unknown value was returned.
	kCameraOffline	Camera offline.
	kNotSupported	Not supported.
	kStorageOffline	Storage device offline.
	kStreamLoss	Stream loss.
	kStreamSourceChanged	Stream source changed.
	kTimeJump	Time jump.
	kTransportChanged	Transport changed.
	kWriteError	Write error.
VxHddStatus	kUnknown	An error or unknown value was returned.
	kFailed	The disk has failed.
	kMissing	The disk is missing.
	kOk	The disk is OK.
	kRebuilding	The disk is rebuilding.
	kUnconfigured	The disk has not been configured.
VxInternalEventType	kUnknown	An error or unknown value was returned.
	kSystemConnectionLost	Connection to the VideoXpert system lost.
	kSystemConnectionRestored	Connection to the VideoXpert system restored.
	kGraceLicenseExpired	The grace license has expired.
	kSystemLicenseExpired	The license on the system has expired.
VxIntersectionArea	kUnknown	An error or unknown value was returned.
	kEastboundLeft	East-bound left area.
	kEastboundPedestrian	East-bound pedestrian area.
	kEastboundRight	East-bound right area.
	kEastboundThrough	East-bound through area.
	kNorthboundLeft	North-bound left area.
	kNorthboundPedestrian	North-bound pedestrian area.
	kNorthboundRight	North-bound right area.
	kNorthboundThrough	North-bound through area.
	kSouthboundLeft	South-bound left area.
	kSouthboundPedestrian	South-bound pedestrian area.
	kSouthboundRight	South-bound right area.

Type	Values	Description
	kSouthboundThrough	South-bound through area.
	kWestboundLeft	West-bound left area.
	kWestboundPedestrian	West-bound pedestrian area.
	kWestboundRight	West-bound right area.
	kWestboundThrough	West-bound through area.
VxIrisDirection	kStop	Stop iris movement.
	kClose	Close the iris.
	kOpen	Open the iris.
VxLineCounterType	kUnknown	An error or unknown value was returned.
	kUnidirectional	Type of line-counter that counts objects that cross a configured line in a specific direction.
	kBidirectional	Type of line-counter that maintains two counts for objects that cross a configured line on either side of the line. For example in/out door counts.
	kOmnidirectional	Type of line-counter that maintains a single count for objects that cross a configured line in any direction.
VxLoadType	kUnknown	An error or unknown value was returned.
	kCpu	CPU load.
	kGpu	GPU load.
	kMemory	Memory load.
VxLogLevel	kTrace	Trace log level.
	kDebug	Debug log level.
	kInfo	Info log level.
	kWarning	Warning log level.
	kError	Error log level.
	kFatal	Fatal log level.
	kNone	Disable logging.
VxMemberState	kUnknown	Unable to determine status or an error was returned.
	kOnline	Online; available.
	kPending	Being added; unavailable.
	kRemoving	Being removed; unavailable.
	kUnauthorized	Invalid credentials; unavailable.
	kUnavailable	Offline; unavailable.
VxMgTranscast	kUnknown	An error or unknown value was returned.
	kMulticast_Multicast	Multicast between the data source and VxMG; multicast between VxMG and client.
	kMulticast_Unicast	Multicast between the data source and VxMG; unicast between VxMG and client.
	kUnicast_Multicast	Unicast between the data source and VxMG; multicast between VxMG and client.
	kUnicast_Unicast	Unicast between the data source and VxMG; unicast between VxMG and client.

Type	Values	Description
VxMotionMode	kUnknown	An error or unknown value was returned.
	kCamera	Camera based motion detection.
	kDisabled	Motion detection disabled.
	kRecorder	Recorder based motion detection.
VxNodeType	kUnknown	An error or unknown value was returned.
	kCore	Core node.
	kDatabase	Database node.
	kSupport	Support node.
VxOverlayType	kUnknown	An error or unknown value was returned.
	kMotion	Motion overlay.
VxPermissionId	kUnknown	An error or unknown value was returned.
	kViewLiveMedia	View live media.
	kUsePtzMode	Use PTZ mode.
	kLockPtzMode	Lock PTZ mode.
	kRecordMedia	Record media.
	kLaunchSavedViewsRemotely	Launch saved views remotely.
	kAccessAlarms	Access alarms.
	kAccessRelays	Access relays.
	kViewRecordedMedia	View recorded media.
	kSystemBookmarks	System bookmarks.
	kSystemLocks	System locks.
	kExportMediaClips	Export media clips.
	kManageExports	Manage exports.
	kUseMap	Use map.
	kViewMaps	View maps.
	kPlaceCamerasOnMap	Place cameras on map.
	kManageMapFiles	Manage map files.
	kDefinePtzPresets	Define ptz presets.
	kManageCameraTours	Manage camera tours.
	kAccessUserWorkspaces	Access user workspaces.
	kManageSystemWorkspaces	Manage system workspaces.
	kAuditUserActivity	AuditUser activity.
	kMultiviewQty	Multiview qty.
	kViewEventHistory	View event history.
	kHandleEvents	Handle events.
	kConfigureEvents	Configure events.
	kManageUserAccounts	Manage user accounts.
	kAssignRolesToUsers	Assign roles to users.
	kResetUserPasswords	Reset user passwords.
	kManageRoles	Manage roles.
	kManageSystemTags	Manage system tags.
	kManageIO	Manage I/O.
	kManageDeviceLicenses	Manage device licenses.
	kUpdateDeviceSoftware	Update device software.
	kSetupEdgeDevices	Setup edge devices.

Type	Values	Description
	kManageDisplayDevices	Manage display devices.
	kConfigureMonitorWallDecoders	Configure monitor wall decoders.
	kManageSystemLicenses	Manage system licenses.
	kSetSystemLocaleOptions	Set system locale options.
	kDefineSystemShortcuts	Define system shortcuts.
	kConfigureRecording	Configure recording.
	kViewSystemHealth	View system health.
	kManageSystemServers	Manage system servers.
	kManageMemberSystem	Manage member systems.
VxPhoneType	kHome	Home number.
	kHomeFax	Home fax number.
	kMobile	Mobile number.
	kOther	Other number.
	kPager	Pager number.
	kWork	Work number.
	kWorkFax	Work fax number.
VxPrivacyProtocol	kUnknown	An error or unknown value was returned.
	kAES	AES encryption.
	kDES	DES encryption.
	kNone	No encryption.
VxRecordingFramerate	kUnknown	An error or unknown value was returned.
	kLow	Low framerate.
	kNormal	Normal framerate.
VxRecordingType	kUnknown	An error or unknown value was returned.
	kAlarm	Hardware or software alarm.
	kAnalytic	Video analytic (non-motion).
	kEvent	General system event.
	kManual	Manual user initiation.
	kMotion	Motion anayltic.
	kTimed	Time-based (continuous); no event.
VxRelayState	kUnknown	An error or unknown value was returned.
	kActive	The relay output is active.
	kInactive	The relay output is inactive.
VxRenderType	kUnknown	An error or unknown value was returned.
	kEvo	Evo renderer.
	kOptera180	Optera 180 renderer.
	kOptera270	Optera 270 renderer.
	kOptera360	Optera 360 renderer.
	kStandard	Standard renderer.
VxResourceType	kUnknown	An error or unknown value was returned.
	kDataSource	A data source resource.
	kDevice	A device resource.
	kUser	A user resource.
	kDrawing	A drawing resource.
	kDataStorage	A data storage resource.
	kTag	A tag resource.

Type	Values	Description
	kRelayOutput	A relay output resource.
VxResult	See Result Values section.	
VxRotationType	kUnknown	An error or unknown value was returned.
	k0	0 degrees.
	k90	90 degrees.
	k180	180 degrees.
	k270	270 degrees.
VxRtspCapability	kUnknown	An error or unknown value was returned.
	kTcp	TCP streaming capability.
	kTcpUdp	TCP and UDP streaming capabilities.
	kTcpUdpMulticast	TCP, UDP and multicast streaming capabilities.
VxScheduleAction	kUnknown	An error or unknown value was returned.
	kEventSourceRecord	Record the resource that triggered an event.
	kRecord	Record all resources associated with the schedule.
VxSearchStatus	kUnknown	An error or unknown value was returned.
	kComplete	The search has completed.
	kInProgress	The search is in progress.
VxSnapshotFilterItem	kNone	No filter.
	kStartTime	Time where the initial image should start .
	kEndTime	Time where no further images should be returned.
	kWidth	Scale to given width in pixels, maintaining ratio.
	kOffset	Offset time, in seconds, between images.
VxSnmpVersion	kUnknown	An error or unknown value was returned.
	kNone	SNMP is disabled.
	kSNMP2c	SNMPv2c protocol.
	kSNMP3	SNMPv3 protocol.
VxStreamFormat	kUnknown	An error or unknown value was returned.
	kH264	H.264 encoding format.
	kH265	H.265 encoding format.
	kMpeg4	MPEG-4 encoding format.
	kJpeg	JPEG encoding format.
	kG711	G.711 encoding format.
	kMetadata	Metadata encoding format.
VxStreamProtocol	kUnknown	An error or unknown value was returned.
	kMjpegPull	The Mjpeg (pull) protocol.
	kRtspRtp	The RTSP/RTP protocol.
VxStreamSource	kUnknown	An error or unknown value was returned.
	kPrimary	The 1st stream.
	kSecondary	The 2nd stream.
	kTertiary	The 3rd stream.
	kQuaternary	The 4th stream.
	kQuinary	The 5th stream.

Type	Values	Description
	kSenary	The 6th stream.
	kSeptenary	The 7th stream.
	kOctonary	The 8th stream.
	kNonary	The 9th stream.
	kDenary	The 10th stream.
VxSupEventType	kUnknown	An error or unknown value was returned.
	kCommissioned	A license commissioned event.
	kDecommissioned	A license decommissioned event.
	kExpiredSup1	A 1-year license expired event.
	kExpiredSup3	A 3-year license expired event.
	kExtendedSup	A license extended event.
	kInstalledSup1	A 1-year license installed event.
	kInstalledSup3	A 3-year license installed event.
	kTampered	A license tampered event.
VxSystemLicenseType	kUnknown	An error or unknown value was returned.
	kEnterprise	Enterprise system license.
	kProfessional	Professional system license.
	kUnlicensed	Unlicensed system.
VxTransmissionType	kUnknown	An error or unknown value was returned.
	kMulticast	Multicast transmission.
	kUnicast	Unicast transmission.
VxZoomDirection	kNone	Do not perform a zoom action.
	kStop	Stop zoom movement.
	kIn	Zoom in.
	kOut	Zoom out.

Result Values

This section documents the VxSDK result values that are returned from methods.

3.1 Data Retrieval

Code	Description
`kCameraUnavailable`	Camera is unavailable, data cannot be retrieved at this time.
`kEdgeOfStream`	No further stream data is *currently* available (the edge of a currently recording clip has been reached); more data will be available shortly.
`kEndOfStream`	No further stream data is available (the end of all recorded data has been reached; no further data is currently be recorded).
`kNoAvailableStreams`	The server is unable to initiate any new stream sessions due to having reached its stream count capacity.
`kStorageUnavailable`	Storage is unavailable, data cannot be retrieved at this time.

3.2 Exporting

Code	Description
`kExportDataUnretrievable`	The data needed to perform the export operation can not be retrieved.
`kExportStorageFull`	The export storage location does not have enough free space to store the export.
`kExportStorageUnauthenticated`	The export storage location is not accessible due to invalid credentials.
`kExportStorageUnavailable`	The export storage location is not accessible; this may be due to an invalid location, network issue, or storage issue.

3.3 General

Code	Description
kUnknownError	An error or unknown value was returned.
kOK	The action was successful.
kUnsupportedVersion	The VideoXpert system version is not supported.
kInsufficientSize	The size value was not sufficient enough to allocate the collection.
kInvalidLoginInfo	The login credentials were invalid.
kActionUnavailable	The attempted action is unsupported by the system.
kInvalidParameters	A parameter was invalid.
kCommunicationError	There was an error communicating to the device.
kConflict	The requested operation is not possible due to a conflict with the resource. Typically this is due to a violation of a uniqueness property on one of the resource's fields. If this is the case, the server MUST return the field causing the conflict as part of the returned VxError.
kInsufficientResources	The server has insufficient resources to satisfy the request.
kNotReady	The server is not in an appropriate state to be able to service this request. The server requires intervention in order to resolve this. This is not a temporary condition.
kNotReadyUnauthenticated	The server is not in an appropriate state to be able to service this request due to an authentication issue between it and another entity. The server requires intervention in order to resolve this. This is not a temporary condition.
kOperationFailed	The requested operation failed.
kResponseTooLarge	The server is incapable of handling the client request due to the size of the resulting response. What constitutes 'too large' is entirely up to the server. Servers SHOULD only use this code when no other options are available. Clients SHOULD attempt to page the request (or reduce the page size, if already paging) if this response is received.

3.4 Licensing (SDK)

Code	Description
kSdkLicenseKeyEmpty	The SDK license key value is empty.
kSdkLicenseKeyInvalid	The SDK license key data is invalid.
kSdkLicenseVersionInvalid	The license on the server does not support this version of the VxSDK.
kSdkLicenseExpired	The license on the server has expired.
kSdkLicenseGracePeriodActive	The license associated with the license key was not found on the server, but the license grace period is active.
kSdkLicenseGracePeriodExpired	The license associated with the license key was not found on the server and the license grace period has expired.

3.5 Licensing (System)

Code	Description
kActivationConflict	The license that was supplied has an activation conflict with an existing license (e.g. duplicate activation IDs).
kActivationHostNotFound	The activation failed due to communication error with the FNO licensing server.
kActivationFailed	The activation failed.
kIncompatibleLicense	The license that was supplied is not compatible with the device and/or system that it is being applied to.
kInvalidLicense	The license that was supplied is invalid.
kLicenseCountExceeded	A valid license is available but the available count on that license is fully utilized.
kLicenseRequired	A valid license is required to utilize this method on the resource; no valid license found.
kLicenseReqLdapAdmin	Unable to apply the license; valid LDAP administrator credentials are required.
kNoLicense	Unable to commission (or float) a feature because no valid license is available for it.

3.6 Locking / Prioritization

Code	Description
kCameraInUse	Camera is in use (or the usage dwell time is active) by same or higher authority user.
kCameraLocked	Camera is locked by same or higher authority user.
kNeedOverride	Locked by lower authority user; MAY override.

3.7 Resource Editing

Code	Description
kInvalidValue	An attempt to set an invalid value on a writable field was made. The value may be invalid due to being out of range, unavailable, etc. Note that if the value is invalid due to being in conflict (i.e. it would violate the field's uniqueness property), the kConflict code should be used instead.
kPortInUse	An attempt to set a new port number failed because the port number is already in use.
kReadOnlyField	An attempt to edit a read-only field was made.
kResourceLocked	An attempt to edit a locked resource was made by a user that does not own the IVxResourceLock.

3.8 Security

Code	Description
kAuthExpired	A request was made using expired authentication credentials.
kPasswordReqMoreDigits	A password with an insufficient number of digits was supplied in an attempt to create a new user password.
kPasswordReqMoreLower	A password with an insufficient number of lowercase letters was supplied in an attempt to create a new user password.
kPasswordReqMoreSpecial	A password with an insufficient number of special characters was supplied in an attempt to create a new user password.
kPasswordReqMoreUpper	A password with an insufficient number of uppercase letters was supplied in an attempt to create a new user password.
kPasswordTooShort	A password of insufficient length was supplied in an attempt to create a new user password.
kPasswordTooSimilar	A password too similar to a previous password was supplied in an attempt to create a new user password.
kPermissionConflict	The requested operation is not possible due to a permission conflict with the resource. Typically this is due to a violation of permission hierarchy (e.g. a nested permission is being assigned without its parent permission already assigned).
kUnauthenticated	An unauthenticated request was made (i.e. invalid username and/or password).
kUnauthorized	An unauthorized request was made (i.e. user does not have permission to access the resource).

Data Types

This section defines the different data types available within the VideoXpert SDK. Clients MUST provide the value for all *required* fields sent to the server. Clients MAY provide support for *optional* fields. If a server does not support the functionality required by an optional field, or a client does not have authorization to it, the server will omit the field from the representation. If a server is unable to provide the value of a field (e.g. if the field has no value or the field is write-only), the server will omit it.

The columns that make up the representation descriptions have the following meanings:

- **Field Name**: The name of the field within the VxSDK.
- **Type**: Primitive type of the value contained in this field.
- **Req**: Indicates whether this field is required to be present in requests.
- **Description**: A brief description of the field.

4.1 VxCollection

Description
Represents a collection of resources.

Field Name	Type	Req	Description
collection	Template	Y	The collection of resources.
collectionSize	Integer	Y	The size of `collection`.
filters	VxCollectionFilter[]	N	The filters to be applied to the collection request.
filterSize	Integer	N	The size of `filters`.
startIndex	Integer	N	The start index.
totalItems	Integer	N	The total amount of items.

4.2 VxCollectionFilter

Description
Represents a filter to be used when creating a VxCollection.

Field Name	Type	Req	Description
key	VxCollectionFilterItem	Y	The filter key.
value	String	Y	The filter value.

4.3 VxDiagnostics

Description
Represents a collection of device diagnostics. Each diagnostic is represented by a list of that diagnostic's type. When multiple diagnostics exist in the list, the diagnostic identifier may be used to distinguish between them.

Field Name	Type	Req	Description
assignments	VxDiagnostics::Assignments	N	The assignments diagnostics.
backupPowers	VxDiagnostics::BackupPower[]	N	The backup power diagnostics.
backupPowerSize	Integer	N	The size of `backupPower`.
databases	VxDiagnostics::Database[]	N	The database diagnostics.
databasesSize	Integer	N	The size of `databases`.
eventing	VxDiagnostics::Eventing	N	The eventing diagnostics.
fans	VxDiagnostics::Fan[]	N	The fan diagnostics.
fansSize	Integer	N	The size of `fans`.
hdds	VxDiagnostics::Hdd[]	N	The hdd diagnostics.
hddsSize	Integer	N	The size of `hdds`.
loads	VxDiagnostics::Load[]	N	The load diagnostics.
loadsSize	Integer	N	The size of `loads`.
networks	VxDiagnostics::Network[]	N	The network diagnostics.
networksSize	Integer	N	The size of `networks`.
powers	VxDiagnostics::Power[]	N	The power supply diagnostics.

Field Name	Type	Req	Description
powersSize	Integer	N	The size of `powers`.
retention	VxDiagnostics::Retention	N	The retention diagnostics.
storages	VxDiagnostics::Storage[]	N	The storage diagnostics.
storagesSize	Integer	N	The size of `storages`.
temperatures	VxDiagnostics::Temperature[]	N	The temperature diagnostics.
temperaturesSize	Integer	N	The size of `temperatures`.

4.4 VxDiagnostics::Assignments

Description
Represents data session information.

Field Name	Type	Req	Description
bitrate	Float	N	Total bitrate that is currently assigned.
maxBitrate	Float	N	Maximum bitrate that may be assigned.
dataSources	Integer	N	Number of IVxDataSource currently assigned.
maxDataSources	Integer	N	Maximum number of IVxDataSource that may be assigned.

4.5 VxDiagnostics::BackupPower

Description
Represents backup power diagnostic information.

Field Name	Type	Req	Description
isEnabled	Boolean	N	True if currently on backup power, false otherwise.
id	String	N	Backup power identifier.
remaining	Integer	N	Estimated backup time remaining (in minutes). 0 if no backup is available.

4.6 VxDiagnostics::Database

Description
Represents database health information.

Field Name	Type	Req	Description
id	String	N	Database identifier.
statusProgress	Integer	N	Specifies how close the current status operation is to completion (e.g. rebuilding). Defaults to 0 if no ongoing status operation is in progress.
status	VxDatabaseStatus	N	Database health status.

4.7 VxDiagnostics::Eventing

Description
Represents eventing information.

Field Name	Type	Req	Description
eventRate	Integer	N	The average event rate (events per hour).

4.8 VxDiagnostics::Fan

Description
Represents fan specific diagnostic information.

Field Name	Type	Req	Description
id	String	N	Fan identifier.
isOk	Boolean	N	True if the fan status is ok, false if the fan is failed.
rpm	Integer	N	RPM of the fan.

4.9 VxDiagnostics::Hdd

Description
Represents hard disk drive (HDD) diagnostic information. Contains information on the health of individual hard disks.

Field Name	Type	Req	Description
id	String	N	Disk identifier.
isOk	Boolean	N	True if the HDD status is ok, false if the HDD is failed.
model	String	N	Drive model identifier.
status	VxHddStatus	N	Current drive health status.
statusProgress	Integer	N	Specifies how close the current status operation is to completion (e.g. rebuilding). Defaults to 0 if no ongoing status operation is in progress.

4.10 VxDiagnostics::Load

Description
Represents load diagnostic information. May be used to indicate load on a CPU, application, etc. The `type` will distinguish the entity being represented.

Field Name	Type	Req	Description
percent	Integer	N	Percentage being used.
type	VxLoadType	N	The type of load this resource is referencing.

4.11 VxDiagnostics::Network

Description
Represents network diagnostic information.

Field Name	Type	Req	Description
id	String	N	Network identifier.
bandwidthTotal	Integer	N	Total amount of bandwidth available (in kbps).
bandwidthUsed	Integer	N	Amount of bandwidth in use (in kbps).

4.12 VxDiagnostics::Power

Description
Represents power supply diagnostic information.

Field Name	Type	Req	Description
id	String	N	Power identifier.
isOk	Boolean	N	True if power supply is online, false otherwise.

4.13 VxDiagnostics::Retention

Description
Represents storage retention information.

Field Name	Type	Req	Description
retentionTime	Integer	N	Current media storage retention time (in hours).

4.14 VxDiagnostics::Storage

Description
Represents storage diagnostic information. Contains information on the capacity and usage of local storage.

Field Name	Type	Req	Description
id	String	N	Storage identifier.
isOnline	Boolean	N	True if storage is online and available, false otherwise.
total	Integer	N	Total amount of storage present (in MB).
used	Integer	N	Amount of storage in use (in MB).

4.15 VxDiagnostics::Temperature

Description
Represents temperature sensor diagnostic information.

Field Name	Type	Req	Description
degrees	Integer	N	Temperature reading (in Celsius).
id	String	N	Temperature identifier.
thresholdHigh	Integer	N	Temperature threshold over which normal operating temperature is exceeded (in Celsius).
thresholdLow	Integer	N	Temperature threshold under which normal operating temperature is exceeded (in Celsius).

4.16 VxExportStreamClip

Description
Represents streaming access to a single IVxExportClip within an IVxExport.

Field Name	Type	Req	Description
audioDataSourceId	String	N	The identifier of the audio IVxDataSource that the exported clip came from.
audioUrl	String	N	The RTSP URL for streaming the audio data of the export.
endTime	DateTime	Y	Time at which the export media ends.
renderType	VxRenderType	Y	The type of rendering required for the media data delivered by this interface. Clients can utilize this to create the correct rendering pipeline.
startTime	DateTime	Y	Time at which the export media begins.
videoDataSourceId	String	N	The identifier of the video IVxDataSource that the exported clip came from.
videoUrl	String	N	The RTSP URL for streaming the video data of the export.

4.17 VxInternalEvent

Description
Represents an internal event generated by the VxSDK.

Field Name	Type	Req	Description
eventType	VxInternalEventType	Y	The event type of this VxInternalEvent.
id	String	Y	Unique VxInternalEvent identifier.
properties	VxKvObject	N	Optional additional information related to the event.
propertySize	Integer	N	The size of `properties`.
systemId	String	Y	The unique identifier of the IVxSystem that generated this event.

4.18 VxKvObject

Description
Represents a key/value pair object.

Field Name	Type	Req	Description
key	String	Y	The object key.
value	String	Y	The object value.

4.19 VxLdapValidationCredentials

Description
Represents credentials that are used to verify the connection to the LDAP server. The information will not be stored.

Field Name	Type	Req	Description
baseDn	String	Y	The location in the LDAP database that will be used to search for and authenticate user entries. Example: `ou=Service Accounts, dc=Pelco, dc=org`
isServerTlsEnabled	Boolean	Y	Indicates whether or not SSL/TLS security is enabled for the LDAP server.
isSsoEnabled	Boolean	N	Indicates whether or not single sign-on is enabled.
isTwoStageBindingEnabled	Boolean	N	Indicates whether or not two stage binding authentication is enabled.
objectClasses	String	N	The LDAP object classes to search for users. Example: `user, inetOrgPerson`
password	String	N	The user password. Ignored if `isSsoEnabled` is `true`.
searchAttributes	String	Y	The attributes against which to match the user's identity/name. Example: `cn, mail`
searchDnAccount	String	N	The superuser DN account in the LDAP directory to use for performing searches there. Example: `CN=users, DC=mycorp`. If `isSsoEnabled` is `true`, this account should have write permissions on the `vxRootDn`. Should be supplied if `isTwoStageBindingEnabled` or `useLdapUsersAndRoles` are `true`; ignored otherwise.
searchDnPassword	String	N	The password for the superuser DN account in the LDAP directory that performs searches. Should be supplied if `isTwoStageBindingEnabled` or `useLdapUsersAndRoles` are `true`; ignored otherwise.
serverName	Host	Y	The hostname or IP address of the LDAP server.
serverPort	Integer	Y	Port number of the LDAP server.
ssoDomain	String	N	The Active Directory/Kerberos domain used for single sign-on. Ignored if `isSsoEnabled` is `false`.
useLdapUsersAndRoles	Boolean	N	Indicates whether or not LDAP will be used to manage users and roles. `false` to only use the LDAP server for authentication; the VideoXpert system will manage users and roles itself. `true` to manage user accounts and role assignments in the LDAP system. Note that permissions on roles will still be managed by the VideoXpert system.
username	String	N	The user login name.

Field Name	Type	Req	Description
vxRootDn	String	N	The LDAP container under which VideoXpert system information for all sites is stored; this should be the same for all sites. Should be supplied if `useLdapUsersAndRoles` is `true`; ignored otherwise.
vxSystemDn	String	N	The site specific LDAP container; unique per VideoXpert site. This DN should contain the `vxRootDn` (which is a suffix of this full DN). Should be supplied if `useLdapUsersAndRoles` is `true`; ignored otherwise.

4.20 VxLicenseSup

Description
Represents software upgrade plan licensing information.

Field Name	Type	Req	Description
catchUpCount	Integer	N	The number of channel-years needed to bring the software upgrade plan license up-to-date today.
currentBalance	Float	N	The current number of channel-years remaining.
events	VxLicenseSupEvent[]	N	List of events that are related to this license software upgrade plan; an audit trail.
eventsSize	Integer	N	The size of `events`.
expirationTime	DateTime	N	The expiration time of the software upgrade plan.
extensionYearCount	Integer	N	The number of channel-years needed to extend the software upgrade plan license for 1 year.

4.21 VxLicenseSupEvent

Description
Represents a specific event that occurred for a software upgrade plan.

Field Name	Type	Req	Description
currentBalance	Float	N	The current number of channel-years available (at the time of this event).
expirationTime	DateTime	N	The projected expiration of the software upgrade plan (at the time of this event).
time	DateTime	N	The time at which this event occurred.
type	VxSupEventType	N	The type of software upgrade plan event.
value	Float	N	A numerical value that provides additional data related to the type of event this is.

4.22 VxLimits

Description
Represents a map of limits related to a resource. A resource will supply a limit for any field that is supported but does not have a valid value available. The presence of a limit indicates that the associated field is supported, regardless of whether the field is present in the resource representation or not.

Field Name	Type	Req	Description
booleanLimitsSize	Integer	Y	The size of `booleanLimits`.
floatLimitsSize	Integer	Y	The size of `floatLimits`.
integerLimitsSize	Integer	Y	The size of `integerLimits`.
listLimitsSize	Integer	Y	The size of `listLimits`.
objectLimitsSize	Integer	Y	The size of `objectLimits`.
stringLimitsSize	Integer	Y	The size of `stringLimits`.
booleanLimits	VxLimits::Boolean[]	N	The collection of boolean limits.
floatLimits	VxLimits::Float[]	N	The collection of float limits.
integerLimits	VxLimits::Integer[]	N	The collection of integer limits.
listLimits	VxLimits::List[]	N	The collection of list limits.
objectLimits	VxLimits::Object[]	N	The collection of object limits.
stringLimits	VxLimits::String[]	N	The collection of string limits.

4.23 VxLimits::Boolean

Description
Represents the valid values for a boolean field.

Field Name	Type	Req	Description
defaultValue	Boolean	N	The field's default value.
fieldName	String	Y	The name of the field that this boolean limit applies to.
hasDefaultValue	Boolean	Y	`true` if a `defaultValue` has been set, otherwise `false`.
options	Boolean[]	N	A list of valid values that may be applied to the field. If no values are present, this indicates that the field is currently read-only.
optionsSize	Integer	Y	The size of `options`.

4.24 VxLimits::Float

Description
Represents the valid values for a numeric float field.

Field Name	Type	Req	Description
defaultValue	Float	N	The field's default value.
fieldName	String	Y	The name of the field that this float limit applies to.
hasDefaultValue	Boolean	Y	`true` if a `defaultValue` has been set, otherwise `false`.
hasMax	Boolean	Y	`true` if a `max` has been set, otherwise `false`.

Field Name	Type	Req	Description
hasMin	Boolean	Y	`true` if a `min` has been set, otherwise `false`.
max	Float	N	The field's maximum valid value.
min	Float	N	The field's minimum valid value.

4.25 VxLimits::Integer

Description
Represents the valid values for a numeric integer field.

Field Name	Type	Req	Description
defaultValue	Integer	N	The field's default value.
fieldName	String	Y	The name of the field that this integer limit applies to.
hasDefaultValue	Boolean	Y	`true` if a `defaultValue` has been set, otherwise `false`.
hasMax	Boolean	Y	`true` if a `max` has been set, otherwise `false`.
hasMin	Boolean	Y	`true` if a `min` has been set, otherwise `false`.
max	Integer	N	The field's maximum valid value.
min	Integer	N	The field's minimum valid value.

4.26 VxLimits::List

Description
Represents the valid values for an array field.

Field Name	Type	Req	Description
fieldName	String	Y	The name of the field that this list limit applies to.
hasMaxItems	Boolean	Y	`true` if a `maxItems` has been set, otherwise `false`.
hasMinItems	Boolean	Y	`true` if a `minItems` has been set, otherwise `false`.
maxItems	Integer	N	The maximum number of items that may appear in the list.
minItems	Integer	N	The minimum number of items that may appear in the list.

4.27 VxLimits::Object

Description
Represents the valid values for an object field.

Field Name	Type	Req	Description
fieldName	String	Y	The name of the field that this object limit applies to.

4.28 VxLimits::String

Description
Represents the valid values for a string field.

Field Name	Type	Req	Description
defaultValue	String	N	The field's default value.
fieldName	String	Y	The name of the field that this string limit applies to.
hasDefaultValue	Boolean	Y	`true` if a `defaultValue` has been set, otherwise `false`.
options	String[]	N	A list of valid values that may be applied to the field. If no values are present, this indicates that the field is currently read-only.
optionsSize	Integer	Y	The size of `options`.

4.29 VxLinkedPtzInfo

Description
Represents linked PTZ information.

Field Name	Type	Req	Description
ip	IP	Y	The IP of the tracking data source.
name	String	N	The friendly name of the tracking data source.

4.30 VxLoginInfo

Description
Represents the information needed to log in to a VideoXpert system.

Field Name	Type	Req	Description
authToken	String	N	The authentication token to use.
ipAddress	IP	Y	The VideoXpert system IP.
licenseKey	String	Y	The license key.
password	String	Y	The password to log in with.
port	Integer	Y	The VideoXpert system port.
username	String	Y	The username to log in with.
useSsl	Boolean	N	Indicates whether the connection will use SSL.

4.31 VxMonitorPosition

Description
Represents a monitor position that describes where a IVxMonitor resides in a coordinate plane (specifically the IV quadrant of a Cartesian plane where 0,0 is the top left point).

Field Name	Type	Req	Description
monitorId	String	Y	The unique identifier of the IVxMonitor that this position represents.
position	VxRect	Y	The position and size of the IVxMonitor.

4.32 VxMonitorSelection

Description
Represents information for a selected monitor/cell within a IVxMonitorWall. Each user can independently select a different monitor/cell for remote-control of a IVxMonitorWall.

Field Name	Type	Req	Description
cell	Integer	N	The index of the selected IVxMonitorCell within the selected IVxMonitor.
inputMode	VxCellInputMode	N	The input mode of the selected IVxMonitorCell.
lastModified	DateTime	N	The time of the last modification for this monitor selection.
monitor	Integer	N	The index of the selected IVxMonitor.
owner	UPN	N	The name of the user that owns this monitor selection.

4.33 VxNewAnalyticBehavior

Description
Represents a request for the creation of a new analytic behavior.

Field Name	Type	Req	Description
behaviorType	VxAnalyticBehaviorType	Y	The type of analytic behavior being performed.
id	String	N	The unique identifier of the analytic behavior.
isEnabled	Boolean	N	Indicates whether or not this analytic behavior is enabled.
name	String	N	The friendly name of the analytic behavior.
objectInZone	VxObjectInZone	N	The object zone data used to configure analytics of `behaviorType` is set to `kObjectInZone`.
objectLineCounter	VxObjectLineCounter	N	The object counter data used when `behaviorType` is set to `kObjectLineCounter`.
objectType	VxAnalyticObjectType	Y	The type of object this analytic pertains to.
severity	Integer	N	The severity value for events generated from this analytic behavior, from 1 (highest) to 10 (lowest). If set, overrides the corresponding situation severity.

4.34 VxNewAnalyticSession

Description
Represents a request for the creation of a new analytic session.

Field Name	Type	Req	Description
dataEncodingId	String	N	The unique identifier specifying which data encoding to use as a source. This value can be used instead of `source` to determine which stream to use.
dataSourceId	String	Y	The unique identifier of the video data source for this analytic session. The source URI should be pointing to an RTSP stream on this IVxDataSource.
deviceId	String	Y	The unique identifier of the Device hosting this analytic session. The device must be of type `kAnalyticServer`.
id	String	N	The unique identifier of the analytic session. The server will generate this value if not supplied.
source	String	N	The RTSP URI to use as the source for the analytic session. This value can be used instead of `dataEncodingId` to determine which stream to use. This value takes priority over the `dataEncodingId` if both values are provided. Note: This field is required when sending to a `VxDeviceType::kAnalyticServer`.

4.35 VxNewBookmark

Description
Represents a request for the creation of a new IVxBookmark.

Field Name	Type	Req	Description
dataSourceId	String	Y	Unique identifier of the IVxDataSource to associate with the IVxBookmark.
description	String	N	Friendly description.
groupId	String	N	IVxBookmark group identifier (supplied by clients). Typically used to identify related bookmarks (such as those bookmarking the same time on audio and video). Recommend using GUID [RFC 4122].
lockEndTime	DateTime	N	The end time of the media to lock.
lockStartTime	DateTime	N	The start time of the media to lock.
name	String	N	Friendly name.
time	DateTime	Y	Time at which the point of interest occurred.

4.36 VxNewClip

Description
Represents request to create a new IVxClip on a IVxDataStorage using data from another IVxDataStorage. The new IVxClip will have the same VxRecordingType as the data it is based on, defaulting to `kTimed`. Typically this is used to save data from a camera supporting edge storage to a recorder.

Field Name	Type	Req	Description
dataSourceId	String	Y	The unique identifier of the IVxDataSource that generated the desired data.
dataStorageId	String	Y	The unique ID of the IVxDataStorage to retrieve the data from. The host IVxDevice MUST be of VxDeviceType `kCamera` (it must be a camera supporting edge storage).
endTime	DateTime	Y	Time at which to end the clip.
startTime	DateTime	Y	Time at which to begin the clip.

4.37 VxNewDataObject

Description
A VxNewDataObject contains the information needed for a client to submit a request to the server for the creation of a new IVxDataObject.

Field Name	Type	Req	Description
clientType	String	Y	IVxDataObject client identifier. Recommend using the Java package naming convention: `com.<company>.<project>.<clientName>`.
data	String	Y	Serialized data object (e.g.: JSON, XML, CSV, etc). The server MUST NOT utilize this data in any way. The maximum allowable size of this field is 1 MB.
isPrivate	Boolean	N	Indicates whether this resource is owned by a IVxUser. If true, the IVxDataObject will be owned by the IVxUser submitting the request.

4.38 VxNewDevice

Description
Represents a request for the creation of a new IVxDevice.

Field Name	Type	Req	Description
driverType	String	N	Type identifier of the IVxDriver to use for the IVxDevice when assigning it to the IVxDataStorage specified by `dataStorageId`. If no type is provided, a driver will be selected automatically. Ignored if `dataStorageId` is not provided.
endpoints	String[]	N	List of source URIs that the IVxDevice will support. If the VxDeviceType of this IVxDevice is `kGeneric` then at least 1 URI MUST be provided. The URI MUST be unique, otherwise `kConflict` will be returned.
endpointsSize	Integer	N	The size of `endpoints`.
host	Host	N	Host address of the IVxDevice. If set, takes precedence over the `ip`. The combination of `host`, `port` and `type` MUST be unique, otherwise `kConflict` will be returned. *Note: One of either `host` or `ip` MUST be provided.*
id	String	N	Unique IVxDevice identifier. Recommend using GUID [RFC 4122]. If not unique to the sytem, a `kConflict` will be returned. If not provided, the server will create this value.
ip	IP	N	DEPRECATED Use `host` instead. Primary IP address of the IVxDevice. Ignored if `host` is also provided. The combination of `ip`, `port` and `type` MUST be unique, otherwise `kConflict` will be returned.
model	String	N	Product model name.
name	String	N	Friendly name.
password	String	N	Account password used to communicate with the IVxDevice, if any.
port	Integer	N	The host port, if non-default. The combination of `host`, `ip`, `port` and `type` MUST be unique, otherwise `kConflict` will be returned.
serial	String	N	Serial number.

Field Name	Type	Req	Description
shouldAutoCommission	Boolean	N	True to have this IVxDevice automatically commissioned when created. Defaults to `False`.
type	VxDeviceType	Y	The particular type of the IVxDevice.
username	String	N	Account username used to communicate with the IVxDevice, if any.
vendor	String	N	IVxDevice vendor, if available.
version	String	N	The current version of the IVxDevice.

4.39 VxNewDeviceAssignment

Description
A VxNewDeviceAssignment represents a IVxDevice assignment request that MAY specify a IVxDriver to utilize to communicate with the IVxDevice.

Field Name	Type	Req	Description
dataSourceIds	String[]	N	DEPRECATED This field is undefined and is only present for legacy reasons; it SHOULD NOT be used.
dataSourceIdSize	Integer	N	The size of `dataSourceIds`.
deviceId	String	Y	The unique ID of the IVxDevice to assign.
volumeGroupId	String	N	The unique identifier of the IVxVolumeGroup to use for this IVxDevice. If not provided, a IVxVolumeGroup will be selected automatically.

4.40 VxNewDrawing

Description
A VxNewDrawing contains the information needed for a client to submit a request to the server for the creation of a new IVxDrawing.

Field Name	Type	Req	Description
name	String	Y	Friendly name.

4.41 VxNewEvent

Description
Represents a request for a new IVxEvent to be generated. The IVxEvent will be generated based upon the VxNewEvent data, its IVxSituation configuration, and will receive a unique ID.

Field Name	Type	Req	Description
generatorDeviceId	String	N	Unique ID of the IVxDevice that generated this IVxEvent. This field MAY be omitted if the generator device is the same as the source device. Note that this device MAY be an external device that is not represented in the system.
id	String	N	Unique IVxEvent identifier. Recommend using GUID [RFC 4122]. If not unique to the sytem, a `kConflict` will be returned. If not provided, the server will create this value.

Field Name	Type	Req	Description
properties	VxKvObject	N	A VxKvObject, specific to the VxSituationType, containing additional information related to the IVxEvent. See the Situations table for the properties required of the IVxEvent. Only properties defined for this situation type SHOULD be used.
propertySize	Integer	N	The size of `properties`.
situationType	String	Y	Identifier for the type of IVxSituation that led to this VxNewEvent.
sourceDeviceId	String	Y	Unique ID of the device that the situation occurred on. Note that this device MAY be an external device that is not represented in the system.
time	DateTime	Y	Time at which the situation occurred.

4.42 VxNewExport

Description
Represents a new IVxExport

Field Name	Type	Req	Description
clips	VxNewExportClip[]	N	Individual ranges of media data that shall be included in the data archived when this IVxExport is triggered.
clipSize	Integer	N	The size of `clips`.
format	VxExportFormat	Y	The format of the export data.
name	String	N	Friendly name.
password	String	N	Set a password for the IVxExport. If set, the export data will be signed and encrypted. This password will be required to decrypt the export data. If not set, the export data will not be signed or encrypted.

4.43 VxNewExportClip

Description
Represents a new IVxExportClip

Field Name	Type	Req	Description
dataEncodingId	String	N	Specify the specific data encoding to export (quality, framerate, and resolution). This can be obtained from the desired IVxDataInterface of the IVxClip. If not specified, the server shall select the data encoding.
dataSourceId	String	Y	The IVxDataSource ID of the export media. Only IVxDataSource of type `kVideo` are allowed.
endTime	DateTime	Y	Time at which the export media ends.
startTime	DateTime	Y	Time at which the export media begins.

4.44 VxNewManualRecording

Description
DEPRECATED: Replaced by VxNewRecording Represents a request for a new IVxManualRecording to be generated.

Field Name	Type	Req	Description
dataSourceId	String	Y	Identifier of the IVxDataSource to manually record.
postRecord	Integer	N	Amount of time, from `0` to `30` seconds, to record after the manual recording is stopped.
preRecord	Integer	N	Amount of time, from `0` to `30` seconds, to record prior to the manual recording start time.

4.45 VxNewMarker

Description
Represents a request for a new IVxMarker to be generated.

Field Name	Type	Req	Description
associatedDataSourceId	String	N	Identifier of the IVxDataSource to associate with the IVxMarker.
direction	Float	N	Angular coordinate indicating the direction, if any, that the IVxMarker is facing on a polar grid (e.g.: 0[right], 90[up], 180[left], 270[down]).
layerName	String	N	Friendly name of the layer that this IVxMarker is on. Clients MAY group IVxMarker with matching layer names.
name	String	N	Friendly name.
resourceRef	VxResourceRef	N	A reference to a resource, if any, that this IVxMarker is associated with (null if there is no association). If an attempt is made to set this to an invalid resource, `kInvalidValue` is returned.
x	Float	Y	X Cartesian coordinate.
y	Float	Y	Y Cartesian coordinate.

4.46 VxNewMember

Description
Represents a request to add a new IVxMember system.

Field Name	Type	Req	Description
bandwidth	Integer	Y	Bandwidth available between the aggregator system and the member system (in kbps).
host	Host	Y	The host address of the system.
password	String	Y	Password for access to the member.
port	Integer	Y	The host port of the system.

Field Name	Type	Req	Description
rtspCapability	VxRtspCapability	N	The network streaming capabilities that the member shall expose to its clients (via IVxDataInterface).
username	String	Y	Username for access to the member.

4.47 VxNewMonitor

Description
Represents a request to create a new IVxMonitor.

Field Name	Type	Req	Description
hostDeviceId	String	Y	Identifier of the IVxMonitor host IVxDevice.
layout	VxCellLayout	N	Cell grid layout.
name	String	N	Friendly name.
number	Integer	N	A unique number used to designate the IVxMonitor.

4.48 VxNewNotification

Description
Represents a new notification configuration for a IVxSituation.

Field Name	Type	Req	Description
roleIds	String[]	N	IDs of the roles for which the constituent users should receive this notification.
roleIdSize	Integer	N	The size of `roleIds`.

4.49 VxNewPixelSearch

Description
Represents a square grid composed of rows and columns with zones indicating the areas of the grid to search. The grid uses a cartesian coordinate system (see: Pixel Search Addendum) with 0-based indices. The search results will be contained by the new IVxPixelSearch that is created.

Field Name	Type	Req	Description
columns	Integer	Y	Number of columns in the search grid (min: `1`, max: `1000`).
rows	Integer	Y	Number of rows in the search grid (min: `1`, max: `1000`).
searchEndTime	String	N	Inclusive bounds to the earliest results to retrieve. Defaults to the oldest results available.
searchStartTime	String	N	Inclusive bounds to the latest results to retrieve. Defaults to the most current results available.
zones	VxRect[]	Y	List of rectangular areas indicating the selected zones to search.
zoneSize	Integer	N	The size of `zones`.

4.50 VxNewPrivilege

Description
Represents a new IVxPrivilege.

Field Name	Type	Req	Description
permission	VxPermissionId	Y	Identifier of the permission being granted by the IVxPrivilege.
priority	Integer	N	Relative priority for the IVxPrivilege, from 1 to 250. Smaller numbers have higher priority than larger numbers (e.g. 1 is the highest priority).

4.51 VxNewRecording

Description
Represents a request for a new IVxRecording to be generated.

Field Name	Type	Req	Description
dataSourceId	String	Y	Identifier of the IVxDataSource to record.
endEvent	String	N	If specified, the recording will automatically end when a matching event occurs.
endEventSource	VxResourceRef	N	If specified, only events from the specified source will end the recording.
framerate	VxRecordingFramerate	N	The framerate to record at.
id	String	Y	Unique IVxRecording identifier.
maxRecordingTime	Integer	N	Maximum amount of time, in seconds, to record before stopping.
name	String	N	Friendly name of the recording which can be used to identify recordings to stop.
postRecord	Integer	N	Amount of time, in seconds, to record after the recording would otherwise be stopped.
preRecord	Integer	N	Amount of time, in seconds, to record prior to the recording start time.
recordType	VxRecordingType	N	The type of this recording which will show up in the corresponding clips.
startTime	DateTime	N	Time at which the recording should begin (may be in the recent past). The recording `preRecord` and `maxRecordingTime` will reference to this time. Defaults to the time at which the server processes the request.

4.52 VxNewRule

Description
Represents a request for the creation of a new IVxRule.

Field Name	Type	Req	Description
isEnabled	Boolean	N	Enable/disable the new IVxRule.
name	String	N	Friendly name.
script	String	N	Script to run when the IVxRule triggers. If this script is triggered by an IVxEvent, its properties will be available in the script `eventProperties` variable.
timeTableIds	String[]	N	The new IVxRule will only automatically run its script if a trigger occurs during the times contained by a IVxTimeTable specified here. If empty, no active time ranges are available for this IVxRule (it must be manually triggered). If `null`, no time filters will be applied (the new IVxRule is always active).
timeTableIdSize	Integer	N	The size of `timeTableIds`.
triggers	VxRuleTrigger[]	N	List of VxRuleTrigger that, when any activate, cause the IVxRule to run its script (if activated during an active time).
triggerSize	Integer	N	The size of `triggers`.

4.53 VxNewSchedule

Description
Represents a new IVxSchedule. A IVxSchedule is a group of 0 or more resources associated with a set of time and/or event based IVxScheduleTrigger that, when *any* are active, cause an action to be performed (depending on the type of IVxScheduleTrigger). See Schedule Diagrams for further information.

Field Name	Type	Req	Description
action	VxScheduleAction	Y	Action to perform when the IVxSchedule is active.
id	String	N	Provide the unique IVxSchedule identifier. Recommend using GUID [RFC 4122]. If not supplied, the server shall create the identifier.
name	String	N	Friendly name.
scheduleTriggers	VxNewScheduleTrigger[]	Y	A list of VxNewScheduleTrigger to create for the new IVxSchedule. Maximum of 16 IVxScheduleTrigger may be created.
scheduleTriggerSize	Integer	N	The size of `scheduleTriggers`.
useAllDataSources	Boolean	N	True indicates that the IVxSchedule shall apply to all IVxDataSources regardless of what IVxDataSources are linked to the IVxSchedule (see: `Link` method). False indicates that only linked IVxDataSources shall be associated with this IVxSchedule.

4.54 VxNewScheduleTrigger

Description
Represents a new IVxScheduleTrigger

Field Name	Type	Req	Description
action	VxScheduleAction	N	The action performed when the IVxScheduleTrigger activates.

Field Name	Type	Req	Description
eventProperties	KVObject	N	If set, the IVxEvent only activates when it occurs with these properties. Ignored if no `eventSituationType` is set.
eventPropertySize	Integer	N	The size of `eventProperties`.
eventSituationType	String	N	If set, the IVxScheduleTrigger only activates when this type of event occurs. The IVxScheduleTrigger event state shall be considered active until the event becomes inactive.
framerate	VxRecordingFramerate	N	Framerate level to record at.
id	String	N	Provide the unique IVxScheduleTrigger identifier. Recommend using GUID [RFC 4122]. If not supplied, the server shall create the identifier.
inactiveEventSituationType	String	N	If set, the IVxScheduleTrigger event state will no longer be considered active when this type of event occurs. If not set, the IVxScheduleTrigger event status will immediately be considered inactive after the trigger event occurs. *Note: This field is ignored if `eventSituationType` is not set.*
postTrigger	Integer	N	Amount of time, from `0` to `300` seconds, to continue to consider the IVxScheduleTrigger active when it becomes inactive ("post alarm").
preTrigger	Integer	N	Amount of time, from `0` to `30` seconds, to consider the IVxScheduleTrigger active prior to when it becomes active ("pre alarm").
timeout	Integer	N	Amount of time, from `1` to `300` seconds, to consider the IVxScheduleTrigger active immediately after it becomes active ("duration recording"). The IVxScheduleTrigger will become inactive when this time has elapsed (though the `postTrigger` may continue to keep it active at this point). No timeout is applied if the value is less than `1`.
timeTableId	String	N	If set, the IVxScheduleTrigger may only be active during the time range(s) defined by this IVxTimeTable.

4.55 VxNewSituation

Description
Represents a new IVxSituation configuration that specifies how corresponding events shall be generated and handled when the situation is detected. A IVxSituation is uniquely identified by both its `type` and `sourceDeviceId`. See Situation Defaults for the default attributes of a IVxSituation when not supplied here.

Field Name	Type	Req	Description
audibleLoopDelay	Integer	N	Seconds to wait between audible notifications (see: `audiblePlayCount`.

Field Name	Type	Req	Description
audiblePlayCount	Integer	N	Play audible notification this many times, separated by the `audibleLoopDelay`.
autoAcknowledge	Integer	N	Number of seconds after which a generated event VxAckState will be set to `kAutoAcked`. If less than 0, a generated event must be manually acknowledged. If 0, a generated event must be set to `kAutoAcked` immediately (prior to any notifications being sent).
isAckNeeded	Boolean	N	If true, generated events shall have an initial VxAckState of `kAckNeeded`. If false, generated events shall have an initial VxAckState of `kNoAckNeeded`.
name	String	N	Friendly name.
severity	Integer	N	Severity of the generated IVxEvent, from 1 (highest) to 10 (lowest).
shouldAudiblyNotify	Boolean	N	True specifies that a notification sound is to play on supporting clients when receiving a notification for IVxEvents corresponding to the IVxSituation.
shouldExpandBanner	Boolean	N	True if the notification banner on clients should be expanded by default.
shouldLog	Boolean	N	If true, events generated from this IVxSituation shall be persisted as long as possible. If false, generated events shall immediately be discarded; unlogged events are hidden from clients (this supersedes all other situation configuration).
shouldNotify	Boolean	N	If true, an IVxEvent generated from the IVxSituation shall generate notifications that are sent to authorized clients, per the notification configuration of the IVxSituation, subscribed to the IVxSystem. Additionally, these notifications will be sent out whenever generated events have a change of `ackState`.
shouldPopupBanner	Boolean	N	True if clients should display a popup notification banner when receiving events of this type (default).
snoozeIntervals	Integer[]	N	List of default snooze intervals, in seconds, for a generated IVxEvent. Note that these are default options and that they do *not* limit the amount of time a generated IVxEvent may be snoozed for.
snoozeIntervalSize	Integer	N	The size of `snoozeIntervals`.
sourceDeviceId	String	N	Together with `type`, is a unique IVxSituation identifier. This field acts an optional constraint on the source of events for this IVxSituation. If specified, any events matching the IVxSituation `type` MUST also match this `sourceDeviceId` in order for the IVxSituation to apply.
type	String	Y	Together with `sourceDeviceId`, is a unique IVxSituation identifier. MUST be of the form `external/<company>/<event>` where `<company>` and `<event>` are UTF-8 strings no greater than 64 characters each; forward slashes are not allowed. These strings describe the `<company>` that manufactured the device that was the source of the `<event>` (e.g.: `external/pelco/swipe`).

4.56 VxNewTag

Description
Represents a request for a new IVxTag to be generated.

Field Name	Type	Req	Description
isFolder	Boolean	N	If true, the new IVxTag will be a folder IVxTag.

Field Name	Type	Req	Description
isPublic	Boolean	N	If true, indicates that this IVxTag is not owned (public). If false, indicates that this IVxTag is owned (private) by a IVxUser. If this IVxTag has an owner, only that owner and users with appropriate permissions will be able to read it. Not applicable if this is a folder IVxTag.
name	String	Y	Unique IVxTag friendly name. Commas are invalid characters for this field and MUST NOT be used.
parentId	String	N	Identifier of the IVxTag that is the parent of this IVxTag. If this is provided on a non-folder IVxTag, it will be ignored. If used, the `parentId` MUST refer to an existing folder IVxTag.

4.57 VxNewTimeTable

Description
Represents a request to create a new IVxTimeTable.

Field Name	Type	Req	Description
endDate	String	N	The IVxTimeTable will be considered inactive after this date.
id	String	N	Provide the unique IVxTimeTable identifier. Recommend using GUID [RFC 4122]. If not supplied, the server shall create the identifier.
name	String	Y	Friendly name
startDate	String	N	The IVxTimeTable will be considered inactive before this date.
weeklyTimeRanges	VxTimeRange[]	N	Active time ranges for the new IVxTimeTable. The list is sorted by `day` ascending (monday through sunday), then by `startTime` ascending, and finally by `endTime` ascending.
weeklyTimeRangeSize	Integer	N	The size of `weeklyTimeRanges`.

4.58 VxNewUser

Description
Used to request a new IVxUser.

Field Name	Type	Req	Description
canBypassLdap	Boolean	N	If true, the IVxUser can login to the system using local credentials instead of using the LDAP authentication.
domain	String	N	Network domain for the IVxUser. If not provided, defaults to `LOCAL`.
email	String	N	Email address of the IVxUser.
employeeId	String	N	Employee badge (or other) personnel identifier associated with the new IVxUser.
firstName	String	N	First name of IVxUser.
isPasswordExpirationDisabled	Boolean	N	If true, password expiration will be disabled for this IVxUser; false will use the global password expiration setting.

Field Name	Type	Req	Description
lastName	String	N	Last name of IVxUser.
mustChangePassword	Boolean	Y	If true, the new user will be forced to change their password the first time they log in.
name	SASLString	Y	The unique name of the user that this resource is representing. This name will be the same as the username that is used to access the VideoXpert system. If the name already exists (case-insensitive) or is restricted, `kConflict` will be returned.
note	String	N	Supplemental information about the new IVxUser.
password	SASLString	Y	The password to associate with the user. If the system is using an external LDAP server, this field will be ignored. MUST contain more than 7 characters.
phoneNumbers	VxPhoneNumber[]	N	Telephone number(s) for the new IVxUser. Maximum of 16 numbers.
phoneNumberSize	Integer	N	The size of `phoneNumbers`.

4.59 VxNewVolume

Description
Represents a new IVxVolume to be created.

Field Name	Type	Req	Description
buffer	Float	N	The percentage of the volume to keep free.
domain	String	N	The domain for the volume network path. Note: Ignored if the `path` is not a UNC path.
isBandwidthReserved	Boolean	N	If true, indicates that this is a NAS volume that shares bandwidth with data streams (bandwidth will be reserved for NAS operations).
password	String	N	The password for access to the network path. Note: Ignored if the `path` is not a UNC path.
path	String	Y	The fully qualified Windows directory path.
username	String	N	The username for access to the network path. Note: Ignored if the `path` is not a UNC path.

4.60 VxNewVolumeGroup

Description
Represents a new IVxVolumeGroup to be created.

Field Name	Type	Req	Description
isArchiveGroup	Boolean	N	If true, indicates that this volume group is an `archive volume group`, which will act as storage for the oldest available data.
name	String	Y	Friendly name.

4.61 VxObjectInZone

Description
Represents the configuration for an object in zone type of IVxAnalyticBehavior.

Field Name	Type	Req	Description
vertices	VxPoint[]	Y	Ordered list of VxPoint that make up the zone. Must contain at least 3 VxPoint, which together specify the closed polygonal zone within which to look for objects of the prescribed type.
verticesSize	Integer	Y	The size of `vertices`.

4.62 VxObjectLineCounter

Description
Represents the configuration for an object line counter type of IVxAnalyticBehavior.

Field Name	Type	Req	Description
endPoint	VxPoint	Y	The ending coordinate for the object counting line.
eventsEnabled	Boolean	N	If true, line counter specific events for this line counter will be enabled.
eventWindowSecs	Integer	N	Define the event's time window for testing if the counts maximum threshold is met within this window. For example, if `maxCountThreshold` is `100` and `eventWindowSecs` is `3600` then an event would be generated if 100 objects cross a counting line over the last hour. Once the hour elapses the event trigger's base count is reset and the count window slides to the next hour and starts over. Ignored if `maxCountThreshold` is not set.
leftCountLabel	String	N	Friendly name associated with a bidirectional counter's left count.
lineCounterType	VxLineCounterType	N	Defines the type of line counter (i.e. unidirectional, bidirectional, or omnidirectional). The type defines how the line counter increments and maintains it's counts.
maxCountThreshold	Integer	N	The maximum count threshold that must be met before an event is triggered. This value is used by both the bidirectional capacity events and the sliding window events.
rightCountLabel	String	N	Friendly name associated with a bidirectional counter's right count.
startPoint	VxPoint	Y	The starting coordinate for the object counting line.

4.63 VxPhoneNumber

Description
Represents a telephone number for a IVxUser.

Field Name	Type	Req	Description
number	String	N	String value representing the phone number.
type	VxPhoneType	Y	Category of phone number.

4.64 VxPoint

Description
Represents a location on a Cartesian grid.

Field Name	Type	Req	Description
x	Integer	Y	The X-axis coordinate value.
y	Integer	Y	The Y-axis coordinate value.

4.65 VxPtzLimits

Description
Represents the value limits for an PTZ device. *Note: The min / max speed values are for informational purposes only. They should not be used in any way to determine the* `speedX` *and* `speedY` *values for continuous movement in IVxPtzController.*

Field Name	Type	Req	Description
minPositionY	Integer	N	The Y coordinate minimum limit.
maxPositionY	Integer	N	The Y coordinate maximum limit.
maxPositionZ	Integer	N	The Z coordinate maximum limit.
minSpeedX	Integer	N	The pan speed minimum limit.
maxSpeedX	Integer	N	The pan speed maximum limit.
minSpeedY	Integer	N	The tilt speed minimum limit.
maxSpeedY	Integer	N	The tilt speed maximum limit.

4.66 VxRect

Description
Represents rectangular integer coordinates indicated by a combination of left, top, width, and height values.

Field Name	Type	Req	Description
height	Integer	N	The height value.
left	Integer	N	The left value.
top	Integer	N	The top value.
width	Integer	N	The width value.

4.67 VxResolution

Description
Represents a resolution in pixels.

Field Name	Type	Req	Description
height	Integer	N	The height in pixels.
width	Integer	N	The width in pixels.

4.68 VxResourceRef

Description
Represents a reference to a resource on the system (i.e. a device, datasource, tag, etc.).

Field Name	Type	Req	Description
id	String	N	The unique resource identifier. This must match exactly with the unique identifier of the resource being referenced.
type	VxResourceType	N	The type of resource being referenced.

4.69 VxRetention

Description
Represents retention information for a referenced resource.

Field Name	Type	Req	Description
oldestRecording	DateTime	Y	The time of the oldest recording for the resource.
resourceRef	VxResourceRef	Y	The reference to the resource.

4.70 VxRuleTrigger

Description
Represents a condition in a IVxRule that, if true, causes the IVxRule to run its script (if triggered during an active time of the IVxRule).

Field Name	Type	Req	Description
situationType	String	N	The VxRuleTrigger will be checked each time this type of IVxSituation occurs. If empty, the VxRuleTrigger will always be inactive.

Field Name	Type	Req	Description
sourceRef	VxResourceRef[]	N	The VxResourceRef that this trigger is associated with. Limited to one resource only. If empty, no source filter will be applied (all sources are valid). Supported resources are limited to the following VxResourceType values: `kDataSource`, `kDevice`, `kDataStorage`, and `kTag`. The VxRuleTrigger will evaluate to true only if its event source matches one of these sources. If the source is a `kdevice`, this will match against the IVxEvent `sourceDeviceId`. If the source is a `kDataSource`, this will match against the `service_property_id` value within the IVxEvent `properties`. Finally, if a source is a `kTag`, this will match against the IVxTag applied to the IVxSituation `serviceType`.
sourceRefSize	Integer	N	The size of `sourceRef`.

4.71 VxSmtpInfo

Description
Represents the SMTP information needed to send an email.

Field Name	Type	Req	Description
fromAddress	String	N	The "Mail From" email address.
host	Host	Y	The SMTP host.
password	String	N	The password for SMTP access.
smtpPort	Integer	Y	The SMTP port.
timeout	Integer	N	The timeout in milliseconds.
toAddress	String	Y	The "Mail To" email address.
username	String	N	The username for SMTP access.
useSmtps	Boolean	N	Indicates whether SMTPS should be used instead of SMTP.

4.72 VxSnapshotFilter

Description
Represents a snapshot filter to be used when taking a snapshot.

Field Name	Type	Req	Description
key	VxSnapshotFilterItem	Y	The filter key.
value	String	Y	The filter value.

4.73 VxTimeRange

Description
Represents rectangular integer coordinates indicated by a combination of left, top, width, and height values.

Field Name	Type	Req	Description
day	VxDayOfWeek	Y	The VxDayOfWeek that this time range applies to.
endTime	TimeOfDay	Y	The time at which this range ends (inclusive), from `00:00:00` to `23:59:59`.
startTime	TimeOfDay	Y	The time at which this range begins (inclusive), from `00:00:00` to `23:59:59`.

Interfaces

This section defines the interfaces that constitute the resources available within the VideoXpert SDK. For any resource provided, servers MUST support all *required* fields and methods. Servers MAY provide support for *optional* fields and methods. If a server does not support the functionality required by an optional field or method, or a client does not have authorization to it, the server will omit the field. Clients MUST provide the value for all *required* fields. The field values for a resource are populated during the creation of the resource. If a resource provides a `Refresh` method it can be used to update these values for the given resource.

The columns that make up the representation descriptions have the following meanings:

- **Field Name**: The name of the field within the VideoXpert SDK.
- **Type**: Primitive type of the value contained in this field.
- **Req**: Indicates whether this field is required to be present in responses sent by the server to the client. If not required, the server will omit this field when the client does not have authorization to the field or the server does not support the functionality exposed by the field.
- **RW**: Indicates whether this field is read-only (R), write-only (W), or read-write (RW). Read-only fields, on a server-hosted resource, can not be modified by a client (i.e. the client can not edit them). For fields on non-served resources, this column is not-applicable; there are no read/write restrictions on these. Note: A writable field must be set through its corresponding set method. Changing value of the field itself will NOT update the value on the server.
- **Description**: A brief description of the field.

The columns that make up the method specifications have the following meanings:

- **Method**: The name of the method within the VideoXpert SDK.
- **Description**: A brief description of the action performed by the method.

Collection filters are provided on select resources. Where provided, a list of available collection filter items will be displayed next to the description of the representation. Except when explicitly stated otherwise, each filter item matches up exactly, in both type and meaning, with the field of the same name in the object it is being used as a filter for (i.e. the `kState` filter item matches the `state` field). These filter items can be used whenever retrieving a collection of the resource in order to reduce the set of results down to a desired subset of the total. These filter items may be used in combination and, when doing so, are “and”ed together.

5.1 Global

Description	Collection Filters
Global methods available within the `VxSdk` namespace.	None.

Method	Description
VxSystemLogin(const VxLoginInfo& loginInfo, IVxSystem*& system)	Logs in to the VideoXpert system.
IsSupportedSystem(const char* ipAddress, int port, bool& isSupported, bool useSsl = true)	Checks whether the system at the specified IP address is supported by the VideoXpert SDK.
VxSetLogLevel(VxLogLevel::Value logLevel)	Sets the minimum severity level of messages to log.
VxSetLogPath(const char* logPath)	Sets the output path for log files.

5.2 IVxAccessPoint

Description	Collection Filters
Represents an access control service (e.g. a door control or badge reader), typically hosted by an access controller IVxDevice.	**kAdvancedQuery** **kHasStatus** **kId** **kModifiedSince** Only IVxAccessPoint that have been modified since the given DateTime will be returned. **kName** **kState** **kType**

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxAccessPoint identifier.
name	String	N	RW	Friendly name.
state	VxDeviceState	N	R	Current operational state of the access point.
status	VxAccessStatus[]	N	R	List of the current access point statuses.
statusSize	Integer	N	R	The size of `status`.
type	VxAccessPointType	N	RW	The particular type of the access point.

Method	Description
Delete()	Deletes this instance.
GetHostDevice(IVxDevice*& hostDevice)	Gets the IVxDevice that hosts this access point.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetRelations(VxCollection<IVxResourceRel**>& resourceRelCollection)	All possible resource relations for this IVxAccessPoint (both linked and non-linked). Each linked resource shall be considered to be associated to this IVxAccessPoint (e.g. relevant IVxDataSource, etc). *Note: Requires UPDATE access on the resource to link it.*
Lock(int seconds)	Locks the access point for the given amount of time (in seconds).
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

Method	Description
SetName(char name[128])	Sets the `name` property.
SetType(VxAccessPointType::Value type)	Sets the `type` property.
Unlock(int seconds)	Unlocks the access point for the given amount of time (in seconds).

5.3 IVxAlarmInput

Description	Collection Filters
Represents a physical alarm input.	**kAdvancedQuery** **kId** **kModifiedSince** Only IVxAlarmInput that have been modified since the given DateTime will be returned. **kName** **kState**

Field Name	Type	Req	RW	Description
description	String	N	RW	Friendly description.
id	String	Y	R	Unique IVxAlarmInput identifier.
name	String	N	RW	Friendly name.
state	VxAlarmState	N	R	Current state of the alarm input.
type	VxAlarmInputType	N	RW	The particular type of this alarm input.

Method	Description
Delete()	Deletes this instance.
GetHostDevice(IVxDevice*& hostDevice)	Gets the IVxDevice that hosts this alarm input.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetDescription(char* description)	Sets the `description` property.
SetName(char name[64])	Sets the `name` property.
SetType(VxAlarmInputType::Value type)	Sets the `type` property.

5.4 IVxAnalyticBehavior

Description	Collection Filters
Represents a specific analytic behavior for an analytic configuration.	**kAdvancedQuery** **kId** **kModifiedSince** Only IVxAnalyticBehavior that have been modified since the given DateTime will be returned. **kName**

Field Name	Type	Req	RW	Description
behaviorType	VxAnalyticBehaviorType	Y	R	The type of analytic behavior being performed.
id	String	Y	R	The unique identifier of the analytic behavior.
isEnabled	Boolean	N	RW	Indicates whether or not this analytic behavior is enabled.
name	String	N	RW	The friendly name of the analytic behavior.

Field Name	Type	Req	RW	Description
objectInZone	VxObjectInZone	N	RW	The object in zone data used to configure analytics of `behaviorType` is set to `kObjectInZone`.
objectLineCounter	VxObjectLineCounter	N	RW	The object line counter data used when `behaviorType` is set to `kObjectLineCounter`.
objectType	VxAnalyticObjectType	Y	RW	The type of object this analytic pertains to.
severity	Integer	N	RW	The severity value for events generated from this analytic behavior, from 1 (highest) to 10 (lowest). If set, overrides the corresponding situation severity.

Method	Description
DeleteAnalyticBehavior()	Delete this analytic behavior.
Disable()	Disable this analytic behavior.
Enable()	Enable this analytic behavior.
Refresh()	Refreshes this objects member values by retrieving its current information from the VideoXpert system.
SetName(char name[64])	Sets the `name` property.
SetObjectInZone(VxObjectInZone& objectInZone)	Sets the `objectInZone` property.
SetObjectLineCounter(VxObjectLineCounter& objectCounter)	Sets the `objectLineCounter` property.
SetObjectType(VxAnalyticObjectType::Value objectType)	Sets the `objectType` property.
SetSeverity(int severity)	Sets the severity value for events generated from this analytic behavior, from 1 (highest) to 10 (lowest). If set, overrides the corresponding situation severity.

5.5 IVxAnalyticConfig

Description	Collection Filters
Represents a specific analytic configuration for a data source.	None.

Field Name	Type	Req	RW	Description
analyticBehaviors	IVxAnalyticBehavior[]	Y	R	The associated analytic behaviors for this config.
analyticBehaviorSize	Integer	Y	R	The size of `analyticBehaviors`.
id	String	Y	R	The unique identifier of the IVxAnalyticConfig.
minConfidence	Float	N	RW	Defines the minimum confidence filtering value for detected objects in a video scene. Minimum confidence is defined as percentage of confidence represented as a decimal. For example, `0.0 = 0%`, and `1.0 = 100%`. Objects that have a detected confidence value less than the minimum will not be processed by the associated list of IVxAnalyticBehavior.
ptzPresetName	String	N	RW	Specifies the name of the IVxPreset that this configuration relates to. PTZ cameras supporting analytics can only be configured on PTZ presets.

Field Name	Type	Req	RW	Description
size	VxResolution	Y	RW	Specifies the resolution of the grid used for all IVxAnalyticBehavior data.

Method	Description
AddAnalyticBehavior(VxNewAnalyticBehavior& newAnalyticBehavior)	Creates a new analytic behavior.
Delete()	Deletes this instance.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetMinConfidence(float minConfidence)	Sets the `minConfidence` property.
SetPtzPresetName(char ptzPresetName[64])	Sets the `ptzPresetName` property.
SetSize(VxResolution& size)	Sets the `size` property.

5.6 IVxAnalyticSession

Description	Collection Filters
Represents an analytic session for a single video data source on an analytic server.	**kAdvancedQuery** **kDataSourceId** Only analytic sessions for this IVxDataSource. **kDeviceId** Only analytic sessions for this IVxDevice. **kId** **kModifiedSince** Only IVxAnalyticSession that have been modified since the given DateTime will be returned.

Field Name	Type	Req	RW	Description
dataEncodingId	String	N	RW	The unique identifier specifying which data encoding to use as a source.
id	String	Y	R	The unique identifier of the IVxAnalyticSession.
source	String	N	RW	The RTSP URI to use as the source for the analytic session.

Method	Description
Delete()	Deletes this instance.
DeleteAnalyticSession()	Delete this IVxAnalyticSession from the VideoXpert system.
GetAnalyticConfig(IVxAnalyticConfig*& analyticConfig)	Gets the IVxAnalyticConfig for this session.
GetDataSource(IVxDataSource*& dataSource)	Gets the associated video IVxDataSource.
GetHostDevice(IVxDevice*& hostDevice)	Gets the the associated host IVxDevice.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetDataEncodingId(char dataEncodingId[64])	Sets the unique identifier specifying which data encoding to use as a source. This value can be used instead of `source` to determine which stream to use.
SetSource(char source[1024])	Sets the RTSP URI to use as the source for the analytic session.

5.7 IVxBookmark

Description	Collection Filters
Represents a point in time of interest with reference to a particular IVxDataSource.	**kAdvancedQuery** **kDataSourceId** Only bookmarks for this IVxDataSource. **kDataSourceType** Only bookmarks for IVxDataSource of this type. **kDescription** **kGroupId** **kId** **kLocked** True to return only IVxBookmark that are locked; false to return only IVxBookmark that are unlocked. **kModifiedSince** Only IVxBookmark that have been modified since the given DateTime will be returned. **kName** **kSearchEndTime** Inclusive bounds to the oldest IVxBookmark to retrieve (DateTime). **kSearchStartTime** Inclusive bounds to the newest IVxBookmark to retrieve (DateTime). **kTime**

Field Name	Type	Req	RW	Description
dataSourceId	String	Y	R	ID of the associated IVxDataSource.
description	String	N	RW	Friendly description.
groupId	String	Y	R	IVxBookmark group identifier (supplied by clients). Typically used to identify related bookmarks (such as those bookmarking the same time on audio and video).
id	String	Y	R	Unique IVxBookmark identifier.
name	String	N	RW	Friendly name.
time	DateTime	Y	R	Indicates the time at which the point of interest occurred.

Method	Description
Delete()	Deletes this instance.
DeleteBookmark()	Delete this IVxBookmark from the VideoXpert system.
GetDataSource(IVxDataSource*& dataSource)	Gets the IVxDataSource associated with this IVxBookmark.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLock(IVxBookmarkLock*& bookmarkLock)	Gets IVxBookmarkLock configuration for this IVxBookmark.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetDescription(char description[255])	Sets the `description` property.
SetName(char name[64])	Sets the `name` property.

5.8 IVxBookmarkLock

Description	Collection Filters
Represents the lock configuration of a IVxBookmark. An enabled lock will prevent the media it locks from being garbage collected on recorders.	None.

Field Name	Type	Req	RW	Description
endTime	DateTime	Y	RW	Time at which the IVxBookmarkLock ends. MUST be after `startTime`. Defaults to IVxBookmark `time` + 30 seconds.
isEnabled	Boolean	Y	RW	True if locked; false otherwise.
startTime	DateTime	Y	RW	Time at which the IVxBookmarkLock begins. MUST be after `endTime`. Defaults to IVxBookmark `time` - 30 seconds.

Method	Description
Delete()	Deletes this instance.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetEndTime(char endTime[64])	Sets the `endTime` property.
SetIsEnabled(bool isEnabled)	Sets the `isEnabled` property.
SetStartTime(char startTime[64])	Sets the `startTime` property.

5.9 IVxClip

Description	Collection Filters
Represents a contiguous duration of stored media originating from a specific encoding for a single IVxDataSource. The clip may contain video, audio, analytic, or any other type of media that can be stored from a IVxDataSource (though a clip will contain only one of these types of media).	**kDataSourceId** **kEndTime** Retrieve all clips that end at this time (DateTime). Note that if you're searching for a range, you will typically use the `searchEndTime` filter. **kEvent** **kFramerate** **kSearchEndTime** Acts as an inclusive bounds to the oldest clip content to retrieve (DateTime). **kSearchStartTime** Acts as an inclusive bounds to the newest clip content to retrieve (DateTime). **kStartTime** Retrieve all clips that begin at this time (DateTime). Note that if you're searching for a range, you will typically use the `searchStartTime` filter.

Field Name	Type	Req	RW	Description
dataInterfaces	IVxDataInterface[]	N	R	All interfaces available for retrieval of the stored media this IVxClip represents. All interfaces for this IVxClip will have the same `dataEncodingId` (multiple interfaces are available when multiple transport protocols are supported).
dataInterfaceSize	Integer	N	R	The size of `dataInterfaces`.
dataSourceId	String	Y	R	ID of the IVxDataSource from which this clips content originated.
dataSourceName	String	N	R	Friendly name of the IVxDataSource from which this clip's content originated.
dataStorageId	String	N	R	ID of the IVxDataStorage on which the media for this IVxClip is stored.
endTime	DateTime	Y	R	Time at which the clip ends.
framerate	VxRecordingFramerate	N	R	Framerate level of this clip. Defaults to `kNormal`.

Field Name	Type	Req	RW	Description
recordingType	VxRecordingType	Y	R	Reason for the existence of this clip. This is the general type of trigger that caused the recording.
sourceDataStorageId	String	N	R	ID of the IVxDataStorage on which the media for this IVxClip was originally stored (SHOULD be empty if not different than `dataStorageId`). If different than `dataStorageId`, indicates that this IVxClip was copied to the IVxDataStorage (`dataStorageId`) from another IVxDataStorage (`sourceDataStorageId`).
startTime	DateTime	Y	R	Time at which the clip begins.
type	String	Y	R	The type of media contained in the clip.

Method	Description
Delete()	Deletes this instance.
GetSnapshotEndpoint(VxSnapshotFilter* filter, int filterSize, char* endpoint, int& size)	Retrieve up to 50 individual images from this IVxClip; only available for video clips. If more than 50 images are requested, only the first 50 will be returned. Available filters: `kWidth`: Scale to given width in pixels, maintaining ratio. `kOffset`: Return multiple images offset by this amount of time, in seconds, between images. `kStartTime`: DateTime at which the initial image should start. Defaults to IVxClip `startTime`. `kEndTime`: DateTime at which no further images should be returned. Defaults to IVxClip `endTime`.

5.10 IVxConfiguration::Auth

Description	Collection Filters
Represents authorization configuration.	None.

Field Name	Type	Req	RW	Description
isPasswordExpirationEnabled	Boolean	N	RW	Indicates whether or not password expiration is enabled for all users.
passwordExpiration	Integer	N	RW	The amount of time, in days, at which a user password will expire (from when it was last set/changed). Ignored if `isPasswordExpirationEnabled` is `false`.

Method	Description
Delete()	Deletes this instance.
DisablePasswordExpiration()	Disables password expiration for all users.
EnablePasswordExpiration()	Enables password expiration for all users.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

Method	Description
SetPasswordExpiration(int passwordExpiration)	Sets the `passwordExpiration` property.

5.11 IVxConfiguration::Bookmark

Description	Collection Filters
Represents bookmark configuration.	None.

Field Name	Type	Req	RW	Description
autoUnlock	Integer	N	RW	Automatically unlock any locked bookmark that has a IVxBookmarkLock `endTime` older than this number of days. A value of `0` will disable automatic unlocking.
retentionLimit		N	RW	The retention limit, in days, for bookmarks; a value of `0` will disable the limit. Any bookmark that exceeds this limit will be deleted. Ignored if the `retentionPolicy` is not set to `kDeleteAfterLimit`.
retentionPolicy	VxBookmarkRetentionPolicy	N	RW	Specifies if and when an old bookmark is automatically removed from the system.

Method	Description
Delete()	Deletes this instance.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetAutoUnlock(int autoUnlock)	Sets the `autoUnlock` property.
SetRetentionLimit(int retentionLimit)	Sets the `retentionLimit` property.
SetRetentionPolicy(VxBookmarkRetentionPolicy::Value retentionPolicy)	Sets the `retentionPolicy` property.

5.12 IVxConfiguration::Cluster

Description	Collection Filters
Represents Vx cluster configuration attributes. This representation shall be identical within a cluster, regardless of the node it was retrieved from.	None.

Field Name	Type	Req	RW	Description
aggregatedEventLimit	Integer	N	R	The retention limit, in days, on logged events from aggregated sources. Any aggregated events that exceed this limit shall be deleted.
coreVirtualIp	Host	N	R	Virtual IP or hostname to use for VxCore devices.
discoveryVirtualIp	Host	N	R	The virtual IP or hostname used for device discovery if the IVxConfiguration::Cluster is not governed by a VideoXpert Accessory Server. The load balancer uses this address for discovery announcements and balances responses across the IVxConfiguration::Cluster.

Field Name	Type	Req	RW	Description
exportPath	String	N	R	The UNC network path to save export media to.
exportPathUsername	String	N	R	The username used to access the network resource at the location specified by `exportPath`.
isExportPathEnabled	Boolean	N	RW	Indicates whether or not the alternative export storage location is enabled.
isStreamProxyEnabled	Boolean	N	RW	Indicates whether camera streams will be proxied through the VxStorage that they are assigned to or will always be pulled directly from the Media Gateway.
localEventLimit	Integer	N	R	The retention limit, in days, on logged events. Any events that exceed this limit shall be deleted.
mediaGatewayTranscast	String	N	R	DEPRECATED *This has been replaced by `transcastMethod` and will be removed in the future.* The default communication method between IVxDataSources to Media Gateways and between Media Gateways to clients. Defaults to `multicast-multicast`.
mediaGatewayVirtualIp	Host	N	R	Virtual IP or hostname to use for VxMG devices.
nodeLicenseLimit	Integer	N	R	The maximum number of allowed nodes under the current license. A license is required to expand the number of nodes that can be added to this IVxConfiguration::Cluster beyond this number. Set to -1 if there is no limit.
status	VxConfigStatus	Y	R	Current configuration status of the entire IVxConfiguration::Cluster.
statusCode	Integer	N	R	Current configuration status code. Servers SHOULD supply this if the `status` is `failed`. This code is opaque to clients though they MAY display it for troubleshooting purposes.
statusDescription	String	N	R	Current configuration status description. Servers SHOULD supply this if the `status` is `failed`. This description is opaque to clients though they MAY display it for troubleshooting purposes. Note that this is *not* localized.
transcastMethod	VxMgTranscast	N	RW	The default communication method between IVxDataSources to Media Gateways and between Media Gateways to clients.

Method	Description
Delete()	Deletes this instance.
DisableExportPath()	Disables the alternative export storage location.
DisableStreamProxy()	Disables the stream proxy, so camera streams are always pulled directly from the Media Gateway.
EnableExportPath()	Enables the alternative export storage location.
EnableStreamProxy()	Enables the stream proxy, which will proxy camera streams through the VxStorage that they are assigned to (if applicable).
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetNodeConfigurations(VxCollection <IVxConfiguration::Node**>& nodeCollection)	Gets the IVxConfiguration::Node collection that the cluster is composed of.
GetTimeConfig(IVxConfiguration::Time*& timeConfig)	Gets the IVxConfiguration::Time for the cluster.

Method	Description
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetAggregatedEventLimit(int aggregatedEventLimit)	Sets the retention limit, in days, on logged events from aggregated sources. Any aggregated events that exceed this limit shall be deleted.
SetCoreVirtualIp(char coreVirtualIp[256])	Sets the virtual IP or hostname to use for VxCore devices.
SetDiscoveryVirtualIp(char discoveryVirtualIp[256])	Sets the virtual IP or hostname to use for device discovery (if the IVxConfiguration::Cluster is not governed by a VideoXpert Accessory Server). The load balancer uses this address for discovery announcements and balances responses across the IVxConfiguration::Cluster.
SetExportPath(char exportPath[512])	Sets the UNC network path to save export media to. If set (and `isExportPathEnabled` is `true`), this location will be used instead of local IVxConfiguration::Cluster storage. If not set, local IVxConfiguration::Cluster storage will be used for storage of export media.
SetExportPathPassword(char exportPathPassword[64])	Sets the export path password. If set (and `isExportPathEnabled` is `true`), this password will be used to access the network resource specified in the `exportPath`.
SetExportPathUsername(char exportPathUsername[64])	Sets the export path username. If set (and `isExportPathEnabled` is `true`), this username will be used to access the network resource specified in the`exportPath`.
SetLocalEventLimit(int localEventLimit)	Sets the retention limit, in days, on logged events. Any events that exceed this limit shall be deleted.
SetMediaGatewayVirtualIp(char mediaGatewayVirtualIp[256])	Sets the virtual IP or hostname to use for Media Gateway devices.
SetTranscastMethod(VxMgTranscast::Value transcastMethod)	Sets the default communication method between data sources to Media Gateways and between Media Gateways to clients.

5.13 IVxConfiguration::Ldap

Description	Collection Filters
Represents the LDAP configuration available.	None.

Field Name	Type	Req	RW	Description
baseDn	String	Y	RW	The location in the LDAP database that will be used to search for and authenticate user entries. Example: `ou=Service Accounts, dc=Pelco, dc=org`
isEnabled	Boolean	Y	RW	Indicates whether or not LDAP username/password authentication is enabled.
isServerTlsEnabled	Boolean	Y	RW	Indicates whether or not SSL/TLS security is enabled for the LDAP server.
isSsoEnabled	Boolean	N	RW	Indicates whether or not single sign-on is enabled.

Field Name	Type	Req	RW	Description
isTwoStageBindingEnabled	Boolean	N	RW	Indicates whether or not two stage binding authentication is enabled.
objectClasses	String	N	RW	The LDAP object classes to search for users. Example: `user, inetOrgPerson`
searchAttributes	String	Y	RW	The attributes against which to match the user's identity/name. Example: `cn, mail`
searchDnAccount	String	N	RW	The superuser DN account in the LDAP directory to use for performing searches there. Example: `CN=users, DC=mycorp`. If `isSsoEnabled` is `true`, this account should have write permissions on the `vxRootDn`. Should be supplied if `isTwoStageBindingEnabled` or `useLdapUsersAndRoles` are `true`; ignored otherwise.
serverName	Host	Y	RW	The hostname or IP address of the LDAP server.
serverPort	Integer	Y	RW	Port number of the LDAP server.
ssoDomain	String	N	RW	The Active Directory/Kerberos domain used for single sign-on. Ignored if `isSsoEnabled` is `false`.
useLdapUsersAndRoles	Boolean	N	RW	Indicates whether or not LDAP will be used to manage users and roles. `false` to only use the LDAP server for authentication; the VideoXpert system will manage users and roles itself. `true` to manage user accounts and role assignments in the LDAP system. Note that permissions on roles will still be managed by the VideoXpert system.
vxRootDn	String	N	RW	The LDAP container under which VideoXpert system information for all sites is stored; this should be the same for all sites. Should be supplied if `useLdapUsersAndRoles` is `true`; ignored otherwise.
vxSystemDn	String	N	RW	The site specific LDAP container; unique per VideoXpert site. This DN should contain the `vxRootDn` (which is a suffix of this full DN). Should be supplied if `useLdapUsersAndRoles` is `true`; ignored otherwise.

Method	Description
Delete()	Deletes this instance.
DisableLdap()	Disables LDAP username/password authentication.
DisableLdapUsersAndRoles()	Disables management of users and roles through LDAP.
DisableServerTls()	Disables the SSL/TLS security for the LDAP server.
DisableSso()	Disables single sign-on.
DisableTwoStageBinding()	Disables two stage binding authentication.
EnableLdap()	Enables LDAP username/password authentication.
EnableLdapUsersAndRoles()	Enables management of users and roles through LDAP.
EnableServerTls()	Enables the SSL/TLS security for the LDAP server.
EnableSso()	Enables single sign-on.
EnableTwoStageBinding()	Enables two stage binding authentication.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetBaseDn(char baseDn[1024])	Sets the `baseDn` property.

Method	Description
SetObjectClasses(char objectClasses[1024])	Sets the `objectClasses` property.
SetSearchAttributes(char searchAttributes[1024])	Sets the `searchAttributes` property.
SetSearchDnAccount(char searchDnAccount[1024])	Sets the `searchDnAccount` property.
SetSearchDnPassword(char searchDnPassword[64])	Sets the password for the search DN account.
SetServerName(char serverName[256])	Sets the `serverName` property.
SetServerPort(int serverPort)	Sets the `serverPort` property.
SetSsoDomain(char ssoDomain[1024])	Sets the `ssoDomain` property.
SetVxRootDn(char vxRootDn[1024])	Sets the `vxRootDn` property.
SetVxSystemDn(char vxSystemDn[1024])	Sets the `vxSystemDn` property.
ValidateLdapCredentials(bool& isValid, VxLdapValidationCredentials& ldapCredentials)	Validate the given credentials on the specified LDAP server.

5.14 IVxConfiguration::Motion

Description	Collection Filters
Represents a motion detection configuration.	None.

Field Name	Type	Req	RW	Description
mode	VxMotionMode	N	RW	Set the motion detection mode. *Note: The `kRecorder` VxMotionMode is required for pixel search functionality. Also note that the recorder mode taxes server resources and setting this option MAY return a `kInsufficientResources` if the server is at capacity.*
sensitivity	Integer	N	RW	The amount of change that needs to occur in order to qualify as motion. Higher values increase sensitivity (less change required to trigger motion).
threshold	Integer	N	RW	The amount of area that needs to occur in order to qualify as motion. Higher values increase sensitivity (less area required to trigger motion).

Method	Description
Delete()	Deletes this instance.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetMode(VxMotionMode::Value mode)	Sets the `mode` property.
SetSensitivity(int sensitivity)	Sets the `sensitivity` property.
SetThreshold(int threshold)	Sets the `threshold` property.

5.15 IVxConfiguration::Node

Description	Collection Filters
Represents an individual host machine (a "node") within a VideoXpert cluster.	None.

Field Name	Type	Req	RW	Description
arbiterAlias	String	N	R	The host alias for the arbiter node.
arbiterPort	Integer	N	R	The port number that the arbiter node uses.
hostAddress	Host	Y	R	Node host address.
isArbiter	Boolean	Y	R	Indicates where or not this node is an arbiter.
status	VxConfigStatus	Y	R	Current configuration status of this node.
type	VxNodeType	N	R	The particular type of node this is.

Method	Description
Delete()	Deletes this instance.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.16 IVxConfiguration::Server

Description	Collection Filters
Represents general server configuration.	None.

Field Name	Type	Req	RW	Description
httpPort	Integer	N	RW	The HTTP port number.
httpsPort	Integer	N	RW	The HTTPS port number.
isReverseDnsLookupEnabled	Boolean	N	RW	Indicates whether or not hostname lookup from IP is enabled.
rtspPort	Integer	N	RW	The RTSP port number.

Method	Description
Delete()	Deletes this instance.
DisableReverseDnsLookup()	Disables hostname lookup from IP.
EnableReverseDnsLookup()	Enables hostname lookup from IP.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
Restart()	Restarts the server; generally required after any port changes to have them take effect.
SetHttpPort(int httpPort)	Sets the `httpPort` property.
SetHttpsPort(int httpsPort)	Sets the `httpsPort` property.
SetRtspPort(int rtspPort)	Sets the `rtspPort` property.

5.17 IVxConfiguration::Smtp

Description	Collection Filters
Represents the SMTP configuration.	None.

Field Name	Type	Req	RW	Description
fromAddress	String	N	RW	The "Mail From" email address.
host	Host	N	RW	The SMTP host.
isValidateCertEnabled	Boolean	N	RW	Indicates whether or not validation of the servers certificate is enforced.
smtpPort	Integer	N	RW	The SMTP port.
timeout	Integer	N	RW	The timeout in milliseconds.
username	String	N	RW	The username for SMTP access.
useSmtps	Boolean	N	RW	Indicates whether SMTPS should be used instead of SMTP.

Method	Description
Delete()	Deletes this instance.
DisableCertValidation()	Disables enforcing validation of the servers certificate.
DisableSmtps()	Disables the use of SMTPS.
EnableCertValidation()	Enables enforcing validation of the servers certificate.
EnableSmtps()	Enables the use of SMTPS instead of SMTP.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetFromAddress(char fromAddress[256])	Sets the `fromAddress` property.
SetHost(char host[256])	Sets the `host` property.
SetPassword(char password[64])	Sets the password for SMTP access.
SetSmtpPort(int smtpPort)	Sets the `smtpPort` property.
SetTimeout(int timeout)	Sets the `timeout` property.
SetUsername(char username[64])	Sets the `username` property.
ValidateSmtpInfo(bool& isValid, VxSmtpInfo& smtpInfo)	Validate the given SMTP information by attempting to use it to send a test email.

5.18 IVxConfiguration::Snmp

Description	Collection Filters
Represents the SNMP agent configuration.	None.

Field Name	Type	Req	RW	Description
snmpPort	Integer	N	RW	The port on which to provide access to SNMP.
v2cReadCommunityString	String	N	RW	The SNMP v2c community string.
v3AuthProtocol	VxAuthProtocol	N	RW	The SNMP v3 authentication protocol.
v3PrivacyProtocol	VxPrivacyProtocol	N	RW	The SNMP v3 privacy protocol.

Field Name	Type	Req	RW	Description
v3ReadUsername	String	N	RW	The SNMP v3 read username.
version	VxSnmpVersion	N	RW	Indicates which version of SNMP is enabled.

Method	Description
Delete()	Deletes this instance.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetSnmpPort(int snmpPort)	Sets the `snmpPort` property.
SetV2cReadCommunityString(char v2cReadCommunityString[256])	Sets the `v2cReadCommunityString` property.
SetV3ReadUsername(char v3ReadUsername[256])	Sets the `v3ReadUsername` property.
SetV3ReadAuthPassword(char password[64])	Sets the SNMP v3 read auth password.
SetV3ReadPrivacyPassword(char password[64])	Sets the SNMP v3 read privacy password.
SetV3AuthProtocol(VxAuthProtocol::Value v3AuthProtocol)	Sets the `v3AuthProtocol` property.
SetV3PrivacyProtocol(VxPrivacyProtocol::Value v3PrivacyProtocol)	Sets the `v3PrivacyProtocol` property.
SetVersion(VxSnmpVersion::Value version)	Sets the `version` property.

5.19 IVxConfiguration::Storage

Description	Collection Filters
Represents VxStorage-specific configuration.	None.

Field Name	Type	Req	RW	Description
failoverGroup	String[]	N	R	List of IVxDevice IDs belonging to this IVxConfiguration::Storage failover group (including itself).
failoverGroupSize	Integer	N	R	The size of `failoverGroup`.
failoverMaxRecordingBitrate	Integer	N	RW	The maximum recording bitrate for this IVxConfiguration::Storage when enabled as a failover unit (in Mbps).
gapFillerAttempts	Integer	N	RW	The maximum number of attempts to fill a IVxGap before giving up.
gapFillerDownloads	Integer	N	RW	The maximum number of concurrent downloads.
gapFillerInterval	Integer	N	RW	The interval, in seconds, to reattempt to automatically fill a IVxGap.
isFailoverEnabled	Boolean	N	RW	Indicated whether or not this IVxConfiguration::Storage is designated as a failover unit.
isGapFillerEnabled	Boolean	N	RW	Indicated whether or not automatic IVxGap filling is enabled.
manualRecordingTimeout	Integer	N	RW	The amount of time, in seconds, to allow a IVxManualRecording to be active.
maxOutboundBitrate	Integer	N	RW	The maximum outbound bitrate for this IVxConfiguration::Storage (in Mbps).

Field Name	Type	Req	RW	Description
monitoredDeviceIds	String[]	N	RW	List of IVxDevice IDs being monitored for failover.
monitoredDeviceIdsSize	Integer	N	R	The size of `monitoredDeviceIds`.
retentionLimit	Integer	Y	RW	Set a retention limit, in hours, on recorded data; a value of `0` will disable the limit. Any recorded data that exceeds this limit will be deleted. Defaults to `0`.
transmissionPreference	VxTransmissionType	Y	RW	The network communication transmission type preference.
transmissionType	String	Y	RW	DEPRECATED *This has been replaced by `transmissionPreference` and will be removed in the future.* Network communication transmission type preference.
videoStreamRecordingSource	VxStreamSource	N	RW	The stream to record from for all assigned IVxDataSources (when scheduled to record).

Method	Description
CreateVolume(VxNewVolume& newVolume)	Creates a new IVxVolume on the IVxConfiguration::Storage.
CreateVolumeGroup(VxNewVolumeGroup& newVolumeGroup)	Creates a new IVxVolumeGroup on the IVxConfiguration::Storage.
Delete()	Deletes this instance.
DisableFailover()	Disables this IVxConfiguration::Storage as a failover unit.
DisableGapFiller()	Disables automatic IVxGap filling.
EnableFailover()	Enables this IVxConfiguration::Storage as a failover unit (it will failover the data storages on any device it's monitoring). *Note: Failover units will not provide device assignment nor schedule management functionality; they are dedicated to failover monitoring functions.*
EnableGapFiller()	Enables automatic IVxGap filling.
GetFileRecovery(IVxFileRecovery*& fileRecovery)	Gets the file recovery control and monitor resource for this IVxConfiguration::Storage.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetVolumes(VxCollection <IVxVolume**>& volumeCollection)	Gets the IVxVolumes for this IVxConfiguration::Storage.
GetVolumeGroups(VxCollection <IVxVolumeGroup**>& volumeGroupCollection)	Gets the IVxVolumeGroups for this IVxConfiguration::Storage.
RebuildDatabase()	Rebuilds the database. Has no effect if the database is already rebuilding.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetFailoverMaxRecordingBitrate(int failoverMaxRecordingBitrate)	Sets the maximum recording bitrate for this IVxConfiguration::Storage when enabled as a failover unit (in Mbps).
SetGapFillerAttempts(int gapFillerAttempts)	Sets the maximum number of attempts to fill a IVxGap before giving up.

Method	Description
SetGapFillerDownloads(int gapFillerDownloads)	Sets the maximum number of concurrent downloads.
SetGapFillerInterval(int gapFillerInterval)	Sets the interval, in seconds, to reattempt to automatically fill a IVxGap.
SetManualRecordingTimeout(int manualRecordingTimeout)	Sets the amount of time, in seconds, to allow a IVxManualRecording to be active. The IVxManualRecording will be automatically deleted after this amount of time elapses if it has not been refreshed. No timeout is applied if the value is less than `1`.
SetMaxOutboundBitrate(int maxOutboundBitrate)	Sets the maximum outbound bitrate for this IVxConfiguration::Storage (in Mbps).
SetMonitoredDeviceIds(char** deviceIds, int deviceIdsSize)	Sets the list of IVxDevice IDs being monitored for failover. If `isFailoverEnabled` is `true` and a failure is detected on one of these, this IVxConfiguration::Storage will failover all IVxDevice assigned to the IVxDataStorage on that IVxDevice. A value of `0` will will clear any existing device IDs.
SetRetentionLimit(int retentionLimit)	Sets the retention limit, in hours, on recorded data; a value of `0` will disable the limit. Any recorded data that exceeds this limit will be deleted.
SetTransmissionPreference(VxTransmissionType::Value transmissionPreference)	Sets the network communication transmission type preference. Note that if the preferred communication fails, this IVxConfiguration::Storage may attempt to utilize unicast.
SetVideoStreamRecordingSource(VxStreamSource::Value videoStreamRecordingSource)	Sets the stream to record from all assigned IVxDataSources (when scheduled to record).

5.20 IVxConfiguration::Time

Description	Collection Filters
Represents the time configuration.	None.

Field Name	Type	Req	RW	Description
isTimeServerEnabled	Boolean	Y	R	Indicates whether or not the external time server is enabled. True will cause the device to synchronize its time with the server specified by `timeServerAddress`, false will disable external time server synchronization.
timeServerAddress	Host	Y	R	Host address of the external time server.

Method	Description
Delete()	Deletes this instance.
GetTime(char currentTime[64])	Gets the current internal system time.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.21 IVxConfiguration::Twilio

Description	Collection Filters
Represents Twilio (twilio.com) SMS account configuration.	None.

Field Name	Type	Req	RW	Description
accountSid	String	N	RW	The Twilio account SID.
fromNumber	String	N	RW	A Twilio phone number capable of sending SMS (e.g., `+15595551234`).
isEnabled	Boolean	N	RW	Indicates whether or not SMS messaging via Twilio is enabled.

Method	Description
Delete()	Deletes this instance.
Disable()	Disables SMS messaging via Twilio.
Enable()	Enables SMS messaging via Twilio.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetAccountSid(char accountSid[256])	Sets the `accountSid` property.
SetAuthToken(char authToken[512])	Sets the Twilio account authorization token.
SetFromNumber(char fromNumber[64])	Sets the `fromNumber` property.
ValidateTwilioInfo(bool& isValid, const char* accountSid, const char* authToken, const char* fromNumber, const char* toNumber)	Validate the given Twilio info by attempting to use it to send a test SMS.

5.22 IVxDataInterface

Description	Collection Filters
Represents a data interface for a IVxDataSource. Clients can retrieve data from the IVxDataSource using the protocol specified by this interface.	None.

Field Name	Type	Req	RW	Description
bitrate	Integer	N	R	Average bitrate of the stream, if available (in bps).
dataEncodingId	String	N	R	ID for a specific data encoding (based on encoding quality, framerate, and resolution). Multiple identical copies of the data may exist across the system; these will have the same `dataEncodingId`. Interfaces that map directly to an ordinal stream of a IVxDataSource SHALL use special ordinal number strings for this ID: “primary”, “secondary”, “tertiary”, etc.

Field Name	Type	Req	RW	Description
dataEndpoint	String	Y	R	Protocol-specific stream control URI. Interaction with this link is based upon the protocol being used: **rtsp**: RTSP endpoint. Note that a IVxDataSession is implicitly created by the server whenever an RTSP session is started. The IVxDataSession is removed when the RTSP session is stopped (or explicitly deleted). The RTSP protocol is used for control. **mjpeg-pull**: MJPEG endpoint. Not that this should not be used directly, instead use the IVxDataSource `CreateMjpegDataSession` method.
format	VxStreamFormat	N	R	The media stream encoding format.
framerate	Float	N	R	Framerate of the data.
isTranscoded	Boolean	N	R	True if the interface provides a transcoded stream. If False, the stream is not transcoded. Note that transcoded streams utilize more server resources; non-transcoded streams should be preferred.
multicastTestIp	IP	N	R	IP used to test multicast transmission capabilities.
multicastTestPort	Integer	N	R	Port used to test multicast transmission capabilities.
overlayTypes	VxOverlayType[]	N	R	Available overlays. See the protocol-specific stream controls for how to enable these overlays.
overlayTypeSize	Integer	N	R	The size of `overlayTypes`.
protocol	VxStreamProtocol	Y	R	Interface protocol.
renderType	VxRenderType	N	R	Type of rendering required for the media data delivered by this interface. Clients can utilize this to create the correct rendering pipeline. Defaults to `kStandard`.
supportsMulticast	Boolean	Y	R	True if the interface provides multicast transmission. If False, the transmission is unicast.
xResolution	Integer	N	R	Horizontal resolution of the data.
yResolution	Integer	N	R	Vertical resolution of the data.

5.23 IVxDataObject

Description	Collection Filters
A IVxDataObject contains a custom serialized data object submitted by a client and stored on the server. The data is completely opaque to the server and other types of clients.	**kClientType** **kModifiedSince** Only IVxDataObject that have been modified since the given DateTime will be returned. **kOwned** True to return only owned (private) IVxDataObject; false to return only non-owned (public) IVxDataObject. **kOwner**

Field Name	Type	Req	RW	Description
clientType	String	Y	R	IVxDataObject client identifier.
id	String	Y	R	Unique IVxDataObject identifier.
owner	UPN	N	R	If present, indicates that this resource is owned by a IVxUser (it is private); this is their username. If not present, indicates that this resource is not owned (it is public). If this resource has an owner, only the owner and users with appropriate permissions will be able to read it.

Method	Description
Delete()	Deletes this instance.
DeleteDataObject()	Remove the IVxDataObject from the system.
GetData(char* data, int& size)	Serialized data object (e.g.: JSON, XML, CSV, etc). The server MUST NOT utilize this data in any way. The maximum allowable size of this field is 1 MB.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetOwner(IVxUser*& user)	Retrieve the IVxUser that owns this IVxDataObject, if any. This will be omitted if there is no owner.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetData(char* data)	Sets the serialized data value.

5.24 IVxDataSession

Description	Collection Filters
Represents a data session that is being transmitted by a IVxDataSource, via a IVxDataInterface, to a client. If this is an `kMjpegPull` session, the session MAY timeout if the client performs no request to the `jpegUri` within 30 seconds. If a `404 Not Found` is received when interacting with this IVxDataSession, it is safe to assume the session has timed out; a new IVxDataSession should be retrieved from the IVxDataSource.	None.

Field Name	Type	Req	RW	Description
id	String	Y	R	A unique identifier for this IVxDataSession.
jpegUri	JPG	N	R	The URI to the `kMjpegPull` frame of data. Refreshes the session. If there is no remaining data, `kEndOfStream` will be returned. If there is currently no data, but more will become available, `kEdgeOfStream` will be returned.
quality	Integer	N	RW	Image quality of the data; from `1` to `100`. Smaller numbers have better quality than larger numbers (e.g. `1` is the best quality). Defaults to `1`.
speed	Float	N	RW	Play speed of the data. Negative values indicate reverse speeds while positive values indicate forward speeds (`1` is normal speed). A value of `0` will pause the session. Defaults to `1`.
xResolution	Integer	N	RW	Horizontal resolution of the data. Defaults to native horizontal resolution of the data.
yResolution	Integer	N	RW	Vertical resolution of the data. Defaults to native vertical resolution of the data.

Method	Description
Delete()	Deletes this instance.
DeleteDataSession()	Terminate this session.
GetAuthToken(char* token, int& size)	Gets the authentication token used by this session.
GetDataInterface(IVxDataInterface*& dataInterface)	The IVxDataInterface used by this session.
GetDataSource(IVxDataSource*& dataSource)	The IVxDataSource from which this session is retrieving data.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetUser(IVxUser*& user)	The IVxUser, if any, that this session is being used by.
GoLive()	Set the temporal position of the session as close to live as possible.

Method	Description
Pause()	Pauses the session.
Play()	Starts or resumes the session.
RefreshSession()	Refresh the session.
Seek(long long unixTime, float speed)	Absolute temporal position of the session. Editing this field directs the session to seek to the frame nearest this time (starting at this time and scanning in the direction of the current speed). If no frame is available within a reasonable period, a `kInvalidValue` is returned and `time` will remain unchanged. A IVxDataSession defaults to live; this value shall then be the time of the frame as close to live as possible.
SetSpeed(float speed)	Sets the play speed of the data session. Negative values indicate reverse speeds while positive values indicate forward speeds (`1` is normal speed).
SetQuality(unsigned short quality)	Sets the quality property.
SetResolution(unsigned short xResolution, unsigned short yResolution)	Sets the resolution properties.
Update()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.25 IVxDataSource

Description	Collection Filters
Represents a data producer in the system (e.g. an audio, video, or metadata output stream) hosted by a system IVxDevice (e.g. a camera or microphone). Each IVxDataSource provides a list of IVxDataInterface, each of which provides a particular protocol used to transmit and control the data produced by this IVxDataSource.	**kAdvancedQuery** **kAllTags** Comma-separated list of public tag names. Only IVxDataSource tagged by *all* of these public tags will be returned. **kCapturing** **kCommissioned** True to return only commissioned IVxDevice-hosted IVxDataSource; false to return only non-commissioned IVxDevice-hosted IVxDataSource. **kEnabled** **kEncoding** **kHasFolderTags** True to return IVxDataSource tagged by 1..* folder tag(s); false to return IVxDataSource with no folder tags. **kId** **kIp** **kManualRecording** **kModifiedSince** Only IVxDataSource that have been modified since the given DateTime will be returned. **kName** **kNumber** **kRecording** **kState** **kType** **kUnassigned** True to return only IVxDataSource that are *not* assigned to a IVxDataStorage.

Field Name	Type	Req	RW	Description
capabilities	VxDataSourceCapability[]	N	R	List of capabilities supported by this IVxDataSource, if any.
capabilitiesSize	Integer	N	R	The size of `capabilities`.
dataInterfaces	IVxDataInterface[]	Y	R	All interfaces available for retrieval of data from this IVxDataSource, sorted by preference of use from highest to lowest.
dataInterfaceSize	Integer	N	R	The size of `dataInterfaces`.
encoding	String	N	R	Internet media type representing the format of the source encoding.
hasLive	Boolean	Y	R	True if the client has authorization to access live media from this IVxDataSource (based on client credentials).
hasRecorded	Boolean	Y	R	True if the client has authorization to access recorded media from this IVxDataSource (based on client credentials).
id	String	Y	R	A unique identifier for this IVxDataSource.
index	Integer	N	R	Zero-based index (per host IVxDevice) indicating the "position" of the IVxDataSource (per IVxDataSource type) within the host IVxDevice. The definition of "position" in this case is IVxDevice-dependant, though the definition SHOULD be consistent for all indexed services for the IVxDevice (e.g. IVxAlarmInputs on the same IVxDevice SHOULD match up by index).
ip	IP	Y	R	Primary IP address.
isCapturing	Boolean	N	R	True indicates that this IVxDataSource is currently being captured by (streaming to) a recorder. False indicates that this IVxDataSource is *not* currently being captured by a recorder.
isEnabled	Boolean	N	RW	Indicates whether this IVxDataSource is enabled (true) or disabled (false). A disabled IVxDataSource shall not expose any IVxDataInterface, will not be recorded, and will not consume any system license feature counts. Enabling a IVxDataSource may fail if there is not a sufficient license available (the IVxDataSource shall remain disabled). Disabling a IVxDataSource will terminate all active recordings and free any license feature counts in use.
isManuallyRecording	Boolean	N	R	True indicates that this IVxDataSource is currently being manually recorded (by one or more IVxDataStorage).
isRecording	Boolean	N	R	True indicates that this IVxDataSource is currently being recorded by a recorder. False indicates that this IVxDataSource is *not* currently being recorded by a recorder.
linkedPtzInfo	VxLinkedPtzInfo[]	N	R	Provides information on any IVxDataSources that are tracking this IVxDataSource. Only available if this IVxDataSource supports linked PTZ.

Field Name	Type	Req	RW	Description
linkedPtzInfoSize	Integer	N	R	The size of `linkedPtzInfo`.
name	String	N	RW	Friendly name.
number	Integer	N	RW	A number used to designate the IVxDataSource.
retentionLimit	Integer	N	RW	The maximum retention (in hours) that the system will keep recorded data for this IVxDataSource. Any recorded data that exceeds this limit will be deleted. If a retention limit is also set in IVxConfiguration::Storage, the lowest non-zero value will be used. A value of `0` means no retention limit will be used.
snapshotUri	String	N	R	Provides a simple means for clients to obtain a snapshot image from this IVxDataSource. This is only available when the IVxDataSource `type` is `video`. When retrieving the snapshot, a `width` query filter may be used to return the image using the given width in pixels, maintaining its ratio.
sourceEndpoint	String	N	R	The source URI of the IVxDataSource.
state	VxDeviceState	Y	R	Current operational state.
type	VxDataSourceType	Y	R	The particular type of this IVxDataSource.

Method	Description
AddAnalyticSession(VxNewAnalyticSession& newAnalyticSession)	Adds a new analytic session.
CanPixelSearch(bool& canPixelSearch)	Gets a value indicating whether pixel search is available for this IVxDataSource.
CanPtz(bool& canPtz)	Gets a value indicating whether PTZ is enabled.
CreateMjpegDataSession(IVxDataSession*& dataSession)	Create a new MJPEG IVxDataSession.
CreatePixelSearch(VxNewPixelSearch& newPixelSearch, IVxPixelSearch*& pixelSearch)	Request a new IVxPixelSearch. The new IVxPixelSearch will be returned in the response. Only available if pixel search data is available.
Delete()	Deletes this instance.
Disable()	Disable this IVxDataSource. A disabled IVxDataSource shall not expose any IVxDataInterface, will not be recorded, and will not consume any system license feature counts.
Enable()	Enable this IVxDataSource.
GetAllDataStorages(VxCollection<IVxDataStorage**>& dataStorageCollection)	Retrieve *all* IVxDataStorage that this IVxDataSource is associated with; includes edge and failover IVxDataStorage.
GetAnalyticConfigs(VxCollection<IVxAnalyticConfig**>& analyticConfigCollection)	Gets the collection of IVxAnalyticConfig for this IVxDataSource.
GetAnalyticSessions(VxCollection<IVxAnalyticSession**>& analyticSessionCollection)	Gets the collection of IVxAnalyticSession for this IVxDataSource.

Method	Description
GetAudioRelations(VxCollection<IVxResourceRel**>& resourceRelCollection)	All possible *audio* resource relations for this IVxDataSource (both linked and non-linked). Each linked resource shall be considered to be associated to this IVxDataSource while non-linked resources shall not be (they are available to be associated). *Note: Requires UPDATE access on the IVxDataSource in order to link.*
GetBookmarks(VxCollection<IVxBookmark**>& bookmarkCollection)	Collection of IVxBookmark associated with this IVxDataSource.
GetClips(VxCollection<IVxClip**>& clipCollection)	Query for clips associated with this IVxDataSource. Edge clips are not returned (see: `GetEdgeClips`).
GetDataStorages(VxCollection<IVxDataStorage**>& dataStorageCollection)	Retrieve all IVxDataStorage that this IVxDataSource is assigned to.
GetEdgeClips(VxCollection<IVxClip**>& clipCollection)	Query for *edge* clips associated with this IVxDataSource; may take a long time (minutes).
GetGaps(VxCollection<IVxGap**>& gapCollection)	Collection of IVxGap for this IVxDataSource.
GetHostDevice(IVxDevice*& hostDevice)	Host IVxDevice of this IVxDataSource.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetMember(IVxMember*& member)	Gets the IVxMember that this IVxDataSource resides in.
GetMetadataRelations(VxCollection<IVxResourceRel**>& resourceRelCollection)	All possible *metadata* resource relations for this IVxDataSource (both linked and non-linked). Each linked resource shall be considered to be associated to this IVxDataSource while non-linked resources shall not be (they are available to be associated). *Note: Requires UPDATE access on the IVxDataSource in order to link.*
GetMotionConfiguration(IVxConfiguration::Motion*& motionConfig)	Motion detection configuration for this IVxDataSource.
GetMultiviewInfo(VxCollection<IVxUserInfo**>& userInfoCollection)	Multiple viewer information for this IVxDataSource.
GetPtzController(IVxPtzController*& ptzController)	Control PTZ settings.
GetRtspEndpoint(char* endpoint, int& size)	Convenience method to retrieve the first available RTSP stream endpoint URI for this IVxDataSource.
GetTags(VxCollection<IVxTag**>& tagCollection)	Collection of IVxTag associated with this IVxDataSource.
GetVideoRelations(VxCollection<IVxResourceRel**>& resourceRelCollection)	All possible *video* resource relations for this IVxDataSource (both linked and non-linked). Each linked resource shall be considered to be associated to this IVxDataSource while non-linked resources shall not be (they are available to be associated). *Note: Requires UPDATE access on the IVxDataSource in order to link.*
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetName(char name[64])	Sets the `name` property.
SetNumber(int number)	Sets the `number` property.

Method	Description
SetRetentionLimit(int retentionLimit)	Sets the `retentionLimit` property.

5.26 IVxDataStorage

Description	Collection Filters
Represents a data storage provider in the system (e.g. an NSM5200 storage pool or a VxStorage) hosted by a system IVxDevice. The IVxDataStorage can be directed to store media produced by a IVxDevice by assigning the IVxDevice to it (see method: `AssignDevice`).	**kAdvancedQuery** **kCommissioned** True to return only commissioned IVxDevice-hosted IVxDataStorage; false to return only non-commissioned IVxDevice-hosted IVxDataStorage. **kId** **kModifiedSince** Only IVxDataStorage that have been modified since the given DateTime will be returned. **kName** **kType**

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxDataStorage identifier.
isFailover	Boolean	N	R	`True` if this IVxDataStorage is configured as a failover unit; `false` otherwise. Defaults to `false`.
name	String	N	RW	Friendly name.
type	VxDataStorageType	Y	R	The particular type of the IVxDataStorage.

Method	Description
AddClip(VxNewClip& newClip)	Save the data specified by `newClip` to this IVxDataStorage. Once saved, the data will be represented as a new IVxClip (not immediately available). Data that is already saved will not be retrieved again. If there is overlap with data that is already saved, only the new data will be saved.
AssignDevice(VxNewDeviceAssignment& newDeviceAssignment)	Assign existing IVxDevice resources to this IVxDataStorage to be managed and recorded by the IVxDataStorage based on its configuration. Any assignments already present have no effect (considered successful). If any of the assigned IVxDevice do not exist or are not applicable to the IVxDataStorage, a `kInvalidValue` is returned. If any of the assigned IVxDevice are deleted, their assignment here shall also be removed.
Delete()	Deletes this instance.
GetDeviceAssignments(VxCollection<IVxDeviceAssignment**>& deviceAssignmentCollection)	Current IVxDevice assignments (assignments to this IVxDataStorage).
GetDataSources(VxCollection<IVxDataSource**>& dataSourceCollection)	Collection of IVxDataSource that are assigned to this IVxDataStorage.
GetDeviceAssignments(VxCollection<IVxDeviceAssignment**>& deviceAssignmentCollection)	Current IVxDevice assignments (assignments to this IVxDataStorage).
GetHostDevice(IVxDevice*& hostDevice)	Host IVxDevice of this IVxDataStorage.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetRetentions(VxCollection<VxRetention**>& retentionCollection)	Gets the resource retention information for this data storage.

Method	Description
GetStorageConfiguration(IVxConfiguration::Storage*& storageConfig)	Gets the IVxConfiguration::Storage for this data storage.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetName(char name[64])	Sets the `name` property.
UnassignDevice(IVxDevice& device)	Unassign the given list of IVxDevice from this IVxDataStorage. For any assignments that do not exist, this has no effect (considered successful).

5.27 IVxDbBackup

Description	Collection Filters
Represents a database backup.	None.

Field Name	Type	Req	RW	Description
completed	DateTime	N	R	The date and time at which this backup completed.
id	String	Y	R	Unique IVxDbBackup identifier.
initiated	DateTime	N	R	The date and time at which this backup was initiated.
status	VxBackupStatus	Y	R	The current status of this backup.
statusReason	VxBackupStatusReason	N	R	The optional reason for the current status of this backup; typically used to express the reason for a failure.

Method	Description
Delete()	Deletes this instance.
DeleteBackup()	Deletes this IVxDbBackup. If the backup is in progress, it shall be halted first.
Halt()	Halts the current backup operation that is in progress, if any.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
Restore()	Restore this IVxDbBackup to the database.

5.28 IVxDbBackups

Description	Collection Filters
Represents a collection of database backups and provides database backup and control.	None.

Field Name	Type	Req	RW	Description
backupInterval	Integer	N	RW	The number of days between automatic database backups.
backupPath	String	N	RW	The optional UNC network path to use for alternate backup storage (instead of the server default location).
backupPathUsername	String	N	RW	The username used to access the backup storage location path.
backupTime	Time	N	RW	The time at which daily automatic database backups will begin.
maxBackupsToKeep	Integer	N	RW	The maximum number of backups to keep in the backup directory before overwriting the oldest.

Method	Description
Delete()	Deletes this instance.
GetDatabaseBackups(VxCollection<IVxDbBackup**>& dbBackupCollection))	Gets the current collection of IVxDbBackup.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
SetBackupInterval(int backupInterval))	Sets the number of days between automatic database backups. A value of 0 will disable automatic database backups.
SetBackupPath(char backupPath[512]))	Sets the UNC network path to use for alternate backup storage (instead of the server default location).
SetBackupPathPassword(char backupPathPassword[64]))	Sets the backup path password. If set, this password will be used to access the network resource specified in the `backupPath`.
SetBackupPathUsername(char backupPathUsername[64]))	Sets the backup path username. If set, this username will be used to access the network resource specified in the `backupPath`.
SetBackupTime(char backupTime[9]))	Sets the time at which daily automatic database backups will begin, from `00:00:00` to `23:59:59`.
SetMaxBackupsToKeep(int maxBackupsToKeep))	Sets the maximum number of backups to keep in the backup directory before overwriting the oldest.
TriggerBackup())	Initiate a new unscheduled database backup.
ValidateBackupPath(bool& isValid, const char* backupPath, const char* backupPathUsername, const char* backupPathPassword))	Gets a value indicating whether a network storage path is available and valid as a backup storage.

5.29 IVxDevice

Description	Collection Filters
Represents a particular physical device in the system (e.g.: an NSM5200).	**kAdvancedQuery** **kCommissioned** True to return only commissioned IVxDevice; false to return only non-commissioned IVxDevice. **kDriverType** **kHasStatus** Only IVxDevice that have this status; they may have other status as well. **kId** **kIp** **kModel** **kModifiedSince** Only IVxDevice that have been modified since the given DateTime will be returned. **kName** **kSerial** **kState** **kType** **kVendor** **kVersion**

Field Name	Type	Req	RW	Description
driverDeviceId	String	N	R	Driver IVxDevice identifier.
driverTypeId	String	N	RW	Type identifier of the IVxDriver to use for device communication. If not set, a IVxDriver will be selected automatically, if possible.

Field Name	Type	Req	RW	Description
endpoints	String[]	N	RW	List of source URIs that the IVxDevice will support.
endpointsSize	Integer	N	R	The size of `endpoints`.
hostname	String	N	RW	Hostname of the IVxDevice. If set, takes precedence over the ip. The combination of hostname, ip, port and type must be unique.
id	String	Y	R	Unique IVxDevice identifier.
ip	IP	Y	RW	Primary IP address. The IP address MUST be unique for this type of IVxDevice.
isCommissioned	Boolean	Y	R	True if this IVxDevice is commissioned; false if this IVxDevice is not commissioned. A non-commissioned IVxDevice MAY limit thefeatures it provides.
isLicenseRequired	Boolean	N	R	True if this IVxDevice requires a license in order to be commissioned; commissioning this IVxDevice will consume an available commission on the associated IVxLicenseFeature. False indicates that this IVxDevice may be commissioned without a license.
licensableFeatures	String[]	N	R	List of supported IVxLicenseFeature that may be licensed for use by this IVxDevice.
licensableFeaturesSize	Integer	N	R	The size of `licensableFeatures`.
licensedFeatures	String[]	N	R	List of supported IVxLicenseFeature that are currently licensed for use by this IVxDevice.
licensedFeaturesSize	Integer	N	R	The size of `licensedFeatures`.
model	String	N	R	Product model name.
name	String	N	RW	Friendly name.
port	Integer	N	RW	Host port.
serial	String	N	R	Serial number.
state	VxDeviceState	Y	R	Current operational state.
status	VxDeviceStatus[]	N	R	List of current device status.
statusSize	Integer	N	R	The size of `status`.
type	VxDeviceType	Y	R	The particular type of the IVxDevice.
username	String	N	RW	Account username used to communicate with the IVxDevice, if any.
vendor	String	N	R	IVxDevice vendor, if available.
version	String	N	RW	The current version of the IVxDevice.
virtualIp	IP	N	R	Virtual IP address used by the IVxDevice, if any. If this IVxDevice is part of a recording pool or cluster, the `virtualIp` may be used as a unique pool/cluster identifer.

Method	Description
AddAnalyticSession(VxNewAnalyticSession& newAnalyticSession)	Adds a new analytic session.
CanCreateLogs(bool& canCreateLogs)	Gets a value indicating whether his IVxDevice is capable of generating logs.
CreateLog()	Create a new IVxLog, the contents of which shall be determined by the server by default.
Delete()	Deletes this instance.

Method	Description
DeleteDevice()	Remove the IVxDevice and its hosted IVxDataSource, IVxDataStorage, IVxMarker, IVxMonitor (and any other related resources including resources related to its hosted services). If the IVxDevice is assigned to a IVxDataStorage, it shall be unassigned.
GetAccessPoints(VxCollection<IVxAccessPoint**>& accessPointCollection)	Access points hosted by this IVxDevice.
GetAlarmInputs(VxCollection<IVxAlarmInput**>& alarmInputCollection)	Alarm inputs hosted by this IVxDevice.
GetAnalyticSessions(VxCollection<IVxAnalyticSession**>& analyticSessionCollection)	Collection of IVxAnalyticSession for this IVxDevice.
GetDataSources(VxCollection<IVxDataSource**>& dataSourceCollection)	All IVxDataSources hosted by this IVxDevice; omitted if the IVxDevice is not commissioned.
GetDataStorage(IVxDataStorage*& dataStorage)	IVxDataStorage hosted by this IVxDevice.
GetDatabaseBackups(IVxDbBackups*& databaseBackups)	Gets the database backup information for this device, if any.
GetDeviceAssignments(VxCollection<IVxDeviceAssignment**>& deviceAssignmentCollection)	Assignments to a IVxDataStorage for this IVxDevice.
GetDiagnostics(VxDiagnostics*& diagnostics)	VxDiagnostics information for this IVxDevice.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLogs(VxCollection<IVxLog**>& logCollection)	Device log files.
GetMonitors(VxCollection<IVxMonitor**>& monitorCollection)	All IVxMonitors hosted by this IVxDevice; omitted if the IVxDevice is not commissioned.
GetRelayOutputs(VxCollection<IVxRelayOutput**>& relayOutputCollection)	Relay outputs hosted by this IVxDevice.
HasDiagnostics(bool& hasDiagnostics)	Gets a value indicating whether this IVxDevice provides any diagnostic information.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
Replace(const char* replacementDeviceId)	Replace an offline IVxDevice with a new IVxDevice. Not available for all types of devices.
SetDevicePort(int port)	Sets the `port` property.
SetDriverTypeId(char driverTypeId[64])	Sets the `driverTypeId` property.
SetEndpoints(char** endpoints, int endpointsSize)	Sets the `endpoints` property.
SetHostname(char hostname[256])	Sets the `hostname` property. If set, takes precedence over the `ip`.
SetIp(char ip[64])	Sets the `ip` property.
SetName(char name[64])	Sets the `name` property.
SetPassword(char password[64])	Sets the account password used to communicate with the IVxDevice, if any.
SetUsername(char username[64])	Sets the `username` property.
SetVersion(char version[64])	Sets the `version` property.
Silence()	Silence all audible alarms on the IVxDevice.

Method	Description
TriggerRefresh()	Triggers a refresh of this IVxDevice on the VideoXpert system; updating it by retrieving the latest information directly from the device.

5.30 IVxDeviceAssignment

Description	Collection Filters
Represents a IVxDevice assignment that MAY specify the IVxDriver to use to communicate with the physical device.	**kDataSourceId** Only IVxDeviceAssignment associated with this IVxDataSource. **kDataStorageId** Only IVxDeviceAssignment associated with this IVxDataStorage. **kDeviceId** Only IVxDeviceAssignment associated with this IVxDevice. **kModifiedSince** Only IVxDeviceAssignment that have been modified since the given DateTime will be returned.

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxDeviceAssignment identifier.
volumeGroupId	String	Y	R	Unique identifier of the IVxVolumeGroup to use for this IVxDevice.

Method	Description
Delete()	Deletes this instance.
GetDataSources(VxCollection<IVxDataSource**>& dataSourceCollection)	Collection of IVxDataSource that are assigned by this IVxDeviceAssignment.
GetDataStorage(IVxDataStorage*& dataStorage)	The IVxDataStorage that this IVxDeviceAssignment is for.
GetDevice(IVxDevice*& device)	The assigned IVxDevice.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetVolumeGroupId(char volumeGroupId[64])	Sets the `volumeGroupId` property.

5.31 IVxDrawing

Description	Collection Filters
Represents a diagrammatic representation of an area. A IVxDrawing may contain any number of IVxMarkers to represent points of interest within the area. The IVxDrawing resource MAY be locked to prevent other users from modifying the IVxDrawing. Note however that the IVxDrawing resource MAY be modified by any user when not locked.	**kAdvancedQuery** **kImageType** **kModifiedSince** Only IVxDrawing that have been modified since the given DateTime will be returned. **kName** **kProvider**

Field Name	Type	Req	RW	Description
backgroundColor	ColorArgb	N	RW	An integer that defines an ARGB color for the map background.
cameraNumberColor	ColorArgb	N	RW	An integer that defines an ARGB color for the camera number text.
id	String	Y	R	Unique IVxDrawing identifier.

Field Name	Type	Req	RW	Description
lastModified	DateTime	N	R	The time since the IVxDrawing was last modified.
markerSize	Integer	N	RW	The size, in pixels, of marker icons.
mimeType	String	N	R	Internet media type of the image used by this IVxDrawing.
name	String	Y	RW	Friendly name.
provider	VxDrawingProvider	N	R	IVxDrawing mapping provider.
showCameraNumbers	Boolean	N	RW	Indicates whether the camera number text overlays should be displayed.

Method	Description
AddMarker(VxNewMarker& newMarker)	Add a new IVxMarker to the IVxDrawing that this collection belongs to.
Delete()	Deletes this instance.
DeleteDrawing()	Remove the IVxDrawing from the system. This will also remove associated markers and binary drawing data.
DeleteImage()	Delete the binary drawing image data.
GetImage(char* endpoint, int& size)	Retrieve the URI for the binary image data, if any. The endpoint SHOULD be empty if no image data is available.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLock(IVxResourceLock*& resourceLock)	Get the IVxResourceLock, if any, applied to this resource.
GetMarkers(VxCollection<IVxMarker**>& markerCollection)	Get IVxMarkers contained within this drawing.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetBackgroundColor(int backgroundColor)	Sets the `backgroundColor` property.
SetCameraNumberColor(int cameraNumberColor)	Sets the `cameraNumberColor` property.
SetImage(char* imagePath)	Set the binary drawing image data using the path to a local image file located on the client machine. The maximum allowable size of the image is 128 MB.
SetLock()	Set an IVxResourceLock on this resource, owned by the current user. The `SetImage` and `SetName` methods provided by this resource and the `Set` methods on the IVxMarker sub-resource shall return `kResourceLocked` to any user other than the IVxResourceLock owner.
SetMarkerSize(int markerSize)	Sets the `markerSize` property.
SetName(char name[64])	Sets the `name` property.
SetShowCameraNumbers(bool showCameraNumbers)	Sets the `showCameraNumbers` property.

5.32 IVxDriver

Description	Collection Filters
Represents a IVxDevice communication driver.	**kAdvancedQuery** **kModifiedSince** Only IVxDriver that have been modified since the given DateTime will be returned. **kName** **kVendor** **kVersion**

Field Name	Type	Req	RW	Description
name	String	Y	R	Friendly name.
type	String	Y	R	IVxDriver type identifier. The IVxDriver `name`, `vendor`, and `version` are uniquely identified together by this `type`. Note that multiple of the same type of IVxDriver MAY exist in a system.
vendor	String	Y	R	Name of the vendor that manufactured the IVxDevice that the IVxDriver is for.
version	String	Y	R	IVxDriver version number.

Method	Description
Delete()	Deletes this instance.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.33 IVxEvent

Description	Collection Filters
Represents an instance of a particular situation that has occurred. An IVxEvent's `ackState`, through the use of its `Acknowledge` and `Silence` methods, may transition between the states specified by VxAckState (see: Event State Transitions). Several Event Workflow Examples are also available for further reference.	**kAdvancedQuery** **kAckState** **kAckUser** **kGeneratorDeviceId** **kId** **kModifiedSince** Only IVxEvent that have been modified since the given DateTime will be returned. **kNotifies** If true, returns only events that would have been sent to the client based on the event's IVxSituation notification configuration (it is configured to notify the client). Ignored if false. **kSearchEndTime** Acts as an inclusive bounds to the oldest event content to retrieve (DateTime). **kSearchStartTime** Acts as an inclusive bounds to the newest event content to retrieve (DateTime). **kSeverity** **kSituationType** **kSourceDeviceId** **kSourceUserName** **kTime**

Field Name	Type	Req	RW	Description
ackClientId	String	N	R	Client identifier of the client that set the current `ackState`, if any.

Field Name	Type	Req	RW	Description
ackClientName	String	N	R	Friendly name of the client that set the current `ackState`, if any (see: `ackClientId`).
ackState	VxAckState	Y	R	The current acknowledgement state of the event.
ackTime	DateTime	Y	R	The time at which the current `ackState` was set.
ackUser	String	N	R	The user that set the current `ackState`, if any.
generatorDeviceId	String	N	R	Unique identifier of the device that created and injected this IVxEvent into the system. This field MAY be omitted if the generator device is the same as the source device.
generatorDeviceName	String	N	R	Friendly name of the generator IVxDevice, if any (see: `generatorDeviceId`).
id	String	Y	R	Unique IVxEvent identifier created by the device that generated the IVxEvent.
isInitial	Boolean	N	R	True indicates that this IVxEvent is in its initial generated state (its `ackState` has not been updated).
properties	VxKvObject	N	R	A VxKvObject, specific to the VxSituationType, containing additional information related to the event. See the Situations table for the properties required of the event. If the event has no properties, this may be omitted.
propertySize	Integer	N	R	The size of `properties`.
severity	Integer	Y	R	Severity of the event, from 1 (highest) to 10 (lowest).
shouldAudiblyNotify	Boolean	Y	R	True specifies that a notification sound is to play on supporting clients when receiving a notification for this IVxEvent.
situationName	String	N	R	Friendly name of the IVxSituation that led to the generation of this IVxEvent.
situationType	String	Y	R	Identifier for the type of IVxSituation that led to the generation of this IVxEvent.
sourceClientId	String	N	R	Client identifier of the client that was the cause of the situation, if any.
sourceDeviceId	String	Y	R	Unique identifier of the IVxDevice that the situation occurred on. Note that in the case of situations caused by client API calls, this is *not* the client ID, it is the ID of the server that is handling the action.
sourceDeviceName	String	N	R	Friendly name of the source IVxDevice (see: `sourceDeviceId`).
sourceUserName	UPN	N	R	User name of the user that was the cause of the situation, if any.
time	DateTime	Y	R	Time at which the situation occurred.
wakeup	Integer	N	R	Delay, in seconds, prior to bringing the event to the user's attention; typically used for silencing the event.

Method	Description
Acknowledge()	Acknowledge the event. If the IVxEvent is silenced, this shall cancel the silence (it will no longer wake up).
Delete()	Deletes this instance.
GetGeneratorDevice(IVxDevice*& device)	The IVxDevice that the IVxEvent was generated on.
GetSituation(IVxSituation*& situation)	The IVxSituation that this IVxEvent represents.
GetSourceDevice(IVxDevice*& device)	The IVxDevice that the IVxSituation occurred on.
GetUser(IVxUser*& user)	The IVxUser that was the cause of the situation.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

Method	Description
Silence(int wakeup)	Silence the event for a given amount of time. When the IVxEvent wakes, it will return to the `kAckNeeded` state (`ackUser` is cleared).

5.34 IVxExport

Description	Collection Filters
Represents an exported data set that is archived within the system.	**kDataSourceAllTags** Comma-separated list of tag names. Only IVxExport with an IVxExportClip containing a IVxDataSource with *all* of these public tags will be returned. **kDataSourceAllPrivateTags** Comma-separated list of tag names. Only IVxExport with an IVxExportClip containing a IVxDataSource with *all* of these private tags will be returned. **kDataSourceName** Only IVxExport with an IVxExportClip containing a IVxDataSource with this name will be returned. **kDataSourceNumber** Only IVxExport with an IVxExportClip containing a IVxDataSource with this number will be returned. **kModifiedSince** Only IVxExport that have been modified since the given DateTime will be returned. **kName** **kOwner** **kPercentComplete** **kSize** Retrieve IVxExport that are exactly this size. **kStatus** **kTrashed** True to return only trash; false to return only non-trash.

Field Name	Type	Req	RW	Description
completedTime	DateTime	N	R	Date and time at which the triggered IVxExport completed the export operation.
dataUri	String	N	R	The URI to the actual exported data. This endpoint SHOULD only be present if the exported data is completed and available for download.
exportClips	IVxExportClip[]	Y	R	Individual ranges of media data that shall be included in the data archived when this IVxExport is triggered.
exportClipSize	Integer	N	R	The size of `exportClips`.
exportPath	String	N	R	Storage path that the export data is saved to. MAY be omitted if the server's default storage location is being used.
fileSizeKb	Integer	N	R	File size of the exported data in kilobytes (kB).
format	VxExportFormat	Y	R	The format of the exported data.
id	String	Y	R	Unique IVxExport identifier.
initiatedTime	DateTime	Y	R	Date and time at which the IVxExport was requested.

Field Name	Type	Req	RW	Description
isProtected	Boolean	N	R	`True` if the export data is signed and encrypted (requires the password supplied in the VxNewExport to decrypt); `false` otherwise. Defaults to `false`.
isTrashed	Boolean	N	R	`True` if this IVxExport has been marked as trash.
name	String	N	RW	Friendly name.
owner	UPN	N	R	If present, indicates that this resource is owned by a IVxUser (it is private) and this is their username. If not present, indicates that this resource is not owned (it is global).
percentComplete	Float	Y	R	A value from 0 to 100. This specifies how close the export is to completion. 0 indicates that the export has not been triggered; 100 indicates that the export is complete.
secondsRemaining	Integer	N	R	Estimated time remaining, in seconds, until the export is 100 percent complete. If not known, this value will be 0.
status	VxExportStatus	Y	R	Current status of this IVxExport.
statusReason	VxExportStatusReason	N	R	Optional reason for the current status of this IVxExport; typically used to express the reason for a failure.

Method	Description
Delete()	Deletes this instance.
DeleteExport()	Delete this IVxExport and its associated data, if any. If the IVxExport is currently exporting, this will stop the exporting operation prior to deletion of the IVxExport.
GetOwner(IVxUser*& user)	The owner of this IVxExport, if any.
GetPassword(char password[64])	Gets the plain text password for this IVxExport if it's protected. *Note: Only available when logged in as the built-in admin user.*
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
RestoreExport()	Restore this IVxExport to a non-trashed state.
TrashExport()	Mark this IVxExport as trash. Trashed exports MAY be removed by the server if additional space is needed (the oldest trash shall be removed first).

5.35 IVxExportClip

Description	Collection Filters
Represents an individual range of media data belonging to an IVxExport. This data shall be exported when the IVxExport is triggered.	None.

Field Name	Type	Req	RW	Description
dataSourceId	String	Y	R	The IVxDataSource ID of the export media.
endTime	DateTime	Y	R	Time at which the export media ends.
id	String	Y	R	Unique IVxExportClip identifier.
startTime	DateTime	Y	R	Time at which the export media begins.

5.36 IVxExportStream

Description	Collection Filters
Represents streaming access information for an IVxExport resource.	None.

Field Name	Type	Req	RW	Description
exportStreamClips	VxExportStreamClip[]	N	R	The export stream clips, which provide streaming playback information for each IVxExportClip.
exportStreamClipSize	Integer	N	R	The size of `exportStreamClips`.
percentComplete	Float	N	R	A value from 0 to 100. This specifies how close stream preparation is to completion, where 100 indicates that stream is ready for streaming. This will only be populated when the `status` is `kPreparing`, `kFailed`, or `kReady`. In a `kFailed` state, this represents how much progress was made (if any) prior to the failure.
secondsRemaining	Integer	N	R	Estimated time remaining, in seconds, until the export is 100% ready to be streamed. This will only be populated when the `status` is `kPreparing` or `kReady`.
status	VxExportStatus	Y	R	Current status of the export stream.
statusReason	VxExportStatusReason	N	R	Optional reason for the current status of this export stream.

Method	Description
Delete()	Deletes this instance.
Halt()	Halts any pending stream preparation. May only be performed when the export stream `status` is `kPreparing`.
Prepare()	Begins or re-starts preparation for streaming. May only be performed when the export stream `status` is `kNeedsPreparation` or `kFailed`.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.37 IVxFileRecovery

Description	Collection Filters
Represents a file recovery controller/monitoring resource for a recorder.	None.

Field Name	Type	Req	RW	Description
filesRecovered	Integer	N	R	The number of files recovered.
status	VxFileRecoveryStatus	N	R	The current status of the file recovery.
statusMessage	String	N	R	The message about the current status of the file recovery, if any.

Method	Description
Delete()	Deletes this instance.
Halt()	Halts any file recovery in process.

Method	Description
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
Trigger(const char* volumeId, const char* startTime, const char* endTime)	Triggers a file recovery.

5.38 IVxGap

Description	Collection Filters
Represents a span of time that a IVxClip should exist for a particular IVxDataSource on a particular IVxDataStorage but for some reason the IVxClip does not exist. Not all instances of gaps may be detectable by the system and thus may not be reported. Examples of gaps include: • Missing time period within a continuous recording schedule. • Missing time period within any type of recording due to a known reason such as network error or graceful recorder restart. Examples that are not gaps include: • Non-recorded time period between event-based recordings. • Missing time period at the end of recording retention due to garbage collection.	**kDataStorageId** **kSearchEndTime** Inclusive bounds to the latest IVxGap to retrieve (DateTime). **kSearchStartTime** Inclusive bounds to the earliest IVxGap to retrieve (DateTime). **kStartTime**

Field Name	Type	Req	RW	Description
dataSourceId	String	Y	R	Unique identifier of the gapped IVxDataSource.
dataStorageId	String	N	R	Unique identifier of the IVxDataStorage with the gap.
endTime	DateTime	Y	R	End time of the gap.
gapFillerStatus	VxGapFillerStatus	N	R	The status of filling this gap.
reason	VxGapReason	N	R	Reason for this gap. Note that gaps on NSM5200 storage devices will always be `kUnknown`.
reasonData	VxKvObject[]	N	R	Additional reason data. Clients SHOULD be careful about exposing this data to users as it is raw non-standardized data in a non-friendly format. Servers MAY provide whatever data they deem relevant about the reason for this gap, not exceeding 1 MB.
reasonDataSize	String	N	R	The size of `reasonData`.
startTime	DateTime	Y	R	Start time of the gap.

Method	Description
Delete()	Deletes this instance.
GetDataSource(IVxDataSource*& dataSource)	Gets the gapped IVxDataSource.
GetDataStorage(IVxDataStorage*& dataStorage)	Gets the IVxDataStorage with the gap.

5.39 IVxLicense

Description	Collection Filters
Represents license information for the resource that owns this resource (typically a IVxDevice or a IVxSystem).	None.

Field Name	Type	Req	RW	Description
companyName	String[1]	N	R	Friendly name of the owning company / organization. [1]Characters are restricted to uppercase and lowercase letters (`a-z`, `A-Z`), digits (`0-9`), and special characters (`& - _ . # : $ @`).
systemLicenseType	VxSystemLicenseType	Y	R	Type of system that this IVxLicense is applied to.

Method	Description
CommissionDevice(IVxDevice& device)	Specify resources of type IVxDevice to commission them (this will consume an available IVxLicenseFeature `count`). Any resources already commissioned have no effect (considered successful). If any of the resources do not exist or are not applicable, `kInvalidValue` is returned. If any of the resources are deleted, their commission shall be released for reuse. If an attempt to commission more resources than the IVxLicenseFeature `count` allows for, `kLicenseCountExceeded` is returned. If no IVxLicenseFeature is available for commissioning, `kNoLicense` is returned.
DecommissionDevice(IVxDevice& device)	Remove IVxDevice commissions (this will release used commissions for reuse). For any resources that do not exist, this has no effect (considered successful).
Delete()	Deletes this instance.
GetLicenseFeatures(VxCollection<IVxLicenseFeature**>& licenseFeatureCollection)	A list of features that this IVxLicense contains.
GetLicenseSup(VxLicenseSup*& licenseSup)	Gets the software upgrade licensing information. *Note: Only available when logged in as the built-in admin user.*

5.40 IVxLicenseFeature

Description	Collection Filters
Represents device functionality that is enabled with a valid license.	None.

Field Name	Type	Req	RW	Description
activationId	String	N	R	The activation identifier.
count	Integer	N	R	The allowable commisions count.
expiration	DateTime	N	R	The date and time at which the feature will expire. If the feature does not have an expiration, this field will be omitted.
id	String	Y	R	Unique IVxLicenseFeature identifier.
installation	DateTime	Y	R	The date and time at which the feature was installed on the host.
isPending	Boolean	Y	R	True if this is a "pending" IVxLicenseFeature; this is a IVxLicenseFeature that an activation has been requested for but for which no valid license has yet been applied. A pending IVxLicenseFeature does not enable any functionality. False indicates this is a normal IVxLicenseFeature that enables device functionality.
name	String	N	R	The name of the feature; REQUIRED if `isPending` is false.
used	Integer	Y	R	Amount of the allowable `count` that is currently used by commissions. The remaining commissions available can be calculated by `count - used`.

Field Name	Type	Req	RW	Description
version	String	N	R	The feature version; REQUIRED if `isPending` is false.

Method	Description
Delete()	Deletes this instance.
GetCommissionedDevices(VxCollection<IVxDevice**>& deviceCollection)	Returns an IVxDevice collection representing all IVxDevices that have been commissioned for this IVxLicenseFeature.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.41 IVxLog

Description	Collection Filters
Represents a log file or set of log files	**kInitiated** **kModifiedSince** Only IVxLog that have been modified since the given DateTime will be returned.

Field Name	Type	Req	RW	Description
completed	DateTime	Y	R	Date and time at which the IVxLog was completed.
id	String	Y	R	Unique IVxLog identifier.
initiated	DateTime	N	R	Date and time at which the IVxLog was requested.

Method	Description
Delete()	Deletes this instance.
DeleteLog()	Delete this IVxLog and its associated data.
GetLogEndpoint(char* endpoint, int& size)	Get the URI to the actual IVxLog data to download. May be presented in whatever format is suitable to the server. This link will only be present if the log data is complete and available for download.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.42 IVxManualRecording

Description	Collection Filters
DEPRECATED: Replaced by IVxRecording Represents an instance of an active manual recording (the action, not the recorded clip). Each IVxManualRecording will cause its specified IVxDataSource to be manually recorded by all IVxDataStorage that the IVxDataSource is assigned to. Only 1 IVxManualRecording per IVxUser per IVxDataSource is allowed. A IVxClip created via the application of a IVxManualRecording will have an `event` value of `kManual`.	**kAdvancedQuery** **kDataSourceId** **kId** **kModifiedSince** Only IVxManualRecording that have been modified since the given DateTime will be returned. **kOwner**

Field Name	Type	Req	RW	Description
dataSourceId	String	Y	R	Identifier of the IVxDataSource to manually record.
id	String	Y	R	Unique IVxManualRecording identifier.
owner	UPN	Y	R	The name of the user that created this IVxManualRecording.

Field Name	Type	Req	RW	Description
time	DateTime	N	R	Time at which this IVxManualRecording was created (not necessarily the exact time that the manual record IVxClip starts).

Method	Description
Delete()	Deletes this instance.
DeleteManualRecording()	Delete this IVxManualRecording. If all IVxManualRecording are deleted for a IVxDataSource, manual recording will be stopped for that IVxDataSource.
GetDataSource(IVxDataSource*& dataSource)	The IVxDataSource that is being manually recorded.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
ResetExpirationTimers()	Reset expiration timers in order to extend this recording.

5.43 IVxMarker

Description	Collection Filters
Represents the location of a resource on a IVxDrawing. A IVxMarker is associated with exactly 1 IVxDrawing, though a IVxDrawing may have many IVxMarkers. A IVxMarker may be associated with 1 resource, though a resource may have many IVxMarkers.	**kLayerName** **kModifiedSince** Only IVxMarker that have been modified since the given DateTime will be returned. **kResourceId** Return only markers with the id of the associated IVxDataSource resource.

Field Name	Type	Req	RW	Description
direction	Float	N	RW	Angular coordinate indicating the direction, if any, that the IVxMarker is facing on a polar grid (e.g.: 0 (right), 90 (up), 180 (left), 270 (down)).
id	String	Y	R	Unique IVxMarker identifier.
layerName	String	N	RW	Friendly name of the layer that this IVxMarker is on. Clients MAY group IVxMarker with matching layer names.
name	String	N	RW	Friendly name.
resource	VxResourceType	N	R	The type of resource associated with this IVxMarker. Can be used to determine which `GetAssociation` method to use in order to retrieve the associated resource.
x	Float	Y	RW	X Cartesian coordinate.
y	Float	Y	RW	Y Cartesian coordinate.

Method	Description
Delete()	Deletes this instance.
DeleteAssociation()	Deletes any association in use for this IVxMarker.
DeleteMarker()	Delete this IVxMarker.
GetAssociation(IVxDataSource*& dataSource)	Retrieve the associated IVxDataSource, if any.
GetAssociation(IVxDrawing*& drawing)	Retrieve the associated IVxDrawing, if any.
GetDrawing(IVxDrawing*& drawing)	Retrieve the IVxDrawing that this IVxMarker belongs to.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.

Method	Description
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetAssociation(IVxDataSource& dataSource)	Sets the IVxDataSource associated with this IVxMarker.
SetAssociation(IVxDrawing& drawing)	Sets the IVxDrawing associated with this IVxMarker.
SetDirection(float direction)	Sets the `direction` property.
SetLayerName(char layerName[64])	Sets the `layerName` property.
SetName(char name[64])	Sets the `name` property.
SetCoordinates(float x, float y)	Sets the `x` and `y` properties.

5.44 IVxMember

Description	Collection Filters
Represents a member system.	**kAdvancedQuery** **kHost** **kId** **kModifiedSince** Only IVxMember that have been modified since the given DateTime will be returned. **kName** **kState**

Field Name	Type	Req	RW	Description
bandwidth	Integer	Y	RW	Bandwidth available between the aggregator system and the member system (in kbps).
host	String	Y	RW	The host address.
hostPort	Integer	Y	RW	The host port.
id	String	Y	R	The unique IVxSystem identifier (`IVxSystem::id`) of the IVxMember system.
name	String	Y	R	The friendly name for the member system. This value comes from the IVxSystem`::name` field, containing the system's friendly name. This is not an alias for the system.
rtspCapability	VxRtspCapability	N	RW	Network streaming capabilities that this IVxMember shall expose to its clients (via IVxDataInterface).
state	VxMemberState	Y	R	The current operational state.
username	String	Y	RW	The administrator account username used to communicate with this member.

Method	Description
Delete()	Deletes this instance.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
RemoveMember()	Removes the member system; it will no longer be aggregated.
SetBandwidth(int bandwidth)	Sets the `bandwidth` property.
SetHost(char host[256])	Sets the `host` property.
SetPassword(char password[64])	Sets the administrator account password used to communicate with this member.

Method	Description
SetHostPort(int hostPort)	Sets the `hostPort` property.
SetRtspCapability(VxRtspCapability::Value rtspCapability)	Sets the `rtspCapability` property.
SetUsername(char username[64])	Sets the administrator account username used to communicate with this member.
TriggerRefresh()	Triggers a refresh of this member system; ensures that the aggregating system is in sync with the member system.

5.45 IVxMonitor

Description	Collection Filters
A IVxMonitor represents a display for viewing data (typically video).	**kAdvancedQuery** **kId** **kModifiedSince** Only IVxMonitor that have been modified since the given DateTime will be returned. **kName** **kNumber**

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxMonitor identifier.
layout	VxCellLayout	N	RW	Cell grid layout.
maximizedCell	Integer	N	RW	The index of the full screen IVxMonitorCell *(-1 if no cells are maximized)*.
name	String	N	RW	Friendly name.
number	Integer	N	RW	A unique number used to designate the IVxMonitor.
syncSpeed	Float	N	RW	The play speed of the display data.
syncTime	DateTime	N	RW	The time at which the data should initially seek to (does not track time as the data plays).
syncTimeAnchor	DateTime	N	R	The wall clock time at which the data playback should begin.

Method	Description
Delete()	Deletes this instance.
GetAvailableLayouts(VxCollection<VxCellLayout::Value*>& layoutCollection)	Gets the VxCellLayouts available for this IVxMonitor.
GetHostDevice(IVxDevice*& hostDevice)	Host IVxDevice of this IVxMonitor (e.g an OCC shared display).
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetMonitorCells(VxCollection<IVxMonitorCell**>& cellCollection)	An ordered list of all IVxMonitorCell currently active on the IVxMonitor (cells in the active tab of the active window).
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
RemoveMonitor()	Delete the IVxMonitor from the system.
SetLayout(VxCellLayout::Value layout)	Sets the `layout` property.
SetMaximizedCell(int cellIndex)	Sets the `maximizedCell` property.
SetName(char name[64])	Sets the `name` property.

Method	Description
SetNumber(int number)	Sets the `number` property.
SetSyncSpeed(float syncSpeed)	Sets the play speed of the display data. Negative values indicate reverse speeds while positive values indicate forward speeds (`1` is normal speed). A value of `0` will pause the data playback.
SetSyncTime(char syncTime[64])	Sets the time to seek to for all synced cells on the IVxMonitor. A value of `nullptr` will set the streams to live.

5.46 IVxMonitorCell

Description	Collection Filters
A IVxMonitorCell represents a single viewport, hosted on a IVxMonitor, that can play media from a IVxDataSource.	None.

Field Name	Type	Req	RW	Description
cropHeight	Float	N	RW	The height of the crop bounds expressed as a percentage `(0-1)`. *(e.g. 0.5 for 50% of the video frame height)*
cropWidth	Float	N	RW	The width of the crop bounds expressed as a percentage `(0-1)`. *(e.g. 0.5 for 50% of the video frame width)*
dataSourceId	String	N	RW	The IVxDataSource that this cell shall display.
index	Integer	Y	R	Unique IVxMonitorCell index (unique per active tab on the host IVxMonitor). Matches the position of the cell in the host IVxMonitor `cells` list (1-based; the first item is index `1`).
isAnalyticsOverlayEnabled	Boolean	N	RW	Indicates whether or not the analytics overlay is enabled.
isInSync	Boolean	N	RW	Indicates whether or not the IVxMonitorCell is part of the IVxMonitor's sync group.
isStatisticsOverlayEnabled	Boolean	N	RW	Indicates whether or not the statistics overlay is enabled.
rotation	VxRotationType	N	RW	The rotation applied to the video.
speed	Float	N	RW	Play speed of the display data. Negative values indicate reverse speeds while positive values indicate forward speeds (`1` is normal speed). A value of `0` will pause the data playback. Defaults to `1`.
time	DateTime	N	RW	Time at which the data should initially seek to (does not track time as the data plays). Omitted if live. If no frame is available at this time, `kInvalidValue` is returned and time will remain unchanged. *Note: Updating this attribute will automatically update the `timeAnchor` to the current time.*
timeAnchor	DateTime	N	R	Wall clock time at which the data playback should begin. In general, this will be the time at which `time` was last modified. Automatically updated by the server whenever `time` is modified.

Field Name	Type	Req	RW	Description
x	Integer	N	RW	The X location of the crop bounds or immersive position.
y	Integer	N	RW	The Y location of the crop bounds or immersive position.
z	Integer	N	RW	The Z location of the immersive position.

Method	Description
Delete()	Deletes this instance.
DisableAnalyticsOverlay()	Disables the analytics overlay.
DisableStatisticsOverlay()	Disables the statistics overlay.
EnableAnalyticsOverlay()	Enables the analytics overlay.
EnableStatisticsOverlay()	Enables the statistics overlay.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
JoinSyncGroup()	Adds this IVxMonitorCell is to the IVxMonitor's sync group.
LeaveSyncGroup()	Removes this IVxMonitorCell is to the IVxMonitor's sync group.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetCropBounds(float height, float width)	Sets the height and width of the crop bounds applied to the video.
SetCropHeight(float height)	Sets the height of the crop bounds applied to the video.
SetCropWidth(float width)	Sets the width of the crop bounds applied to the video.
SetDataSource(char dataSourceId[1024])	Sets the data source that this cell shall display. A value of `nullptr` will remove the current data source.
SetLocation(int x, int y, int z)	Sets the X, Y and Z locations of the crop bounds or immersive position.
SetLocationX(int x)	Sets the X location of the crop bounds or immersive position.
SetLocationY(int y)	Sets the Y location of the crop bounds or immersive position.
SetLocationZ(int z)	Sets the Z location of the immersive position.
SetRotation(VxRotationType::Value rotation)	Sets the rotation applied to the video.
SetSpeed(float speed)	Sets the play speed of the monitor data.
SetTime(char time[64])	Sets the time to seek to on the monitor. A value of `nullptr` will set the stream to live. If no frame is available for the given time the stream will remain unchanged.

5.47 IVxMonitorWall

Description	Collection Filters
A IVxMonitorWall represents a group of IVxMonitor.	**kAdvancedQuery** **kModifiedSince** Only IVxMonitorWall that have been modified since the given DateTime will be returned. **kName**

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxMonitorWall identifier.
name	String	N	RW	Friendly name.
monitorPositions	VxMonitorPosition[]	N	RW	List of VxMonitorPosition for this IVxMonitorWall. A IVxMonitor may only exist at a single position per IVxMonitorWall–an attempt to set the same IVxMonitor more than once here will result in a `kInvalidValue` response.
monitorPositionsSize	Integer	N	R	The size of `monitorPositions`.

Method	Description
ClearMonitorSelection()	Clears the current VxMonitorPosition from this IVxMonitorWall.
Delete()	Deletes this instance.
DeleteMonitorWall()	Delete this IVxMonitorWall.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetMonitors(VxCollection<IVxMonitor**>& monitorCollection)	A collection of IVxMonitor associated with this IVxMonitorWall.
GetMonitorSelections(VxCollection<VxMonitorSelection**>& monitorSelectionCollection)	A collection of VxMonitorSelection associated with this IVxMonitorWall.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetName(char name[64])	Sets the `name` property.
SetMonitorPositions(VxMonitorPosition* monitorPositions, int monitorPositionsSize)	Sets the `monitorPositions` property.
SetMonitorSelection(int cellIndex, int monitorIndex, VxCellInputMode::Value inputMode)	Updates the monitor/cell selection (for this user) on the IVxMonitorWall. *Note: Sending a value of 0 for the cellIndex or monitorIndex fields will clear that selection from the IVxMonitorWall.*

5.48 IVxNotification

Description	Collection Filters
Represents a particular notification configuration for a IVxSituation. This includes the list of recipients that should receive the event.	**kModifiedSince** Only IVxNotification that have been modified since the given DateTime will be returned.

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxNotification identifier.
roleIds	String[]	Y	R	IDs of the IVxRoles for which the constituent users should receive this notification.
roleIdSize	Integer	N	R	The size of `roleIds`.

Method	Description
AddRole(IVxRole& role)	Adds the IVxRole to the list of roles that this IVxNotification serves.
Delete()	Deletes this instance.
DeleteNotification()	Delete this IVxNotification

Method	Description
GetRoles(VxCollection<IVxRole**>& roleCollection)	The collection of IVxRole that will receive this IVxNotification.
RemoveRole(IVxRole& role)	Removes the IVxRole from the list of roles that this IVxNotification serves.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.49 IVxPattern

Description	Collection Filters
Represents a predefined sequence of movement that a PTZ device can perform when triggered. A running pattern will be halted upon any other PTZ request.	**kModifiedSince** Only IVxPattern that have been modified since the given DateTime will be returned.

Field Name	Type	Req	RW	Description
description	String	N	R	Friendly description of this IVxPattern.
name	String	Y	R	Friendly name.

Method	Description
Delete()	Deletes this instance.

5.50 IVxPixelSearch

Description	Collection Filters
Represents the results of a pixel search. Servers will automatically delete this resource due to inactivity (no clients accessing this resource or its clips for at least `60` seconds once the VxSearchStatus is set to `kComplete`.	None.

Method	Description
Delete()	Deletes this instance.
DeletePixelSearch()	Delete this IVxPixelSearch
GetClips(VxCollection<IVxClip**>& clipCollection)	Results of the pixel search; new results MAY continue to be available while the VxSearchStatus is `kInProgress`.
GetSearchStatus(VxSearchStatus::Value& status)	Current status of the pixel search.

5.51 IVxPreset

Description	Collection Filters
Represents a predefined point that a PTZ device can PTZ to when triggered. Note that digital presets are simply stored coordinates that clients can digitally PTZ to (no camera movement occurs). Digital presets are typically used with immersive cameras.	**kModifiedSince** Only IVxPreset that have been modified since the given DateTime will be returned.

Field Name	Type	Req	RW	Description
description	String	N	R	Friendly description of this IVxPreset.
index	Integer	N	R	Unique numerical sequence value of this IVxPreset.
isDigital	Boolean	N	R	True if this is a digital preset; false otherwise.
name	String	Y	R	Friendly name. *NOTE: This attribute is currently used more as an ID than as a friendly name–use description instead.*
x	Float	N	R	X (pan) coordinate absolute position, in degrees, relative to the (0, 0) position. The `-180` position is leftmost while the `180` position is rightmost. *For digital presets only.*
y	Float	N	R	Y (tilt) coordinate absolute position, in degrees, relative to the (0, 0) position. The `-180` position is bottommost while the `180` position is topmost. *For digital presets only.*
z	Float	N	R	Z (zoom) coordinate absolute position relative to the `0` position. The `0` position is minimum zoom while the `100` position is maximum zoom. *For digital presets only.*

Method	Description
Delete()	Deletes this instance.

5.52 IVxPrivilege

Description	Collection Filters
Represents a specific permission that is given to a IVxRole.	None.

Field Name	Type	Req	RW	Description
excludeRestricted	Boolean	N	RW	True excludes this IVxPrivilege from associated resources. False indicates that this IVxPrivilege follows normal resource restrictions. *Note: Modifying this setting will also change all of this IVxPrivilege's parent/child IVxPrivilege's to the same `excludeRestricted` value.*
id	String	Y	R	Unique IVxPrivilege identifier.
permissionId	VxPermissionId	Y	R	Permission being granted by this IVxPrivilege.
resourceType	VxResourceType	N	R	Resource type supported by this IVxPrivilege, if any.

Method	Description
Delete()	Deletes this instance.
DeletePrivilege()	Remove this IVxPrivilege from its IVxRole. If the IVxRole contains a child permission it MUST be deleted first.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLinks(VxCollection<IVxDataSource**>& dataSourceCollection)	Returns an IVxDataSource collection representing the resources associated with this IVxPrivilege. The IVxPrivilege is restricted to these resources if restriction level is `true`.
GetLinks(VxCollection<IVxDevice**>& deviceCollection)	Returns an IVxDevice collection representing the resources associated with this IVxPrivilege. The IVxPrivilege is restricted to these resources if restriction level is `true`.

Method	Description
GetLinks(VxCollection<IVxDrawing**>& drawingCollection)	Returns an IVxDrawing collection representing the resources associated with this IVxPrivilege. The IVxPrivilege is restricted to these resources if restriction level is `true`.
GetLinks(VxCollection<IVxRelayOutput**>& relayOutputCollection)	Returns an IVxRelayOutput collection representing the resources associated with this IVxPrivilege. The IVxPrivilege is restricted to these resources if restriction level is `true`.
GetLinks(VxCollection<IVxUser**>& userCollection)	Returns an IVxUser collection representing the resources associated with this IVxPrivilege. The IVxPrivilege is restricted to these resources if restriction level is `true`.
GetPriority(int& priority)	Gets the relative priority for this IVxPrivilege, from 1 to 250. Smaller numbers have higher priority than larger numbers (e.g. 1 is the highest priority).
GetRestricted(bool& isRestricted)	Gets the restriction level for this IVxPrivilege. True restricts this IVxPrivilege to associated resources. False indicates that this IVxPrivilege has no resource restrictions. True is only valid if the permission has available resource restrictions.
GetUnLinked(VxCollection<IVxDataSource**>& dataSourceCollection)	Returns IVxDataSources *not* associated with this IVxPrivilege.
GetUnLinked(VxCollection<IVxDevice**>& deviceCollection)	Returns IVxDevices *not* associated with this IVxPrivilege.
GetUnLinked(VxCollection<IVxDrawing**>& drawingCollection)	Returns IVxDrawings *not* associated with this IVxPrivilege.
GetUnLinked(VxCollection<IVxRelayOutput**>& relayOutputCollection)	Returns IVxRelayOutputs *not* associated with this IVxPrivilege.
GetUnLinked(VxCollection<IVxUser**>& userCollection)	Returns IVxUsers *not* associated with this IVxPrivilege.
Link(IVxDataSource& dataSource)	Associate an IVxDataSource with this IVxPrivilege. This grants authority to use this IVxPrivilege on the given resources. Resources not associated with this IVxPrivilege SHALL NOT be authorized for this IVxPrivilege. Any associations already present have no effect (considered successful). If any of the associated resources are deleted, their association here shall also be removed.
Link(IVxDevice& device)	Associate an IVxDevice with this IVxPrivilege. This grants authority to use this IVxPrivilege on the given resources. Resources not associated with this IVxPrivilege SHALL NOT be authorized for this IVxPrivilege. Any associations already present have no effect (considered successful). If any of the associated resources are deleted, their association here shall also be removed.

Method	Description
Link(IVxDrawing& drawing)	Associate an IVxDrawing with this IVxPrivilege. This grants authority to use this IVxPrivilege on the given resources. Resources not associated with this IVxPrivilege SHALL NOT be authorized for this IVxPrivilege. Any associations already present have no effect (considered successful). If any of the associated resources are deleted, their association here shall also be removed.
Link(IVxRelayOutput& relayOutput)	Associate an IVxRelayOutput with this IVxPrivilege. This grants authority to use this IVxPrivilege on the given resources. Resources not associated with this IVxPrivilege SHALL NOT be authorized for this IVxPrivilege. Any associations already present have no effect (considered successful). If any of the associated resources are deleted, their association here shall also be removed.
Link(IVxUser& user)	Associate an IVxUser with this IVxPrivilege. This grants authority to use this IVxPrivilege on the given resources. Resources not associated with this IVxPrivilege SHALL NOT be authorized for this IVxPrivilege. Any associations already present have no effect (considered successful). If any of the associated resources are deleted, their association here shall also be removed.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetExcludeRestricted(bool excludeRestricted)	Sets whether this IVxPrivilege is excluded from the associated resources.
SetPriority(int priority)	Sets the relative priority for this IVxPrivilege, from 1 to 250. Smaller numbers have higher priority than larger numbers (e.g. 1 is the highest priority).
SetRestricted(bool isRestricted)	Sets the restriction level for this IVxPrivilege. True restricts this IVxPrivilege to associated resources. False indicates that this IVxPrivilege has no resource restrictions. True is only valid if the permission has available resource restrictions.
UnLink(IVxDataSource& dataSource)	Remove an IVxDataSource association from this IVxPrivilege. For any associations that do not exist, this has no effect (considered successful).
UnLink(IVxDevice& device)	Remove an IVxDevice association from this IVxPrivilege. For any associations that do not exist, this has no effect (considered successful).
UnLink(IVxDrawing& drawing)	Remove an IVxDrawing association from this IVxPrivilege. For any associations that do not exist, this has no effect (considered successful).
UnLink(IVxRelayOutput& relayOutput)	Remove an IVxRelayOutput association from this IVxPrivilege. For any associations that do not exist, this has no effect (considered successful).
UnLink(IVxUser& user)	Remove an IVxUser association from this IVxPrivilege. For any associations that do not exist, this has no effect (considered successful).

5.53 IVxPtzController

Description	Collection Filters
Exists for a resource that has pan, tilt, and/or zoom functionality. This resource can be used to manipulate the PTZ movement of its parent resource. Absolute positioning is used, for supporting devices, to move the field of view to the given coordinates. Continuous positioning is used, for supporting devices, to continuously move the field of view at the given velocity speed until stopped. Velocity speeds are percentage based and can be both positive and negative (-100 to +100). Negative X values pan left and positive X values pan right while negative Y values tilt downward and positive Y values tilt upward (see: Continuous Move Velocity Diagram). A IVxPtzController can be locked by using its associated IVxPtzLock resource (see: `GetPtzLock`). When the IVxPtzController is locked, only the IVxPtzLock owner can update the IVxPtzController. If a different user attempts to update the IVxPtzController, they will receive a `kCameraLocked` response. However, if they are an equal or higher priority user than the IVxPtzLock owner, they will instead receive a `kNeedOverride` response indicating that they MAY claim the IVxPtzLock.	None.

Field Name	Type	Req	RW	Description
isLocked	Boolean	N	R	Tells whether this IVxPtzController is locked or not. When this is true, all IVxPtzController fields are to be considered read-only to everyone other than the user who owns the IVxPtzLock.
lockExpireTime	Integer	N	R	The time remaining (in seconds) until the IVxPtzLock expires.

Method	Description
AbsoluteMove(int positionX, int positionY)	Moves to the absolute position of the given coordinates.
AbsoluteMove(int positionX, int positionY, int positionZ)	Moves to the absolute position of the given coordinates.
AbsolutePan(int positionX)	Pans to the absolute position of the given coordinate.
AbsoluteTilt(int positionY)	Tilts to the absolute position of the given coordinate.
AbsoluteZoom(int positionZ)	Zooms to the absolute position of the given coordinates.
AddPreset(int index)	Creates a new preset using the current PTZ spatial coordinates.
ContinuousFocus(VxFocusDirection::Value nearFar)	Continuously focuses near or far.
ContinuousIris(VxIrisDirection::Value openClose)	Continuously opens or closes the iris.
ContinuousMove(int speedX, int speedY, VxZoomDirection::Value inOut)	Continuously moves the field of view at the given speed until stopped.
Delete()	Deletes this instance.
DeletePreset(IVxPreset& preset)	Deletes a preset from the system.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetPatterns(VxCollection<IVxPattern**>& patternCollection)	Retrieve PTZ patterns.
GetPosition(int& positionX, int& positionY, int& positionZ)	Gets the current absolute position coordinates.

Method	Description
GetPresets(VxCollection<IVxPreset**>& presetCollection)	Retrieve PTZ presets.
GetPtzLimits(VxPtzLimits*& ptzLimits)	Gets the value limits for this PTZ controller.
GetPtzLock(IVxPtzLock*& ptzLock)	Retrieve the IVxPtzLock for this IVxPtzController. This resource provides PTZ lock information and control.
PtzStop()	Stops all PTZ actions.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
RelativeMove(int deltaX, int deltaY, int deltaZ)	Moves the camera position relative to the current position.
RelativePercentageMove(int percentageX, int percentageY)	PTZ the device within a percentage relative to the device's current field of view. The rotational x and y parameters are used to move within a percentage of the current field of view of the camera. For instance, if the x parameter were 50 and the y parameter were -50, the pan/tilt would move halfway to the edge of the field of view along the x axis and halfway to the field of view along the negative y axis from its present position.
RepositionPreset(IVxPreset& preset)	Repositions a preset to the current PTZ spatial coordinates.
TriggerPattern(IVxPattern& pattern)	Triggers this pattern; PTZ the camera according to the predefined sequence of movement that this IVxPattern represents. *Note: Other PTZ requests may cause this pattern to halt.*
TriggerPreset(int index)	Trigger the PTZ preset at the given index. Useful for triggering presets unlisted in IVxPresets such as special preset functions (e.g. auto pan, random pan, wiper, etc). *Note: MAY succeed even on cameras without support for the given preset.*
TriggerPreset(IVxPreset& preset)	PTZ the camera to the position that this IVxPreset represents.
TriggerRefresh()	Trigger a refresh of this IVxPtzController; updates this IVxPtzController's IVxPatterns and IVxPresets based on current camera configuration.

5.54 IVxPtzLock

Description	Collection Filters
Represents the lock for a IVxPtzController.	None.

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxPtzLock identifier.

Method	Description
Delete()	Deletes this instance.
GetExpireTime(int& expireTime)	Gets the amount of time (in seconds) that the IVxPtzLock will be held if not explicitly unlocked. If locked and this field is empty, then the lock will not expire until it is explicitly unlocked.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLockState(bool& isLocked)	Gets the current state of the PTZ lock. If true, then the associated IVxPtzController will be locked and only the lock owner, or a user with equal or higher priority, can modify the IVxPtzController and IVxPtzLock.
GetOwner(char* owner, int& size)	Retrieve the user name that owns this IVxPtzLock, if any.
Lock(int expireTime)	Locks the IVxPtzController.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
Unlock()	Unlocks the IVxPtzController.

5.55 IVxRecording

Description	Collection Filters
Represents an instance of an active recording (the action, not the recorded clip). Each IVxRecording will cause its specified IVxDataSource to be recorded by all IVxDataStorage that the IVxDataSource is assigned to.	**kAdvancedQuery** **kDataSourceId** **kId** **kModifiedSince** Only IVxManualRecording that have been modified since the given DateTime will be returned. **kName** **kOwner** **kRecordType**

Field Name	Type	Req	RW	Description
dataSourceId	String	Y	R	Identifier of the IVxDataSource to record.
endEvent	String	N	R	If specified, the recording will automatically end when a matching event occurs.
endEventSourceId	String	N	R	If specified, only events from the specified source will end the recording.
endEventSourceType	VxResourceType	N	R	If specified, the resource type of `endEventSourceId`.
framerate	VxRecordingFramerate	N	R	The framerate to record at.
id	String	Y	R	Unique IVxRecording identifier.
maxRecordingTime	Integer	N	R	Maximum amount of time, in seconds, to record before stopping.
name	String	N	R	Friendly name of the recording which can be used to identify recordings to stop.
owner	UPN	Y	R	The name of the user that created this IVxManualRecording.
postRecord	Integer	N	R	Amount of time, in seconds, to record after the recording would otherwise be stopped.
preRecord	Integer	N	R	Amount of time, in seconds, to record prior to the recording start time.
recordType	VxRecordingType	N	R	The type of this recording which will show up in the corresponding clips.

Field Name	Type	Req	RW	Description
startTime	DateTime	N	R	Time at which the recording should begin (may be in the recent past). The recording `preRecord` and `maxRecordingTime` will reference to this time. Defaults to the time at which the server processes the request.

Method	Description
Delete()	Deletes this instance.
DeleteRecording()	Delete this IVxRecording. If all IVxRecording are deleted for a IVxDataSource, recording will be stopped for that IVxDataSource.
GetDataSource(IVxDataSource*& dataSource)	The IVxDataSource that is being recorded.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.

5.56 IVxRelayOutput

Description	Collection Filters
Represents a physical relay output. *Note: For legacy devices, this represents a single relay (auxiliary) output in the `RelayArrayConfiguration`.*	**kAdvancedQuery** **kEnabled** **kId** **kModifiedSince** Only IVxRelayOutput that have been modified since the given DateTime will be returned. **kName** **kState**

Field Name	Type	Req	RW	Description
description	String	N	RW	Friendly description.
id	String	Y	R	Unique IVxRelayOutput identifier.
isEnabled	Boolean	N	RW	True if this IVxRelayOutput is enabled.
name	String	N	RW	Friendly name.
state	VxRelayState	N	R	Current state of the IVxRelayOutput.

Method	Description
Activate()	Activate this IVxRelayOutput.
Deactivate()	Deactivate this IVxRelayOutput.
Delete()	Deletes this instance.
Disable()	Disable this IVxRelayOutput.
Enable()	Enable this IVxRelayOutput.
GetHostDevice(IVxDevice*& hostDevice)	Host IVxDevice of this IVxRelayOutput.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetDescription(char[255] description)	Sets the `description` property.
SetName(char name[64])	Sets the `name` property.

5.57 IVxResourceLock

Description	Collection Filters
A IVxResourceLock represents a lock upon a resource. A resource that has been locked shall permit only the owner of the lock access to any editable values for the resource.	None.

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxResourceLock identifier.
owner	UPN	Y	R	User name of the user that owns the lock. Only this user is authorized to modify a resource that owns this lock.

Method	Description
Delete()	Deletes this instance.
DeleteResourceLock()	Delete this IVxResourceLock. This will unlock the resource that this IVxResourceLock is applied to.

5.58 IVxResourceRel

Description	Collection Filters
An IVxResourceRel represents a related resource; that is, a resource that is related in some way to a parent resource. This related resource may be "linked" to its parent resource for some effect. The effect of linking depends on the parent being linked to. With regards to permissions, an IVxResourceRel is considered to be the same resource as the related resource (e.g. READ access is required on the related resource in order to READ the IVxResourceRel).	**kAdvancedQuery** **kAllTags** Comma-separated list of public tag names. Only resource relations for a resource tagged by *all* of these public tags will be returned. **kAllPrivateTags** Comma-separated list of private tag names. Only resource relations for a resource tagged by *all* of these private tags, owned by the current user, will be returned. **kLinked**

Field Name	Type	Req	RW	Description
isLinked	Boolean	Y	RW	True if the related resource is linked to its parent; false otherwise.

Method	Description
Delete()	Deletes this instance.
GetResource(IVxDataSource*& dataSource)	Retrieve the related resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetLinkState(bool isLinked)	Sets the `isLinked` property.

5.59 IVxRole

Description	Collection Filters
Represents a collection of permissions which can be assigned to a user.	**kAdvancedQuery** **kId** **kInternal** **kModifiedSince** Only IVxRole that have been modified since the given DateTime will be returned. **kName**

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxRole identifier.
isReadOnly	Boolean	Y	R	True indicates that the IVxRole was created internally on the server and is read-only; clients may *not* modify it. `DeleteRole` and `Set` methods are unavailable for read-only roles. *Note: A read-only IVxRole is for internal use only and SHOULD NOT be provided to clients.*
name	String	Y	RW	Unique friendly name for the role.

Method	Description
AddPrivilege(VxNewPrivilege& newPrivilege)	Add a new IVxPrivilege to the IVxRole that this collection belongs to. If the IVxPrivilege permission already exists for the IVxRole, `kConflict` is returned. If the IVxRole does not already contain the parent permission for the permission being assigned, if one exists, `kPermissionConflict` is returned.
Delete()	Deletes this instance.
DeleteRole()	Deletes the IVxRole, and all of its IVxPrivileges, from the system.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetPrivileges(VxCollection<IVxPrivilege**>& privilegeCollection)	All IVxPrivilege assigned to this IVxRole.
GetUsers(VxCollection<IVxUser**>& userCollection)	The users which are currently assigned this role.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetName(char name[64])	Sets the `name` property.

5.60 IVxRule

Description	Collection Filters
Represents a set of VxRuleTrigger that, when any occur during times when the IVxRule is active, cause a script to run. A IVxRule can be read as follows: "If enabled, when any specified triggers occur during specified times, run the associated script."	**kAdvancedQuery** **kId** **kModifiedSince** Only IVxRule that have been modified since the given DateTime will be returned. **kName** **kNumber**

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxRule identifier.
isEnabled	Boolean	N	RW	True if this IVxRule is enabled.
name	String	N	RW	Friendly name.
timeTableIds	String[]	N	RW	The IVxRule will only automatically run its script if a trigger occurs during the times contained by a IVxTimeTable specified here. If empty, no active time ranges are available for this IVxRule (it must be manually triggered). If `null`, no time filters will be applied (the IVxRule is always active).
timeTableIdSize	Integer	N	R	The size of `timeTableIds`.

Field Name	Type	Req	RW	Description
triggers	VxRuleTrigger[]	N	RW	List of VxRuleTrigger that, when any activate, cause the IVxRule to run its script (if activated during an active time).
triggerSize	Integer	N	R	The size of `triggers`.

Method	Description
Delete()	Deletes this instance.
DeleteRule()	Deletes this IVxRule from the system.
Disable()	Disables this IVxRule.
Enable()	Enables this IVxRule.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetScript(char* script, int& size)	Get the script for this IVxRule.
GetTimeTables(VxCollection<IVxTimeTable**>& timeTableCollection)	A collection of all IVxTimeTable used by this IVxRule.
HaltScript()	Halt the script that this IVxRule is running, if any. No effect if the script is not currently running.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetName(char name[64])	Sets the `name` property.
SetScript(char* script)	Set the script to run when the IVxRule triggers. If this script is triggered by an IVxEvent, its properties will be available in the script `eventProperties` variable.
SetTimeTables(char** timeTableIds, int timeTableIdSize)	Sets the `timeTableIds` property.
SetTriggers(VxRuleTrigger** triggers, int triggerSize)	Sets the `triggers` property.

5.61 IVxSchedule

Description	Collection Filters
Represents a group of 0 or more resources associated with a set of time and/or event based IVxScheduleTrigger that, when *any* are active, cause an action to be performed (depending on the type of IVxScheduleTrigger). See Schedule Diagrams for further information.	**kAdvancedQuery** **kDataSourceId** Only IVxSchedule that apply to this IVxDataSource. **kModifiedSince** Only IVxSchedule that have been modified since the given DateTime will be returned. **kName**

Field Name	Type	Req	RW	Description
action	VxScheduleAction	Y	RW	Action to perform when this IVxSchedule is active.
id	String	Y	R	Unique IVxSchedule identifier.
name	String	N	RW	Friendly name.
useAllDataSources	Boolean	N	RW	True indicates that the IVxSchedule shall apply to all IVxDataSources regardless of what IVxDataSources are linked to the IVxSchedule. False indicates that only linked IVxDataSources shall be associated with this IVxSchedule.

Method	Description
AddScheduleTrigger(VxNewScheduleTrigger& newTrigger)	Add a new IVxScheduleTrigger; a maximum of 16 IVxScheduleTrigger are allowed per IVxSchedule.
Delete()	Deletes this instance.
DeleteSchedule()	Delete this IVxSchedule.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLinks(VxCollection<IVxDataSource**>& dataSourceCollection)	Returns a resources collection representing the resources linked with this IVxSchedule.
GetScheduleTriggers(VxCollection<IVxScheduleTrigger**>& triggerCollection)	Time ranges and events that cause this IVxSchedule to be active (or not).
Link(IVxDataSource& dataSource)	Link IVxDataSource resources with this IVxSchedule. Any links already present have no effect (considered successful). If any of the linked resources do not exist, `kInvalidValue` is returned. If any of the linked resources are deleted, their link here shall also be removed. Linked IVxDataSources have no effect if `useAllDataSources` is `True`.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetAction(VxScheduleAction::Value action)	Sets the `action` property.
SetName(char name[64])	Sets the `name` property.
SetUseAllDataSources(bool useAllDataSources)	Sets the `useAllDataSources` property.
UnLink(IVxDataSource& dataSource)	Remove IVxDataSource resource links from this IVxSchedule. For any links that do not exist, this has no effect (considered successful).

5.62 IVxScheduleTrigger

Description	Collection Filters
Represents a time range and an optional event that together act as a trigger to activate a IVxSchedule. The IVxScheduleTrigger is considered active when *all* of its time and event attributes indicate it should be active. A "timed" IVxScheduleTrigger will cause its IVxSchedule to record *all* associated resources; an "event" IVxScheduleTrigger will cause its IVxSchedule to record only the resource(s) that was the source of the event.	None.

Field Name	Type	Req	RW	Description
action	VxScheduleAction	N	RW	The action performed when the IVxScheduleTrigger activates.
eventProperties	VxKvObject	N	RW	If set, the `eventSituationType` only activates when it occurs with these properties. Ignored if no `eventSituationType` is set.
eventPropertySize	Integer	N	RW	The size of `eventProperties`.

Field Name	Type	Req	RW	Description
eventSituationType	String	N	RW	If set, this IVxScheduleTrigger is considered an "event" trigger, otherwise it is considered a "timed" trigger. If set, the IVxScheduleTrigger only activates when this type of event occurs. The IVxScheduleTrigger event state shall be considered active until the event becomes inactive. *See* `inactiveEventSituationType` *for information on when the IVxScheduleTrigger will deactivate.*
framerate	VxRecordFramerate	Y	RW	Framerate level to record at when this trigger is active. Note that if multiple triggers are active, the highest framerate level among them will be used.
id	String	Y	R	Unique IVxScheduleTrigger identifier.
inactiveEventSituationType	String	N	RW	If set, the IVxScheduleTrigger will no longer be considered active when this type of event occurs. If not set, the IVxScheduleTrigger will immediately be considered inactive after the triggering event occurs.
postTrigger	Integer	Y	RW	Amount of time, in seconds, to continue to consider the IVxScheduleTrigger active when it becomes inactive ("post alarm").
preTrigger	Integer	Y	RW	Amount of time, from `0` to `30` seconds, to consider the IVxScheduleTrigger active prior to when it becomes active ("pre alarm").
timeout	Integer	Y	RW	Amount of time, in seconds, to consider the IVxScheduleTrigger active immediately after it becomes active ("duration recording"). The IVxScheduleTrigger will become inactive when this time has elapsed (though the `postTrigger` may continue to keep it active at this point). No timeout is applied if the value is less than `1`.
timeTableId	String	N	RW	If set, the IVxScheduleTrigger may only be active during the time range(s) defined by this IVxTimeTable.

Method	Description
Delete()	Deletes this instance.
DeleteScheduleTrigger()	Delete this IVxScheduleTrigger.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetAction(VxScheduleAction::Value action)	Sets the `action` property.
SetEventProperties(VxKvObject* eventProperties, int eventPropertySize)	Sets the `eventProperties` property.
SetEventSituationType(char eventSituationType[128])	Sets the `eventSituationType` property.
SetFramerate(VxRecordingFramerate::Value framerate)	Sets the `framerate` property.

Method	Description
SetInactiveEventSituationType(char inactiveEventSituationType[128])	Sets the `inactiveEventSituationType` property.
SetPostTrigger(int postTrigger)	Sets the `postTrigger` property.
SetPreTrigger(int preTrigger)	Sets the `preTrigger` property.
SetTimeout(int timeout)	Sets the `timeout` property.
SetTimeTableId(char timeTableId[64])	Sets the `timeTableId` property.

5.63 IVxSituation

Description	Collection Filters
Represents a particular situation configuration that specifies how corresponding events shall be generated and handled when the situation is detected. A IVxSituation is uniquely identified by both its `type` and `sourceDeviceId`. See Situation Defaults for the factory default configuration of a IVxSituation.	**kAdvancedQuery** **kAudibleNotify** **kHasProperty** Only IVxSituation that have this property defined (see: Situations). **kLog** **kModifiedSince** Only IVxSituation that have been modified since the given DateTime will be returned. **kName** **kNotify** **kServicePropertyId** **kSeverity** **kSourceDeviceId** **kType**

Field Name	Type	Req	RW	Description
audibleLoopDelay	Integer	N	RW	Seconds to wait between audible notifications (see: `audiblePlayCount`.
audiblePlayCount	Integer	N	RW	Play audible notification this many times, separated by the `audibleLoopDelay`.
autoAcknowledgeTime	Integer	Y	RW	Number of seconds after which a generated event `ackState` will be set to `autoAcked`. If less than 0, a generated event must be manually acknowledged. If 0, a generated event must be set to `autoAcked` immediately (prior to any notifications being sent).
isAckNeeded	Boolean	Y	RW	If true, generated events shall have an initial `ackState` of `ackNeeded`. If false, generated events shall have an initial `ackState` of `noAckNeeded`.
name	String	N	RW	Friendly name.
notificationIds	String[]	Y	R	IDs of the IVxNotifications configured for this IVxSituation.
notificationIdSize	Integer	N	RW	The size of `notificationIds`.
servicePropertyId	String	N	R	The name of the key in the properties map corresponding to the service identifier for the source of this IVxSituation (e.g. "`data_source_id`", "`alarm_id`", etc). Omitted if not applicable (e.g. the source is directly from a IVxDevice).

Field Name	Type	Req	RW	Description
serviceType	String	N	R	Specifies the resource type of the service corresponding to the source of this IVxSituation, if any. The type is declared using the full case-sensitive name of the resource. Service resources include: `"AlarmInput"`, `"DataSource"`, `"DataStorage"`, `"Monitor"`, and `"RelayOutput"`. Omitted if not applicable (e.g. the source is directly from an IVxDevice).
severity	Integer	Y	RW	Severity of the generated event, from `1` (highest) to `10` (lowest).
shouldAudiblyNotify	Boolean	Y	RW	True specifies that a notification sound is to play on supporting clients when receiving a notification for IVxEvents corresponding to this IVxSituation.
shouldExpandBanner	Boolean	N	RW	True if clients should display an in-cell alert notification banner when receiving events of this type (default).
shouldLog	Boolean	Y	RW	If true, events generated from this IVxSituation shall be persisted as long as possible. If false, generated events shall immediately be discarded; unlogged events are hidden from clients (this supersedes all other situation configuration).
shouldNotify	Boolean	Y	RW	If true, events generated from this IVxSituation shall generate notifications that are sent to authorized clients, per the notification configuration of this IVxSituation, subscribed to the IVxSystem. Additionally, these notifications will be sent out whenever generated events have a change of `ackState`.
shouldPopupBanner	Boolean	N	RW	True if clients should display a popup notification banner when receiving events of this type (default).
snoozeIntervals	Integer[]	Y	RW	List of default snooze intervals, in seconds, for generated events. Note that these are default options and that they do *not* limit the amount of time the generated events may be snoozed for.
snoozeIntervalSize	Integer	N	RW	The size of `snoozeIntervals`.
sourceDeviceId	String	N	R	Together with `type`, is a unique IVxSituation identifier. This field acts an optional constraint on the source of events for this IVxSituation. If specified, any events matching the IVxSituation `type` MUST also match this `sourceDeviceId` in order for the IVxSituation to apply.
sourceDeviceTypes	VxDeviceType[]	N	R	A list of VxDeviceType that may be the source of events for this IVxSituation. *Note: This field is informational for aiding clients, especially in rule creation.*
sourceDeviceTypesSize	Integer	N	R	The size of `sourceDeviceTypes`.
type	String	Y	R	Together with `sourceDeviceId`, is a unique IVxSituation identifier. Possible types are specified by the Situations table, in the Type column.

Method	Description
AddNotification(VxNewNotification& newNotification, IVxNotification*& notificationItem)	Add a new IVxNotification to this IVxSituation. The server will return the new IVxNotification in response to a successful request.
Delete()	Deletes this instance.
DeleteAudioFile()	Deletes the custom audio file.
DeleteSituation()	Delete this IVxSituation from the system. This link shall only be available for custom situations (`id` of the form `external/<type>`).
GetAudioFile(char* endpoint, int& size)	Gets the custom audio file, if any.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLinks(VxCollection<IVxDataSource**>& dataSourceCollection)	Returns an IVxDataSource collection representing the IVxDataSource resources associated with this IVxSituation.
GetLinks(VxCollection<IVxDevice**>& deviceCollection)	Returns an IVxDevice collection representing the IVxDevice resources associated with this IVxSituation.
GetNotifications(VxCollection<IVxNotification**>& notificationCollection)	The collection of IVxNotification representing this situation's notification configuration. If this situation's `shouldNotify` is `true`, then event notifications will be sent to all authorized users/roles configured here.
Link(IVxDataSource& dataSource)	Associate the IVxDataSource with this IVxSituation. If the association is already present this will have no effect (considered successful). If the associated resource is deleted, this association shall also be removed.
Link(IVxDevice& device)	Associate the IVxDevice with this IVxSituation. If the association is already present this will have no effect (considered successful). If the associated resource is deleted, this association shall also be removed.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetAckNeeded(bool isAckNeeded)	Sets the `isAckNeeded` property.
SetAudibleLoopDelay(int audibleLoopDelay)	Sets the `audibleLoopDelay` property.
SetAudiblePlayCount(int audiblePlayCount)	Sets the `audiblePlayCount` property.
SetAudiblyNotify(bool shouldAudiblyNotify)	Sets the `shouldAudiblyNotify` property.
SetAudioFile(char* audioFilePath)	Sets the custom audio file used by clients for audible notifications. The maximum allowable size of the file is 5 MB.
SetAutoAcknowledgeTime(int autoAcknowledgeTime)	Sets the `autoAcknowledgeTime` property.
SetExpandBanner(bool shouldExpandBanner)	Sets the `shouldExpandBanner` property.
SetLog(bool shouldLog)	Sets the `shouldLog` property.
SetNotify(bool shouldNotify)	Sets the `shouldNotify` property.
SetName(char name[64])	Sets the `name` property.
SetPopupBanner(bool shouldPopupBanner)	Sets the `shouldPopupBanner` property.
SetSeverity(int severity)	Sets the `severity` property.

Method	Description
SetSnoozeIntervals(int* snoozeIntervals, int snoozeIntervalSize)	Sets the `snoozeIntervals` property.
UnLink(IVxDataSource& dataSource)	Remove the IVxDataSource association from this IVxSituation. If the association does not exist, this has no effect (considered successful).
UnLink(IVxDevice& device)	Remove the IVxDevice association from this IVxSituation. If the association does not exist, this has no effect (considered successful).

5.64 IVxSystem

Description	Collection Filters
Represents a VideoXpert system and allows the user to manage the system and devices.	None.

Field Name	Type	Req	RW	Description
graceLicenseExpirationTime	DateTime	N	R	The grace license expiration time, if applicable.
id	String	Y	R	Unique IVxSystem identifier.
name	String	N	RW	Friendly name of the entire system.

Method	Description
AcknowledgeAllEvents()	Acknowledge all IVxEvent (with `ackState` equal to `kAckNeeded`). IVxEvent with other `ackState` or not in this collection (e.g. due to permissions) will not be acknowledged. *Note: Requires VxPermissionId: `kHandleEvents`.*
AddAnalyticSession(VxNewAnalyticSession& newAnalyticSession)	Adds a new analytic session.
AddBookmark(VxNewBookmark& newBookmark)	Add a new IVxBookmark to the system.
AddDataObject(VxNewDataObject& newDataObject)	Add a new IVxDataObject to the system.
AddDevice(VxNewDevice& newDevice)	Add a new IVxDevice to the system.
AddDrawing(VxNewDrawing& newDrawing)	Add a new IVxDrawing to the system.
AddManualRecording(VxNewManualRecording& newManualRecording, IVxManualRecording*& manualRecordingItem)	DEPRECATED Add a new IVxManualRecording to the system. The server will return the new IVxManualRecording in response to a successful request.
AddMember(VxNewMember& newMember)	Adds a new member system to be aggregated.
AddRecording(VxNewRecording& newRecording, IVxRecording*& recordingItem)	Add a new IVxRecording to the system. The server will return the new IVxRecording in response to a successful request.
AddRole(const char* roleName)	Add a new IVxRole to the system.
AddRule(VxNewRule& newRule)	Add a new IVxRule to the system.
AddSchedule(VxNewSchedule& newSchedule)	Add a new IVxSchedule to the system.
AddSituation(VxNewSituation& newSituation)	Add a new IVxSituation, `external/<type>` only, to the system.
AddTag(VxNewTag& newTag)	Add a new IVxTag to the system.

Method	Description
AddTimeTable(VxNewTimeTable& newTimeTable)	Add a new IVxTimeTable to the system.
AddUser(VxNewUser& newUser)	Add a new IVxUser to the system. If the IVxUser `name` already exists (case-insensitive), `kConflict` will be returned.
CreateExport(VxNewExport& newExport, IVxExport*& exportItem)	Create a new IVxExport on the system. This will trigger the archival of all VxNewExportClip contained by the VxNewExport. Any VxNewExportClip that are not available will be omitted from the archived IVxExport data. Note that the archived IVxExport data may not be immediately available; use the IVxExport `percentComplete` and `status` fields to determine the status of the archival operation. A new export request is limited to a maximum of 24 hours of media.
CreateMonitor(VxNewMonitor& newMonitor)	Create a new IVxMonitor on the system.
CreateMonitorWall(const char* monitorWallName)	Create a new IVxMonitorWall on the system.
Delete()	Deletes this instance.
GetAccessPoints(VxCollection<IVxAccessPoint**>& accessPointCollection)	A collection of all IVxAccessPoint on this system.
GetAlarmInputs(VxCollection<IVxAlarmInput**>& alarmInputCollection)	A collection of all IVxAlarmInput on this system.
GetAnalyticSessions(VxCollection<IVxAnalyticSession**>& analyticSessionCollection)	A collection of all IVxAnalyticSession on this system.
GetBookmarkAutoUnlockTime(int& autoUnlockTime)	Get the current IVxBookmark automatic unlock time in days.
GetBookmarks(VxCollection<IVxBookmark**>& bookmarkCollection)	A collection of all IVxBookmark on this system.
GetClusterConfiguration(IVxConfiguration::Cluster*& clusterConfig)	Retrieve the cluster configuration.
GetCurrentUser(IVxUser*& user)	Get the user currently accessing the system.
GetDataObjects(VxCollection<IVxDataObject**>& dataObjectCollection)	A collection of all IVxDataObject on this system.
GetDataSources(VxCollection<IVxDataSource**>& dataSourceCollection)	A collection of all IVxDataSource on this system.
GetDataStorages(VxCollection<IVxDataStorage**>& dataStorageCollection)	A collection of all IVxDataStorage on this system.
GetDeviceAssignments(VxCollection<IVxDeviceAssignment**>& deviceAssignmentCollection)	A collection of all IVxDeviceAssignment on this system.
GetDevices(VxCollection<IVxDevice**>& deviceCollection)	A collection of all IVxDevice on this system.
GetDrawings(VxCollection<IVxDrawing**>& drawingCollection)	A collection of all IVxDrawing on this system.
GetDrivers(VxCollection<IVxDriver**>& driverCollection)	A collection of all IVxDriver on this system.
GetEvents(VxCollection<IVxEvent**>& eventCollection)	A collection of all IVxEvent on this system.
GetExports(VxCollection<IVxExport**>& exportCollection)	A collection of all IVxExport on this system.
GetHostDevice(IVxDevice*& hostDevice)	Gets the IVxDevice that hosts this system.

Method	Description
GetLicense(IVxLicense*& license)	Retrieve and/or manage IVxSystem license information.
GetManualRecordings(VxCollection<IVxManualRecording**>& manualRecordingCollection)	DEPRECATED A collection of all IVxManualRecording on this system.
GetMembers(VxCollection<IVxMember**>& memberCollection)	Gets the IVxMember systems that this IVxSystem is aggregating.
GetMonitors(VxCollection<IVxMonitor**>& monitorCollection)	A collection of all IVxMonitor on this system.
GetMonitorWalls(VxCollection<IVxMonitorWall**>& monitorWallCollection)	A collection of all IVxMonitorWall on this system.
GetRecordings(VxCollection<IVxRecording**>& recordingCollection)	A collection of all IVxRecording on this system.
GetRelayOutputs(VxCollection<IVxRelayOutput**>& relayOutputCollection)	A collection of all IVxRelayOutput on this system.
GetRoles(VxCollection<IVxRole**>& roleCollection)	A collection of all IVxRole on this system.
GetRules(VxCollection<IVxRule**>& ruleCollection)	A collection of all IVxRule on this system.
GetSchedules(VxCollection<IVxSchedule**>& scheduleCollection)	A collection of all IVxSchedule on this system.
GetSituations(VxCollection<IVxSituation**>& situationCollection)	A collection of all IVxSituation on this system.
GetSystemTime(char systemTime[64])	Gets the current system time (UTC).
GetTags(VxCollection<IVxTag**>& tagCollection)	A collection of all IVxTag on this system.
GetUsers(VxCollection<IVxUser**>& userCollection)	A collection of all IVxUser on this system.
InsertEvent(VxNewEvent& newEvent)	Generate a new IVxEvent based on the given data. If the event does not correspond to any IVxSituation it will be rejected.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetBookmarkAutoUnlockTime(int autoUnlockTime)	Automatically unlock any locked IVxBookmark that have a IVxBookmarkLock `endTime` older than this number of days. If `0`, no automatic unlock will be performed.
SetName(char name[64])	Sets the `name` property.
StartInternalNotifications(`VxInternalEventCallback` callback)	Start receiving internal event notifications sent by the VideoXpert SDK.
StartNotifications(`VxEventCallback` callback)	Start receiving system event notifications using the settings for the current user.
StartNotifications(`VxEventCallback` callback, VxCollection<IVxSituation**>& situationCollection, bool userNotification = false)	Start receiving system event notifications by situation type, regardless of user settings.
StopInternalNotifications()	Stop receiving all internal event notifications.
StopNotifications()	Stop receiving all system event notifications.

Method	Description
ValidateMember(bool& isValid, const char* host, int port, const char* username, const char* password)	Validates member administrator credentials.

5.65 IVxTag

Description	Collection Filters
Represents a non-hierarchical identifier typically associated with one or more resources. A IVxTag helps to describe a resource and allows it to be found more efficiently by browsing or searching.	**kAdvancedQuery** **kFolder** **kId** **kModifiedSince** Only IVxTag that have been modified since the given DateTime will be returned. **kName** Filter by IVxTag name. **kOwned** True to return only owned (private) IVxTag; false to return only non-owned (public) IVxTag. **kOwner** **kParentId** **kResourceId** Return only IVxTag with an association to the given resource ID. **kResourceType** Return only IVxTag with an association to the given resource type.

Field Name	Type	Req	RW	Description
id	String	Y	R	Unique IVxTag identifier.
isFolder	Boolean	N	R	True if this IVxTag is part of a folder-like hierarchy where it has a reference to a parent IVxTag or is a top-level folder IVxTag. A folder IVxTag CANNOT have an owner.
name	String	Y	RW	Unique together with `owner`. IVxTag friendly name. Commas are invalid characters for this field and MUST NOT be used.
owner	UPN	N	R	Unique together with `name`. If present, indicates that this resource is owned (private) by a IVxUser and this is their username. If not present, indicates that this resource is not owned (public). If this resource has an owner, only that owner and users with appropriate permissions will be able to read it. Not applicable if this is a folder IVxTag.
parentId	String	N	RW	A unique identifier for the parent IVxTag if this IVxTag is a folder IVxTag and is not a top-level folder IVxTag.

Method	Description
Delete()	Deletes this instance.
DeleteTag()	Deletes the IVxTag from the system.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetLinks(VxCollection<IVxDataSource**>& dataSourceCollection)	Returns an IVxDataSource collection representing the resources associated with this IVxTag.
GetLinks(VxCollection<IVxDevice**>& deviceCollection)	Returns an IVxDevice collection representing the resources associated with this IVxTag.
GetOwner(IVxUser*& user)	Retrieve the IVxUser that owns this IVxTag, if any.
GetParent(IVxTag*& tag)	Returns the IVxTag that is the parent IVxTag if this IVxTag has `isFolder` set to `true` and the IVxTag is not a top-level folder IVxTag.

Method	Description
Link(IVxDataSource& dataSource)	Associate an IVxDataSource with this IVxTag. If already associated this will have no effect (considered successful). If the associated resource is deleted, this association shall also be removed.
Link(IVxDevice& device)	Associate an IVxDevice with this IVxTag. If already associated this will have no effect (considered successful). If the associated resource is deleted, this association shall also be removed.
Merge(IVxTag& tag)	Merges this IVxTag into the given IVxTag. All links associated with this IVxTag will be re-associated with the given IVxTag; this IVxTag will then be deleted.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetName(char name[64])	Sets the `name` property.
SetParentId(char parentId[64])	Sets the `parentId` property.
UnLink(IVxDataSource& dataSource)	Remove the IVxDataSource association from this IVxTag. If the association does not exist, this has no effect (considered successful).
UnLink(IVxDevice& device)	Remove the IVxDevice association from this IVxTag. If the association does not exist, this has no effect (considered successful).

5.66 IVxTimeTable

Description	Collection Filters
Represents a named set of time ranges.	**kAdvancedQuery** **kId** **kModifiedSince** Only IVxTag that have been modified since the given DateTime will be returned. **kName** Filter by IVxTag name.

Field Name	Type	Req	RW	Description
endDate	DateTime	N	RW	The IVxTimeTable will be considered inactive after this date.
id	String	N	R	The unique IVxTimeTable identifier.
name	String	Y	RW	Friendly name
startDate	DateTime	N	RW	The IVxTimeTable will be considered inactive before this date.
weeklyTimeRanges	VxTimeRange[]	N	RW	Active time ranges for the IVxTimeTable. The list is sorted by `day` ascending (monday through sunday), then by `startTime` ascending, and finally by `endTime` ascending.
weeklyTimeRangeSize	Integer	N	R	The size of `weeklyTimeRanges`.

Method	Description
Delete()	Deletes this instance.
DeleteTimeTable()	Deletes the IVxTimeTable from the system.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.

Method	Description
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetEndDate(char endDate[64])	Sets the `endDate` property.
SetName(char name[64])	Sets the `name` property.
SetStartDate(char startDate[64])	Sets the `startDate` property.
SetWeeklyTimeRanges(VxTimeRange** weeklyTimeRanges, int weeklyTimeRangeSize)	Sets the `weeklyTimeRanges` property.

5.67 IVxUser

Description	Collection Filters
Represents information about a system user. See restricted users for additional information concerning special restrictions on internal system users.	**kAdvancedQuery** **kFirstName** **kLastName** **kModifiedSince** Only IVxUser that have been modified since the given DateTime will be returned. **kName**

Field Name	Type	Req	RW	Description
domain	String	N	RW	Network domain for this IVxUser. If not present, defaults to `LOCAL`.
email	EmailAddress	N	RW	Email address of this IVxUser.
employeeId	String	N	RW	Employee badge (or other) personnel identifier associated with this IVxUser.
firstName	String	N	RW	First name of IVxUser.
id	String	Y	R	Unique IVxUser identifier.
lastName	String	N	RW	Last name of IVxUser.
name	SASLString	Y	RW	The unique name, within the domain, of the user that this resource is representing. Though set and retrieved as case-sensitive, all server-side comparisons are performed as case-insensitive (adding a user, sorting, filtering, etc). This name will be the same as the username that is used to access the system.
note	String	N	RW	Supplemental information about this IVxUser.
passwordExpiration	DateTime	N	R	The time at which the user's password will expire. If there is no password expiration set, then this field will not be returned.
phoneNumbers	VxPhoneNumber[]	N	R	Telephone number(s) for this IVxUser. Maximum of 16 numbers.
phoneNumberSize	Integer	N	RW	The size of `phoneNumbers`.

Method	Description
AddToRole(IVxRole& role)	Adds this IVxUser to an IVxRole.
Delete()	Deletes this instance.
DeleteUser()	Remove the IVxUser from the system. This shall also remove the associated private IVxTags owned by this IVxUser. This action SHALL NOT be available for the IVxUser corresponding to the user making the request (you can't delete the IVxUser representing you).

Method	Description
GetAccountState(bool& isEnabled)	Gets the account state of the user.
GetDataObjects(VxCollection<IVxDataObject**>& dataObjectCollection)	All private IVxDataObject owned by this IVxUser and all public IVxDataObject. Other user's private IVxDataObject will *not* be returned regardless of client permissions.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetRoles(VxCollection<IVxRole**>& roleCollection)	All roles currently assigned to this user's authorization configuration.
GetTags(VxCollection<IVxTag**>& tagCollection)	Collection of IVxTag (private owned by this IVxUser and public; other users' private IVxTag will *not* be returned regardless of permissions).
GetUserAccount(IVxUserAccount*& userAccount)	Gets the IVxUserAccount configuration for this IVxUser.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
RemoveFromRole(IVxRole& role)	Removes this IVxUser from an IVxRole.
SetAccountState(bool isEnabled)	Sets the account state of the user. A disabled account will not be able to access the system.
SetDomain(char domain[64])	Sets the `domain` property.
SetEmail(char email[256])	Sets the `email` property.
SetEmployeeId(char employeeId[64])	Sets the `employeeId` property.
SetFirstName(char firstName[64])	Sets the `firstName` property.
SetLastName(char lastName[64])	Sets the `lastName` property.
SetNote(char note[1024])	Sets the `note` property.
SetPassword(char newPassword[64], bool mustChangePassword)	Submit a request for a password change. If successful, the client will need to use the new credentials for all future requests.
SetPhoneNumbers(VxPhoneNumber** phoneNumbers, int phoneNumberSize)	Sets the `phoneNumbers` property.

5.68 IVxUserAccount

Description	Collection Filters
Represents account information for a specific IVxUser.	None.

Field Name	Type	Req	RW	Description
canBypassLdap	Boolean	N	RW	If set to `true`, the IVxUser can login to the system using local credentials instead of using the LDAP authentication.
isEnabled	Boolean	N	RW	Indicates whether or not this IVxUserAccount is enabled. A disabled account will not be able to access the system unless the account is re-enabled. *Note: This field is typically read-only for the owner of this IVxUserAccount.*
isPasswordExpirationDisabled	Boolean	N	RW	If set to `true`, password expiration is disabled for this account; `false` will use the global password expiration setting for this account.

Method	Description
Delete()	Deletes this instance.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetCanBypassLdap(bool canBypassLdap)	Sets the `canBypassLdap` property.
SetIsEnabled(bool isEnabled)	Sets the `isEnabled` property.
SetIsPasswordExpirationDisabled(bool isPasswordExpirationDisabled)	Sets the `isPasswordExpirationDisabled` property.

5.69 IVxUserInfo

Description	Collection Filters
Represents general information about a IVxUser.	None.

Field Name	Type	Req	RW	Description
employeeId	String	N	R	Employee badge (or other) personnel identifier associated with this IVxUser.
firstName	String	N	R	First name of IVxUser.
lastName	String	N	R	Last name of IVxUser.
name	UPN	Y	R	Username of the IVxUser; used to access the VideoXpert system.
note	String	N	R	Supplemental information about this IVxUser.
phoneNumbers	VxPhoneNumber[]	N	R	Telephone number(s) for this IVxUser. Maximum of 16 numbers.
phoneNumberSize	Integer	N	RW	The size of `phoneNumbers`.

Method	Description
Delete()	Deletes this instance.

5.70 IVxVolume

Description	Collection Filters
Represents a storage volume.	**kId** **kModifiedSince**

Field Name	Type	Req	RW	Description
buffer	Float	N	RW	The percentage of the IVxVolume to keep free.
domain	String	N	RW	The domain for the IVxVolume network path. Note: Ignored if the `path` is not a UNC path.
id	String	Y	R	The unique identifier of the IVxVolume.
isBandwidthReserved	Boolean	N	RW	Indicates whether or not this is a NAS volume that shares bandwidth with data streams (bandwidth will be reserved for NAS operations).
isOnline	Boolean	N	R	Indicates whether or not the storage is online and available.
path	String	Y	RW	The fully qualified Windows directory path.

Field Name	Type	Req	RW	Description
username	String	N	RW	The username for access to the network path. Note: Ignored if the `path` is not a UNC path.
total	Integer	N	R	The total amount of storage present (in MB).
used	Integer	N	R	The amount of storage in use (in MB).

Method	Description
Delete()	Deletes this instance.
DeleteVolume()	Remove the IVxVolume from the system.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.
GetVolumeGroup(IVxVolumeGroup*& volumeGroup)	Retrieves the IVxVolumeGroup that this IVxVolume is in, if any
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetBuffer(float buffer)	Sets the `buffer` property.
SetDomain(char domain[64])	Sets the `domain` property.
SetIsBandwidthReserved(bool isBandwidthReserved)	Sets the `isBandwidthReserved` property.
SetPassword(char password[64])	Sets the password used to access to the network path. Note: Ignored if the `path` is not a UNC path.
SetPath(char path[512])	Sets the `path` property.
SetUsername(char username[64])	Sets the `username` property.

5.71 IVxVolumeGroup

Description	Collection Filters
Represents a group of IVxVolume that can be used together as a storage pool.	**kArchive** **kId** **kModifiedSince** **kName**

Field Name	Type	Req	RW	Description
id	String	Y	R	The unique identifier of the IVxVolumeGroup.
isArchiveGroup	Boolean	N	RW	If true, indicates that this IVxVolumeGroup is an `archive volume group`, which will act as storage for the oldest available data.
name	String	Y	RW	The friendly name of this IVxVolumeGroup.
volumes	IVxVolume[]	Y	R	The IVxVolume that constitute this IVxVolumeGroup.
volumesSize	Integer	Y	R	The size of `volumes`.

Method	Description
Delete()	Deletes this instance.
DeleteVolumeGroup()	Remove the IVxVolumeGroup from the system. This does not affect the IVxVolume within this IVxVolumeGroup nor the contents of those IVxVolumes.
GetLimits(VxLimits*& limits)	Gets the VxLimits related to this resource.

Method	**Description**
GetVolumeRelations(VxCollection<IVxResourceRel**>& resourceRelCollection)	Gets all possible IVxVolume relations for this IVxVolumeGroup (both linked and non-linked). Each linked IVxVolume shall be included in the IVxVolumeGroup while each non-linked IVxVolume shall be excluded. Note that a IVxVolume may only be linked to 1 IVxVolumeGroup.
Refresh()	Refreshes the member values for this object by retrieving its current information from the VideoXpert system.
SetIsArchiveGroup(bool isArchiveGroup)	Sets the `isArchiveGroup` property.
SetName(char name[64])	Sets the `name` property.

Permissions

Clients are granted authorization to access server resources via the roles and permissions that are assigned to them. The following rules apply:

1. A client has role (C_R) equal to the union of all granted roles.

   $C_R = R_1 \cup R_2 \cup R_3 \cup ...R_n$

2. A client has permission (C_P) equal to the union of all granted permissions.

   $C_P = P_1 \cup P_2 \cup P_3 \cup ...P_n$

3. A child's resource restrictions are a subset of its parent's restrictions. Any resources added to a permission resource restriction will be automatically added to all parent resource restrictions. Any resources removed from a permission resource restriction will be automatically removed from all child resource restrictions.

4. Setting `excludeRestricted` to `true` shall invert rule 3, it shall instead be: A child's resource restrictions are a superset of its parent's restrictions. Any resources added to a permission resource restriction will be automatically added to all child resource restrictions. Any resources removed from a permission resource restriction will be automatically removed from all parent resource restrictions.

5. The `excludeRestricted` setting shall match between a parent and child. Any change to the setting shall automatically be reflected in all parent and child permissions.

6. An unrestricted child permission shall always match its parent permission resource restrictions, if any.

7. When a resource restricted client creates a resource of the restricted type, the server shall add that resource as a new restriction to the client's permissions that have that resource restriction type.

The following table lists the complete set of permissions that a VideoXpert server MAY provide. The Permission ID identifies a specific permission. The Permission Level indicates that permission's nesting structure. Each Level is composed of permission names separated by forward slashes. The left-most name is the permission group name; this is *not* itself a permission, it is simply a category name for a group of permissions. The right-most permission name is the name of the specific permission. Names to the left of the permission level, not including the permission group name, indicate parent permissions that MUST first be granted. Any attempt at assigning a IVxRole a nested permission, when that IVxRole does not have all of its parent permissions, will be rejected and will return `kPermissionConflict`.

Each permission specifies the C(reate), R(ead), U(pdate), and D(elete) authorizations it grants upon select resources. Note that when the specification utilizes a lower-case character, this indicates that the authorization applies *only* to those resources owned by the user performing the request (e.g. the `kViewLiveMedia` permission provides a user CRUD access to their own IVxDataSession, but no other user's IVxDataSession).

Finally, each permission specifies resource restrictions that MAY be used for restricting the permission CRUD authorizations to a subset of resources. Only resources listed shall be available for restricting the permission. When a permission is restricted to a subset of resources, that permission's granted authorizations are only applicable to that subset of resources associated with the resource restrictions.

Resources without permissions listed below, or those not provided permissions by the server, shall deny all access except for those requests with the `Admin` permission.

6.1 Surveillance

Standard surveillance activities such as viewing live media, PTZ controls, initiating recording, etc.

Permission ID	Permission Level	CRUD	Restrictions
kViewLiveMedia	/surveil/video	`.R..` IVxDataInterface `crud` IVxDataSession `.R..` IVxDataSource `.R..` IVxDataSources `.R..` IVxDevice `.R..` IVxDevices `.R..` IVxMonitor `.R..` IVxMonitorCell `.R..` IVxMonitors `.R..` IVxMonitorWall `.R..` IVxMonitorWalls *IVxDataInterface provides live media*	IVxDataSource[1]
kUsePtzMode	/surveil/video/ptz	`.RU.` IVxPtzController `.R..` IVxPattern `.R..` IVxPatterns `.R..` IVxPreset `.R..` IVxPresets	IVxDataSource[2]
kLockPtzMode	/surveil/video/ptz/lock	`.RU.` IVxPtzLock	IVxDataSource[3]
kRecordMedia	/surveil/video/record	`crud` IVxManualRecording `.R..` IVxManualRecordings	IVxDataSource[4]
kLaunchSavedViewsRemotely	/surveil/video/launchtabs	`.R..` IVxDevice `.R..` IVxDevices `.RU.` IVxMonitor `.RU.` IVxMonitorCell `.R..` IVxMonitors `C...` IVxEvent (system/client_push)[5]	
kAccessAlarms	/surveil/alarms	`.R..` IVxAlarmInput `.R..` IVxAlarmInputs `.R..` IVxDevice `.R..` IVxDevices	
kAccessRelays	/surveil/relays	`.R..` IVxDevice `.R..` IVxDevices `.R..` IVxRelayOutput `.R..` IVxRelayOutputs `.RU.` IVxRelay (Trigger)	IVxRelayOutput[6]

1. Restricts IVxDataInterface to those hosted by the IVxDataSource; IVxDataSource to IVxDataSource; IVxDevice to those hosting the IVxDataSource.
2. Restricts IVxPtzController, IVxPattern, and IVxPreset to those available on this IVxDataSource.
3. Restricts IVxPtzLock to those available for IVxPtzController on this IVxDataSource.
4. Restricts IVxManualRecording to those recording the IVxDataSource media.
5. Only authorizes CREATE for IVxEvent with `situationType`: `system/client_push`.
6. Restricts IVxRelayOutput to IVxRelayOutput.

6.2 Investigation

Investigative activities such as viewing/managing recorded media, initiating and retrieving exports, etc.

Permission ID	Permission Level	CRUD	Restrictions
kViewRecordedMedia	/invest/clips	.R.. IVxBookmark .R.. IVxBookmarks CR.. IVxClip .R.. IVxClips .R.. IVxDataInterface crud IVxDataSession .R.. IVxDataSource .R.. IVxDataSources .R.. IVxDataStorage .R.. IVxDataStorages .R.. IVxDevice .R.. IVxDevices crud IVxPixelSearch *IVxDataInterface provides recorded media*	IVxDataSource[1]
kSystemBookmarks	/invest/clips/marks	CRUD IVxBookmark (unlocked) .R.. IVxBookmark (locked) .R.. IVxBookmarkLock	IVxDataSource[2]
kSystemLocks	/invest/clips/marks/locks	CRUD IVxBookmark (locked) .RU. IVxBookmarkLock	IVxDataSource[3]
kExportMediaClips	/invest/clips/priexports	crud IVxExport .R.. IVxExports	[4]
kManageExports	/invest/clips/exports	.RUD IVxExport .R.. IVxExports	[5]

1. Restricts IVxBookmark to those marking the IVxDataSource media; IVxClip to those from the IVxDataSource; IVxDataInterface to those hosted by the IVxDataSource; IVxDataSource to IVxDataSource; IVxDevice to those hosting the IVxDataSource.
2. Restricts IVxBookmark to those marking the IVxDataSource media.
3. Restricts IVxBookmark to those marking the IVxDataSource media.
4. Restrictions match parent (`kViewRecordedMedia`). Restricts IVxExport to those containing only IVxDataSource in the restricted set.
5. Restrictions match parent (`kViewRecordedMedia`). Restricts IVxExport to those containing only IVxDataSource in the restricted set.

6.3 Plug-Ins

Permissions related to plug-in capabilities such as mapping and external events.

Permission ID	Permission Level	CRUD	Restrictions
kUseMap	/plugin/usemap		
kViewMaps	/plugin/usemap/view	.R.. IVxDrawing .R.. IVxDrawings .R.. IVxMarker .R.. IVxMarkers	IVxDrawing[1]
kPlaceCamerasOnMap	/plugin/usemap/view/markers	CRUD IVxMarker CRUD IVxResourceLock	IVxDrawing[2]
kManageMapFiles	/plugin/usemap/view/config	CRUD IVxDrawing .RU. IVxDrawings CRUD IVxResourceLock	IVxDrawing[3]

1. Restricts IVxDrawing and IVxMarker to those on the IVxDrawing.
2. Restricts IVxMarker and IVxResourceLock to those on the IVxDrawing.
3. Restricts IVxDrawing.

6.4 Supervision and Reports

Supervisory activities aimed at "shift leaders".

Permission ID	Permission Level	CRUD	Restrictions
kDefinePtzPresets	/super/ptzpresets	.R.. IVxDataSource .R.. IVxDataSources .RU. IVxPtzController CRUD IVxPtzPattern .R.. IVxPtzPatterns CRUD IVxPtzPreset .R.. IVxPtzPresets	IVxDataSource[1]
kManageCameraTours	/super/tours		
kAccessUserWorkspaces	/super/viewworkspaces	.R.. IVxDataObject .R.. IVxDataObjects .R.. IVxUser .R.. IVxUsers	IVxUser[2]
kManageSystemWorkspaces	/super/workspaces	CRUD IVxDataObject .R.. IVxDataObjects .R.. IVxUser .R.. IVxUsers	IVxUser[3]
kAuditUserActivity	/super/viewusrevents	.R.. IVxEvent .R.. IVxEvents *Applies only to IVxEvent with a non-empty* `sourceUserName`	
kMultiviewQty	/super/multiview	.R.. IVxMultiview .R.. IVxUserInfo	

1. Restricts IVxDataSource to IVxDataSource; IVxPtzController, IVxPattern, and IVxPreset to those available on this IVxDataSource.
2. Restricts private IVxDataObject to those owned by this IVxUser (no restrictions on public IVxDataObject; restricts IVxUser to this IVxUser.
3. Restricts private IVxDataObject to those owned by this IVxUser (no restrictions on public IVxDataObject; restricts IVxUser to this IVxUser.

6.5 Event Management

Access and management of system events.

Permission ID	Permission Level	CRUD	Restrictions
kViewEventHistory	/events/viewsysevents	.R.. IVxEvent .R.. IVxEvents *Applies only to IVxEvent with a non-empty* `sourceUserName`	
kHandleEvents	/events/handle	.RU. IVxEvent .R.. IVxEvents	

Permission ID	Permission Level	CRUD	Restrictions
kConfigureEvents	/events/settings	.R.. IVxDataSource .R.. IVxDataSources .R.. IVxDevice .R.. IVxDevices CRUD IVxNotification .R.. IVxNotifications .R.. IVxPrivilege .R.. IVxPrivileges .R.. IVxRole .R.. IVxRoles CRUD IVxSituation .R.. IVxSituations .R.. IVxUser .R.. IVxUsers	

6.6 User Management

Management of user accounts, roles, and authorizations.

Permission ID	Permission Level	CRUD	Restrictions
kManageUserAccounts	/usr/accounts	.R.. IVxPrivilege .R.. IVxPrivileges .R.. IVxRole .R.. IVxRoles CRUD IVxUser .R.. IVxUsers CRUD IVxUser (Account state)	IVxUser[1]
kAssignRolesToUsers	/usr/accounts/assignroles	CRUD IVxUser (Roles)	
kResetUserPasswords	/usr/resetpw	.RU. IVxUser .R.. IVxUsers	IVxUser[2]
kManageRoles	/usr/roles	CRUD IVxPrivilege .R.. IVxPrivileges CRUD IVxRole .R.. IVxRoles	

1. Restricts IVxUser to IVxUser; user account to those owned by the IVxUser.
2. Restricts IVxUser.

6.7 Device Management

Access, licensing, and management of edge devices in the system (e.g. cameras and recorders).

Permission ID	Permission Level	CRUD	Restrictions
kManageSystemTags	/dev/tags	CRUD IVxTag (*public only*)[1] .R.. IVxTags	
kManageIO	/dev/io	.R.. IVxDevice .R.. IVxDevices CRUD IVxAlarmInput CRUD IVxAlarmInputs CRUD IVxRelayOutput CRUD IVxRelayOutputs CRUD IVxRelayTrigger	

Permission ID	Permission Level	CRUD	Restrictions
kManageDeviceLicenses	/dev/licenses	.R.. IVxDevice .R.. IVxDevices CRUD IVxLicense CRUD IVxLicenseFeature *Applies to device licenses only*	IVxDevice[2]
kUpdateDeviceSoftware	/dev/software		
kSetupEdgeDevices	/dev/settings	CRUD IVxConfiguration::Motion CRUD IVxConfiguration::Time CRUD IVxConfiguration::Storage .RU. IVxDataSource .R.. IVxDataSources CRUD IVxDevice .R.. IVxDevices CRUD IVxLog .R.. IVxLogs *Applies to non-manager devices only.*	IVxDevice[3]
kManageDisplayDevices	/dev/displays	CRUD IVxDevice .R.. IVxDevices CRUD IVxMonitor .RU. IVxMonitorCell .R.. IVxMonitors *Only* `kMonitor`, `kUi` *devices.*	
kConfigureMonitorWallDecoders	/dev/monitorwalls	CRUD IVxMonitor .RU. IVxMonitorCell .R.. IVxMonitors CRUD IVxMonitorWall .R.. IVxMonitorWalls	

1. Only authorizes on public IVxTag (those without an `owner`).
2. Restricts IVxDevice, IVxLicenseFeature, and IVxLicense to those of the host IVxDevice.
3. Restricts IVxConfiguration::Time, IVxDataSource, and IVxDevice to those of the host IVxDevice.

6.8 System Management

Access, licensing, and management of entire systems.

Permission ID	Permission Level	CRUD	Restrictions
kManageSystemLicenses	/sys/licenses	.R.. IVxDevice .R.. IVxDevices CRUD IVxLicense CRUD IVxLicenseFeature *Applies to system licenses only*	
kSetSystemLocaleOptions	/sys/locale	CRUD IVxConfiguration::Time	
kDefineSystemShortcuts	/sys/shortcuts		

Permission ID	Permission Level	CRUD	Restrictions
kConfigureRecording	/sys/recording	.R.. IVxDataSource .R.. IVxDataSources .RU. IVxDataStorage .R.. IVxDataStorages .R.. IVxDevice .R.. IVxDevices CRUD IVxDeviceAssignment .R.. IVxDeviceAssignments .R.. IVxDriver .R.. IVxDrivers CRUD IVxSchedule .R.. IVxSchedules CRUD IVxScheduleTrigger CRUD IVxTimeTable .R.. IVxTimeTables	
kViewSystemHealth	/sys/viewhealth	.R.. IVxGap .R.. IVxGaps	
kManageSystemServers	/sys/servers	.RU. IVxBookmarks CRUD IVxConfiguration::Cluster CRUD IVxConfiguration::Node CRUD IVxDevice .R.. IVxDevices .RU. IVxSystem *Applies to manager devices only*	
kManageMemberSystems	/sys/servers/members		

Situations

These types present the complete set of possible situations that can occur that, when detected, cause the generation of an event. Some situations specify additional properties in the table below that events MUST supply in their `properties` field when the value is available (if the value is unavailable, the property SHOULD be omitted). Some situations have a source service property id, indicated in bold; this is the ID of the service source for this type of situation. These bold properties shall be specified by their corresponding IVxSituation `servicePropertyId` attribute. Finally, each situation type specifies an identifier of the form `category/name` that categorizes and uniquely identifies it.

The properties `mod_changes` and `mod_original` are JSON objects in which the first-level key maps to the field of the same name in a resource and the type matches the corresponding field type. These properties indicate modified fields along with their new and original values.

7.1 Admin

A reconfiguration of the system has occurred. Typically administrative-level actions.

Type	Properties	Description
admin/alarm_input_modified	**alarm_input_id**: string alarm_input_name: string mod_changes: object mod_original: object	An existing IVxAlarmInput (`alarm_input_id`) named (`alarm_input_name`) has been modified (`mod_changes, mod_original`).
admin/bookmarks_modified	mod_changes: object mod_original: object	Bookmarks has been modified (`mod_changes, mod_original`).
admin/cert_expired		The TLS certificate for this server has expired.
admin/cert_modified		The TLS certificate for this server has been modified.
admin/clips_removed	data_source_id: string data_source_name: string end_time: DateTime start_time: DateTime	All clips have been removed from time (`start_time`) to time (`end_time`) for a IVxDataSource (`data_source_id`) named (`data_source_name`).
admin/config_smtp_modified	mod_changes: object mod_original: object	The SMTP configuration has been modified (`mod_changes, mod_original`).
admin/core_db_backup_failed		A Core database backup failed.
admin/core_db_backup_halted		A Core database backup was halted.
admin/core_db_backup_removed		A Core database backup was deleted from the system.
admin/core_db_backup_restore		A Core database restoration was requested.
admin/core_db_backup_started		
admin/critical_license_expiring	feature_name: string expiration: DateTime	A critical license for IVxLicenseFeature (`feature_name`) will be expiring on (`expiration`). If this happens, critical system functionality may be impacted.

Type	Properties	Description
admin/data_source_modified	**data_source_id**: string data_source_name: string mod_changes: object mod_original: object	An existing IVxDataSource (data_source_id) named (data_source_name) has been modified (mod_changes, mod_original).
admin/data_storage_modified	**data_storage_id**: string data_storage_name: string mod_changes: object mod_original: object	An existing IVxDataStorage (data_storage_id) named (data_storage_name) has been modified (mod_changes, mod_original).
admin/device_added	device_id: string device_ip: IP device_name: string	A new IVxDevice (device_id) at (device_ip) named (device_name) has been added to the system.
admin/device_assigned	**data_storage_id**: string data_storage_name: string device_assignment_id: string device_id: string device_ip: IP device_name: string	A IVxDevice (device_id) at (device_ip) named (device_name) has been assigned to a IVxDataStorage (data_storage_id) named (data_storage_name). The assignment is represented by IVxDeviceAssignment (device_assignment_id).
admin/device_commissioned	device_id: string device_ip: IP device_name: string	A new IVxDevice (device_id) at (device_ip) named (device_name) has been added to the list of commissioned IVxDevices.
admin/device_decommissioned	device_id: string device_ip: IP device_name: string	An existing IVxDevice (device_id) at (device_ip) named (device_name) has been removed from the list of commissioned IVxDevices.
admin/device_modified	device_id: string device_ip: IP device_name: string mod_changes: object mod_original: object	An existing IVxDevice (device_id) at (device_ip) named (device_name) has been modified (mod_changes, mod_original).
admin/device_rebooted	device_id: string device_ip: IP device_name: string	A IVxDevice (device_id) at (device_ip) named (device_name) has been soft-rebooted. This may via an API call or using a hardware switch.
admin/device_removed	device_id: string device_ip: IP device_name: string device_type: VxDeviceType	A IVxDevice (device_id) at (device_ip) of type (device_type) named (device_name) has been deleted from the system.
admin/device_reset	device_id: string device_ip: IP device_name: string	A IVxDevice (device_id) at (device_ip) named (device_name) has been reset to factory defaults. This may via an API call or using a hardware switch.
admin/device_unassigned	**data_storage_id**: string data_storage_name: string device_assignment_id: string device_id: string device_ip: IP device_name: string	A IVxDevice (device_id) at (device_ip) named (device_name) has been unassigned from a IVxDataStorage (data_storage_id) named (data_storage_name). The assignment was represented by IVxDeviceAssignment (device_assignment_id).
admin/drawing_added	drawing_id: string drawing_name: string	A new IVxDrawing (drawing_id) named (device_name) has been added to the system.
admin/drawing_image_retrieved	drawing_id: string drawing_name: string	A IVxDrawing (drawing_id) named (drawing_name) image has been retrieved from the system (GET on IVxDrawing:/pelco/rel/image).
admin/drawing_image_modified	drawing_id: string drawing_name: string	A IVxDrawing (drawing_id) named (drawing_name) binary image data has been modified.
admin/drawing_image_removed	drawing_id: string drawing_name: string	A IVxDrawing (drawing_id) named (drawing_name) binary image data has been deleted.
admin/drawing_marker_added	drawing_id: string drawing_name: string marker_id: string marker_name: string	A new IVxMarker (marker_id) named (marker_name) has been added to a IVxDrawing (drawing_id) named (drawing_name).

Type	Properties	Description
admin/drawing_marker_modified	drawing_id: string drawing_name: string marker_id: string marker_name: string mod_changes: object mod_original: object	A IVxMarker (marker_id) named (marker_name) has been modified (mod_changes, mod_original) on a IVxDrawing (drawing_id) named (drawing_name).
admin/drawing_marker_removed	drawing_id: string drawing_name: string marker_id: string marker_name: string	A IVxMarker (marker_id) named (marker_name) has been deleted from a IVxDrawing (drawing_id) named (drawing_name).
admin/drawing_modified	drawing_id: string drawing_name: string mod_changes: object mod_original: object	A IVxDrawing (drawing_id) named (drawing_name) has been modified (mod_changes, mod_original).
admin/drawing_removed	drawing_id: string drawing_name: string	A IVxDrawing (drawing_id) named (drawing_name) has been deleted from the system.
admin/drawings_modified	mod_changes: object mod_original: object	IVxDrawings has been modified (mod_changes, mod_original).
admin/export_path_modified		The storage location for IVxExport data has been modified.
admin/geolocation_modified	**resource_id**: string resource_name: string resource_type: string mod_changes: object mod_original: object	The geolocation on an existing resource (resource_id) named (resource_name) of type (resource_type) has been modified (mod_changes, mod_original).
admin/license_added	feature_name: string	A license for IVxLicenseFeature (feature_name) has been added/updated on the system.
admin/license_expired	feature_name: string	A license for IVxLicenseFeature (feature_name) has expired on the system.
admin/license_expiring	feature_name: string expiration: DateTime	A license for IVxLicenseFeature (feature_name) will be expiring on (expiration).
admin/license_failure	feature_name: string	A license for IVxLicenseFeature (feature_name) has failed to apply.
admin/log_added	log_id: string	A new IVxLog (log_id) has been added to the system.
admin/log_downloaded	log_id: string	A IVxLog (log_id) has been downloaded.
admin/log_removed	log_id: string	A IVxLog (log_id) has been deleted from the system.
admin/member_added	member_id: string member_name: string	A new member (member_id) named (member_name) has been added to system aggregation.
admin/member_modified	member_id: string member_name: string mod_changes: object mod_original: object	An aggregated member (member_id) named (member_name) has been modified (mod_changes, mod_original).
admin/member_removed	member_id: string member_name: string	An aggregated member (member_id) named (member_name) has been deleted from the system.
admin/monitor_added	**monitor_id**: string monitor_name: string	A new IVxMonitor (monitor_id) named (monitor_name) has been added to the system.
admin/monitor_cell_modified	mod_changes: object mod_original: object monitor_cell_index: integer **monitor_id**: string monitor_name: string new_data_source_id: string new_speed: float new_time: DateTime	A IVxMonitorCell (monitor_cell_index) on an existing IVxMonitor (monitor_id) named (monitor_name) has been modified (mod_changes, mod_original). The following IVxMonitorCell properties, if present, indicate the new values: (new_data_source_id), (new_speed), (new_time). 4.4 ERRATA To indicate a null value, the following properties MAY be set to an empty string: new_data_source_id, new_time.

Type	Properties	Description
admin/monitor_modified	mod_changes: object mod_original: object **monitor_id**: string monitor_name: string new_layout: CellLayout new_name: string new_number: integer	An existing IVxMonitor (monitor_id) named (monitor_name) has been modified (mod_changes, mod_original). The following IVxMonitor properties, if present, indicate the new values: (new_layout), (new_name), (new_number).
admin/monitor_removed	**monitor_id**: string monitor_name: string	A IVxMonitor (monitor_id) named (monitor_name) has been deleted from the system.
admin/monitorwall_added	monitorwall_id: string monitorwall_name: string	A new IVxMonitorWall (monitorwall_id) named (monitorwall_name) has been added to the system.
admin/monitorwall_modified	mod_changes: object mod_original: object monitorwall_id: string monitorwall_name: string	An existing IVxMonitorWall (monitorwall_id) named (monitorwall_name) has been modified (mod_changes, mod_original).
admin/monitorwall_removed	monitorwall_id: string monitorwall_name: string	A IVxMonitorWall (monitorwall_id) named (monitorwall_name) has been deleted from the system.
admin/new_export_keys		A new public/private key pair has been generated for Exports.
admin/notification_added	notification_id: string situation_name: string situation_source_device_id: string situation_type: string	A new IVxNotification (notification_id) has been added to a IVxSituation (situation_type and situation_source_device_id) named (situation_name).
admin/notification_added_role	notification_id: string role_id: string role_name: string situation_name: string situation_source_device_id: string situation_type: string	A IVxRole (role_id) named (role_name) has been added to an existing IVxNotification (notification_id) on a IVxSituation (situation_type and situation_source_device_id) named (situation_name).
admin/notification_removed	notification_id: string situation_name: string situation_source_device_id: string situation_type: string	A IVxNotification (notification_id) has been deleted from a IVxSituation (situation_type and situation_source_device_id) named (situation_name).
admin/notification_removed_role	notification_id: string role_id: string role_name: string situation_name: string situation_source_device_id: string situation_type: string	A IVxRole (role_id) named (role_name) has been removed from an existing IVxNotification (notification_id) on a IVxSituation (situation_type and situation_source_device_id) named (situation_name).
admin/privilege_added	permission: string privilege_id: string role_name: string	A new IVxPrivilege (privilege_id) has been added with permission (permission) on IVxRole (role_name) to the system.
admin/privilege_modified	mod_changes: object mod_original: object permission: string privilege_id: string role_name: string	A IVxPrivilege (privilege_id) with permission (permission) on IVxRole (role_name) has been modified (mod_changes, mod_original).
admin/privilege_removed	permission: string privilege_id: string role_name: string	An existing IVxPrivilege (privilege_id) with permission (permission) on IVxRole (role_name) has been deleted from the system.
admin/privilege_resource_added	permission: string privilege_id: string resource_id: string resource_name: string role_name: string	A resource (resource_id) named (resource_name) has been added to a IVxPrivilege (privilege_id) with permission (permission) on IVxRole (role_name).

Type	Properties	Description
admin/privilege_resource_removed	permission: string privilege_id: string resource_id: string resource_name: string role_name: string	A resource (resource_id) named (resource_name) has been removed from a IVxPrivilege (privilege_id) with permission (permission) on IVxRole (role_name).
admin/ptz_preset_added	**data_source_id**: string data_source_name: string ptz_preset_description: string	A new IVxPreset (ptz_preset_description) was added to a IVxDataSource (data_source_id) named (data_source_name).
admin/ptz_preset_modified	**data_source_id**: string data_source_name: string ptz_preset_description: string	An existing IVxPreset (ptz_preset_description) on IVxDataSource (data_source_id) named (data_source_name) was repositioned.
admin/ptz_preset_removed	**data_source_id**: string data_source_name: string ptz_preset_description: string	An existing IVxPreset (ptz_preset_description) was removed from a IVxDataSource (data_source_id) named (data_source_name).
admin/relay_output_modified	**relay_output_id**: string relay_output_name: string mod_changes: object mod_original: object	An existing IVxRelayOutput (relay_output_id) named (relay_output_name) has been modified (mod_changes, mod_original).
admin/role_added	role_id: string role_name: string	A new IVxRole (role_id) named (role_name) has been added to the system.
admin/role_modified	mod_changes: object mod_original: object role_id: string role_name: string	An existing IVxRole (role_id) named (role_name) has been modified (mod_changes, mod_original).
admin/role_removed	role_id: string role_name: string	A IVxRole (role_id) named (role_name) has been deleted from the system.
admin/rule_added	rule_id: string rule_name: string	A new IVxRule (rule_id) named (rule_name) has been added to the system.
admin/rule_modified	mod_changes: object mod_original: object rule_id: string rule_name: string	An existing IVxRule (rule_id) named (rule_name) has been modified (mod_changes, mod_original).
admin/rule_removed	rule_id: string rule_name: string	A IVxRule (rule_id) named (rule_name) has been deleted from the system.
admin/schedule_added	schedule_id: string schedule_name: string	A new IVxSchedule (schedule_id) named (schedule_name) has been added to the system.
admin/schedule_modified	mod_changes: object mod_original: object schedule_id: string schedule_name: string	An existing IVxSchedule (schedule_id) named (schedule_name) has been modified (mod_changes, mod_original).
admin/schedule_removed	schedule_id: string schedule_name: string	A IVxSchedule (schedule_id) named (schedule_name) has been deleted from the system.
admin/schedule_resource_linked	resource_id: string resource_name: string schedule_id: string schedule_name: string	A resource (resource_id) named (resource_name) has been linked to a IVxSchedule (schedule_id) named (schedule_name).
admin/schedule_resource_unlinked	resource_id: string resource_name: string schedule_id: string schedule_name: string	A resource (resource_id) named (resource_name) has been unlinked from a IVxSchedule (schedule_id) named (schedule_name).
admin/situation_added	situation_name: string situation_source_device_id: string situation_type: string	A new IVxSituation (situation_type and situation_source_device_id) named (situation_name) has been added to the system.

Type	Properties	Description
admin/situation_modified	mod_changes: object mod_original: object situation_name: string situation_source_device_id: string situation_type: string	An existing IVxSituation (situation_type and situation_source_device_id) named (situation_name) has been modified (mod_changes, mod_original).
admin/situation_removed	situation_name: string situation_source_device_id: string situation_type: string	A IVxSituation (situation_type and situation_source_device_id) named (situation_name) has been deleted from the system.
admin/situation_resource_linked	resource_id: string resource_name: string situation_name: string situation_source_device_id: string situation_type: string	A resource (resource_id) named (resource_name) has been linked to a IVxSituation (situation_type and situation_source_device_id) named (situation_name).
admin/situation_resource_unlinked	resource_id: string resource_name: string situation_name: string situation_source_device_id: string situation_type: string	A resource (resource_id) named (resource_name) has been unlinked from a IVxSituation (situation_type and situation_source_device_id) named (situation_name).
admin/ssh_enabled		SSH access has been enabled.
admin/ssh_disabled		SSH access has been disabled.
admin/timetable_added	timetable_id: string timetable_name: string	A new IVxTimeTable (timetable_id) named (timetable_name) has been added to the system.
admin/timetable_modified	mod_changes: object mod_original: object timetable_id: string timetable_name: string	An existing IVxTimeTable (timetable_id) named (timetable_name) has been modified (mod_changes, mod_original).
admin/timetable_removed	timetable_id: string timetable_name: string	A IVxTimeTable (timetable_id) named (timetable_name) has been deleted from the system.
admin/user_added	first_name: string last_name: string name: UPN	A new IVxUser (name) named (last_name, first_name) has been added to the system.
admin/user_modified	mod_changes: object mod_original: object name: UPN	A IVxUser (name) has been modified (mod_changes, mod_original).
admin/user_new_password	name: UPN	A IVxUser (name) has had their password changed.
admin/user_removed	first_name: string last_name: string name: UPN	An existing IVxUser (name) named (last_name, first_name) has been deleted from the system.
admin/user_role_added	name: UPN role_id: string role_name: string	A IVxRole (role_id) named (role_name) has been added to IVxUser (name).
admin/user_role_removed	name: UPN role_id: string role_name: string	A IVxRole (role_id) named (role_name) has been removed from IVxUser (name).
admin/vxs_db_rebuild		A VxStorage database rebuild was requested.
admin/vxs_db_restore		A VxStorage database restoration was requested.
admin/vxs_reconfigured		A VxStorage was reconfigured.

7.2 Analytic

A software analytic has generated a result (e.g. a license plate has been detected).

Type	Properties	Description
analytic/abandoned_object	**data_source_id**: string data_source_name: string	Abandoned object detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_abandoned_object	**data_source_id**: string data_source_name: string	Abandoned object no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/adaptive_motion	**data_source_id**: string data_source_name: string	Adaptive motion detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_adaptive_motion	**data_source_id**: string data_source_name: string	Adaptive motion no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/directional_motion	**data_source_id**: string data_source_name: string	Directional motion detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_directional_motion	**data_source_id**: string data_source_name: string	Directional motion no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/loitering	**data_source_id**: string data_source_name: string	Loitering detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_loitering	**data_source_id**: string data_source_name: string	Loitering no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/motion	**data_source_id**: string data_source_name: string	Motion detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_motion	**data_source_id**: string data_source_name: string	Motion no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/object_count	**data_source_id**: string data_source_name: string	Object count threshold detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_object_count	**data_source_id**: string data_source_name: string	Object count threshold no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/object_removal	**data_source_id**: string data_source_name: string	Object removal detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_object_removal	**data_source_id**: string data_source_name: string	Object removal no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/sabotage	**data_source_id**: string data_source_name: string	Sabotage detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_sabotage	**data_source_id**: string data_source_name: string	Sabotage no longer detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/stopped_vehicle	**data_source_id**: string data_source_name: string	Stopped vehicle detected on a IVxDataSource (data_source_id) named (data_source_name).
analytic/no_stopped_vehicle	**data_source_id**: string data_source_name: string	Stopped vehicle no longer detected on a IVxDataSource (data_source_id) named (data_source_name).

7.3 Client

A system client has encountered a situation that it will input into the system. Note that these types of events should be considered unreliable as they depend on the honesty and accuracy of the client to report them.

Type	Properties	Description
client/logoff		A user has logged out of the system.
client/logon		A user has logged onto the system.

Type	Properties	Description
client/saved_view_accepted	saved_view_name: string	A saved view named (saved_view_name) was accepted.
client/saved_view_received	saved_view_name: string	A saved view named (saved_view_name) was received.
client/saved_view_sent	saved_view_name: string	A saved view named (saved_view_name) was sent.
client/snooze	event_id: string event_time: DateTime situation_type: string snooze_time: Integer	A user has snoozed an event (event_id) of type (situation_type) that occurred at (event_time) for (snooze_time) seconds.
client/tab_added	tab_name: string	A new tab named (tab_name) has been added.
client/tab_modified	tab_name: string	An existing tab named (tab_name) has been modified.
client/tab_removed	tab_name: string	An existing tab named (tab_name) has been removed.
client/workspace_added	workspace_name: string	A new workspace named (workspace_name) has been added.
client/workspace_modified	workspace_name: string	An existing workspace named (workspace_name) has been modified.
client/workspace_removed	workspace_name: string	An existing workspace named (workspace_name) has been removed.

7.4 External

An external event has been input into the system. These are typically injected by third-party devices.

Type	Properties	Description
external/<company>/<event>	1 See description	An external IVxEvent input to the system where <company> and <event> are UTF-8 strings no greater than 64 characters each; forward slashes are not allowed. These strings describe the <company> that manufactured the device that was the source of the <event> (e.g.: external/pelco/swipe). [1] These are specified by the VxNewSituation when the external IVxSituation is created.

7.5 Hardware

A physical issue has been detected. Typically the result of a sensor reading.

Type	Properties	Description
hardware/cpu_load		The CPU load has exceeded normal operating specifications.
hardware/data_loss	**data_source_id**: string data_source_name: string	Data from an IVxDataSource (data_source_id) (data_source_name) has been lost (typically due to an incoming data rate on a storage system exceeding its maximum write rate).
hardware/disk_failure	disk_id: string	A disk drive (disk_id) failure has been detected.
hardware/fan_failure	fan_id: string	A fan (fan_id) failure has been detected.
hardware/input_loss	**data_source_id**: string data_source_name: string	A video/audio/data input feed on a IVxDataSource (data_source_id) named (data_source_name) has been unexpectedly lost.

Type	Properties	Description
hardware/input_restored	**data_source_id**: string data_source_name: string	A video/audio/data input feed on a IVxDataSource (`data_source_id`) named (`data_source_name`) has been restored.
hardware/link_speed	speed: number	A link speed change has been detected. The new link *<speed>* is reported.
hardware/memory_load		The memory load has exceeded normal operating specifications.
hardware/network_volume_full	volume_id: string	A network volume (`volume_id`) has filled beyond normal operating conditions.
hardware/network_volume_offline	volume_id: string	A network volume (`volume_id`) has gone offline.
hardware/network_volume_online	volume_id: string	A network volume (`volume_id`) has come online.
hardware/packet_loss	percent_loss: number	The percentage of packet loss (`percent_loss`) has exceeded normal operating conditions.
hardware/ps_failure		A power supply failure has been detected.
hardware/temperature	temperature: number	A temperature reading (`temperature`), in Celsius, has exceeded normal operating conditions.
hardware/ups_low	remaining_seconds: number	A UPS has reported a low power reserve. There is limited time remaining (`remaining_seconds`) before the UPS will run out of power.
hardware/volume_full		A disk volume has filled beyond normal operating conditions.
hardware/volume_offline	volume_id: string	A disk volume (`volume_id`) has gone offline.
hardware/volume_online	volume_id: string	A disk volume (`volume_id`) has come online.

7.6 System

A system operation. This is typically the result of an operator action. May also indicate issues with the normal operation of the system.

Type	Properties	Description
system/alarm_active	**alarm_id**: string alarm_index: Integer alarm_name: string	A hardware or software alarm input (`alarm_id`) at index (`alarm_index`) named (`alarm_name`) has gone active.
system/alarm_inactive	**alarm_id**: string alarm_index: Integer alarm_name: string	A hardware or software alarm input (`alarm_id`) at index (`alarm_index`) named (`alarm_name`) has gone inactive.
system/authentication_failure		A request has been denied due to invalid authentication credentials.
system/authorization_failure		A request has been denied due to insufficient authorization privileges.
system/bookmark_added	bookmark_id: string data_source_id: string data_source_name: string time: DateTime	A new IVxBookmark (`bookmark_id`) at (`time`) for IVxDataSource (`data_source_id`) named (`data_source_name`) has been added to the system.
system/bookmark_lock_enabled	bookmark_id: string bookmark_name: string data_source_id: string data_source_name: string start_time: DateTime end_time: DateTime	A IVxBookmarkLock on IVxBookmark (`bookmark_id`) named (`bookmark_name`) from (`start_time`) to (`end_time`) for IVxDataSource (`data_source_id`) named (`data_source_name`) has been enabled.

Type	Properties	Description
system/bookmark_lock_modified	bookmark_id: string bookmark_name: string data_source_id: string data_source_name: string mod_changes: object mod_original: object	A IVxBookmarkLock on IVxBookmark (bookmark_id) named (bookmark_name) for IVxDataSource (data_source_id) named (data_source_name) has been modified (mod_changes, mod_original). *Note: Does not apply to modification of the IVxBookmarkLock* isEnabled *attribute.*
system/bookmark_lock_disabled	bookmark_id: string bookmark_name: string data_source_id: string data_source_name: string start_time: DateTime end_time: DateTime	A IVxBookmarkLock on IVxBookmark (bookmark_id) named (bookmark_name) from (start_time) to (end_time) for IVxDataSource (data_source_id) named (data_source_name) has been disabled.
system/bookmark_modified	bookmark_id: string data_source_id: string data_source_name: string mod_changes: object mod_original: object time: DateTime	An existing IVxBookmark (bookmark_id) at (time) for IVxDataSource (data_source_id) named (data_source_name) has been modified (mod_changes, mod_original).
system/bookmark_removed	bookmark_id: string data_source_id: string data_source_name: string time: DateTime	An IVxBookmark (bookmark_id) at (time) for IVxDataSource (data_source_id) named (data_source_name) has been deleted from the system.
system/client_push	client_id: string request_id: string data: string data_type: integer	A client (client_id) is performing a request (request_id) that pushes custom (data) of type (data_type) to other clients *of the same type*. The client identifier MUST be unique to each type of client. Recommend using the Java package naming convention: com.<company>.<project>.<client_name>. The request identifier MUST be unique to the request. The data is a serialized data object (e.g.: JSON, XML, CSC, etc). The maximum allowable size of data is 1 MB. The server MUST NOT utilize this data in any way. The data_type is a client-specified integer that is opaque to the server and other client types.
system/client_push_ack	client_id: string request_id: string data: string data_type: integer	A client (client_id) is acknowledging the handling of a system/client_push request (request_id) and providing the result of that request (data) of type (data_type). The client identifier MUST be unique to each type of client. Recommend using the Java package naming convention: com.<company>.<project>.<client_name>. The request identifier MUST be unique to the request. The data is a serialized data object (e.g.: JSON, XML, CSC, etc). The server MUST NOT utilize this data in any way. The data_type is a client-specified integer that is opaque to the server and other client types.
system/clip_added	data_source_id: string data_source_name: string **data_storage_id**: string data_storage_name: string end_time: DateTime start_time: DateTime	A new IVxClip has finished being created for IVxDataSource (data_source_id) named (data_source_name) on IVxDataStorage (data_storage_id) named (data_storage_name) from (start_time) to (end_time).
system/clip_failed	data_source_id: string data_source_name: string **data_storage_id**: string data_storage_name: string end_time: DateTime start_time: DateTime	An IVxClip has failed to be created for IVxDataSource (data_source_id) named (data_source_name) on IVxDataStorage (data_storage_id) named (data_storage_name) from (start_time) to (end_time).
system/data_source_offline	data_source_id: string data_source_name: string	A system IVxDataSource (data_source_id) named (data_source_name) has gone offline.

Type	Properties	Description
system/data_source_online	data_source_id: string data_source_name: string	A system IVxDataSource (data_source_id) named (data_source_name) has come online.
system/device_offline		A system IVxDevice has gone offline.
system/device_online		A system IVxDevice has come online.
system/device_status_initialized	device_id: string device_name: string	A system IVxDevice (device_id) named (device_name) has finished initializing (its status was initializing and is no longer).
system/device_status_unauthenticated	device_id: string device_name: string	The status of a system IVxDevice (device_id) named (device_name) is now unauthenticated.
system/export_deleted	export_id: string export_name: string	An IVxExport (export_id) named (export_name) has been deleted.
system/export_downloaded	export_id: string export_name: string	An IVxExport (export_id) named (export_name) has been downloaded.
system/export_failure	export_id: string export_name: string	An IVxExport (export_id) named (export_name) has failed to complete successfully.
system/export_modified	export_id: string export_name: string mod_changes: object mod_original: object	An existing IVxExport (export_id) named (export_name) has been modified (mod_changes, mod_original).
system/export_restored	export_id: string export_name: string	A trashed IVxExport (export_id) named (export_name) has been restored.
system/export_started	export_id: string export_name: string	An IVxExport (export_id) named (export_name) has been started.
system/export_success	export_id: string export_name: string	An IVxExport (export_id) named (export_name) has completed successfully.
system/export_trashed	export_id: string export_name: string	An IVxExport (export_id) named (export_name) has been trashed.
system/failover_completed	**data_storage_id**: string data_storage_name: string device_id: string device_name: string	A VxStorage device (device_id) named (device_name) providing (data_storage_id) named (data_storage_name) has resumed normal operation; failover to its monitoring VxStorage has ceased.
system/failover_started	**data_storage_id**: string data_storage_name: string device_id: string device_name: string	A VxStorage device (device_id) named (device_name) providing (data_storage_id) named (data_storage_name) been failed over to its monitoring VxStorage.
system/fault	fault_code: string fault_data: string fault_message: string software_id: string	An unexpected fault (fault_code) has occurred in a software component (software_id) (using the Java package naming convention: com.<company>.<project>.<software>). A short English description of the fault (fault_message) and additional fault data (fault_data), not exceeding 1 MB, may be available. [couleur=bcbox!80, arrondi=0.2, logo=, couleurBord=bcbox!200] ExceptionThis situation SHALL only used when no other type would better express the situation. Additionally, this type is intended to be used only when the exact nature of the situation is not useful to express (if it is, a new situation type should be created).
system/manual_recording_added	data_source_id: string data_source_name: string manual_recording_id: string	A new IVxManualRecording (manual_recording_id) for IVxDataSource (data_source_id) named (data_source_name) has been added.
system/manual_recording_removed	data_source_id: string data_source_name: string manual_recording_id: string	An IVxManualRecording (manual_recording_id) for IVxDataSource (data_source_id) named (data_source_name) has been deleted.

Type	Properties	Description
system/member_offline	member_id: string member_name: string	An aggregated member (member_id) named (member_name) has gone offline.
system/member_online	member_id: string member_name: string	An aggregated member (member_id) named (member_name) has come online.
system/ptz_lock	**data_source_id**: string data_source_name: string	An IVxPtzLock has been applied to a IVxDataSource (data_source_id) named (data_source_name).
system/ptz_pattern_triggered	**data_source_id**: string data_source_name: string ptzpattern_name: string	An IVxPattern named (ptzpattern_name) has been triggered on a IVxDataSource (data_source_id) named (data_source_name).
system/ptz_preset_triggered	**data_source_id**: string data_source_name: string ptzpreset_name: string	An IVxPreset named (ptzpreset_name) has been triggered on a IVxDataSource (data_source_id) named (data_source_name).
system/ptz_unlock	**data_source_id**: string data_source_name: string	An IVxPtzLock has been deleted from a IVxDataSource (data_source_id) named (data_source_name).
system/qlog_downloaded		A quick log has been downloaded.
system/qreport_downloaded		A quick report has been downloaded.
system/recording_failure_bandwidth	data_source_id: string data_source_name: string **data_storage_id**: string data_storage_name: string	A recording has been denied for IVxDataSource (data_source_id) named (data_source_name) on IVxDataStorage (data_storage_id) named (data_storage_name) due to bandwidth constraints.
system/relay_active	**relay_id**: string relay_name: string	An IVxRelayOutput (relay_id) named (relay_name) has gone active.
system/relay_inactive	**relay_id**: string relay_name: string	An IVxRelayOutput (relay_id) named (relay_name) has gone inactive.
system/retention_low	retention_minutes: number	The retention (retention_minutes) on a IVxDevice is failing to meet expectations.
system/script_failure	fault_code: string fault_data: string fault_message: string rule_id: string rule_name: string	An IVxRule (rule_id) named (rule_name) has failed to run its script successfully; an unexpected fault (fault_code) occurred. A short English description of the fault (fault_message) and additional fault data (fault_data), not exceeding 1 MB, may be available.
system/script_started	rule_id: string rule_name: string	An IVxRule (rule_id) named (rule_name) has started its script.
system/script_success	rule_id: string rule_name: string	An IVxRule (rule_id) named (rule_name) has run its script successfully.
system/stream_loss	**data_source_id**: string data_source_name: string	A network stream from a IVxDataSource (data_source_id) named (data_source_name) has been unexpectedly lost.
system/stream_restored	**data_source_id**: string data_source_name: string	A network stream from a IVxDataSource (data_source_id) named (data_source_name) has been restored.
system/stream_view_denied	**data_source_id**: string data_source_name: string	Stream viewing has been denied on a IVxDataSource (data_source_id) named (data_source_name) due to resource constraints.
system/stream_view_started	**data_source_id**: string data_source_name: string time: DateTime	Stream viewing has been started on a IVxDataSource (data_source_id) named (data_source_name) at time (time). If this is a live stream, (time) will be omitted.
system/stream_view_stopped	**data_source_id**: string data_source_name: string	Stream viewing has been stopped on a IVxDataSource (data_source_id) named (data_source_name).
system/stream_view_timeout	**data_source_id**: string data_source_name: string	Stream viewing has timed out on a IVxDataSource (data_source_id) named (data_source_name).

Type	Properties	Description
system/tag_added	tag_id: string tag_name: string	A new IVxTag (tag_id) named (tag_name) has been added to the system.
system/tag_linked	tag_id: string tag_name: string resource_id: string resource_name: string resource_type: string	An IVxTag (tag_id) named (tag_name) has been linked to a resource (resource_id) named (resource_name) of type (resource_type).
system/tag_merged	tag1_id: string tag1_name: string tag2_id: string tag2_name: string	An IVxTag (tag1_id) named (tag1_name) has been merged into another IVxTag (tag2_id) named (tag2_name).
system/tag_modified	mod_changes: object mod_original: object tag_id: string tag_name: string	An existing IVxTag (tag_id) named (tag_name) has been modified (mod_changes, mod_original).
system/tag_removed	tag_id: string tag_name: string	An IVxTag (tag_id) named (tag_name) has been deleted from the system.
system/tag_unlinked	tag_id: string tag_name: string resource_id: string resource_name: string resource_type: string	An IVxTag (tag_id) named (tag_name) has been unlinked from a resource (resource_id) named (resource_name) of type (resource_type).

Technical Addendum

Additional detailed technical specifications and figures related to the VideoXpert SDK.

8.1 Data Rate Units

Data transfer rate is the average number of bits passing over a network (bitrate). The measurement of data transfer rate is reported in multiples of unit bits per second (bits/s). The following data rate units are utilized in this document:

`kbps` – Kilobits per second:

- 1,000 bits per second
- 125 bytes per second

`mbps` – Megabits per second:

- 1,000,000 bits per second
- 1,000 kilobits per second
- 125,000 bytes per second

`gbps` – Gigabits per second:

- 1,000,000,000 bits per second
- 1,000 megabits per second
- 125,000,000 bytes per second

8.2 Collections

Many of the methods in the VideoXpert SDK utilize the VxCollection data type in order to return a set of resources. These methods require the client to allocate the `collection` array based upon the size of the collection that will be returned by the system. In order to determine the size of the collection, the method must first be called with a reference to a VxCollection. The SDK will set the `collectionSize` field to the size of the `collection` that will be returned by the system. At this point, the client can initialize the `collection` array and call the method again. The SDK will then populate the `collection` with the resources returned from the system. It is important to note that the client is also responsible for deleting the resources within the `collection` as well as the `collection` array itself.

The VxResult values returned from these methods are also important. Unless there were no resources found, the SDK will return `kInsufficientSize` when obtaining the `collectionSize`. Meaning the size of the `collection` array is smaller than the `collectionSize`. If no resources were found, the SDK will return `kOK` since the size of the `collection` array and the `collectionSize` are the same (`0`). There is also the possibility that an error occurred, which will return the appropriate VxResult value. Once the client has allocated the `collection` array the method should return `kOK`.

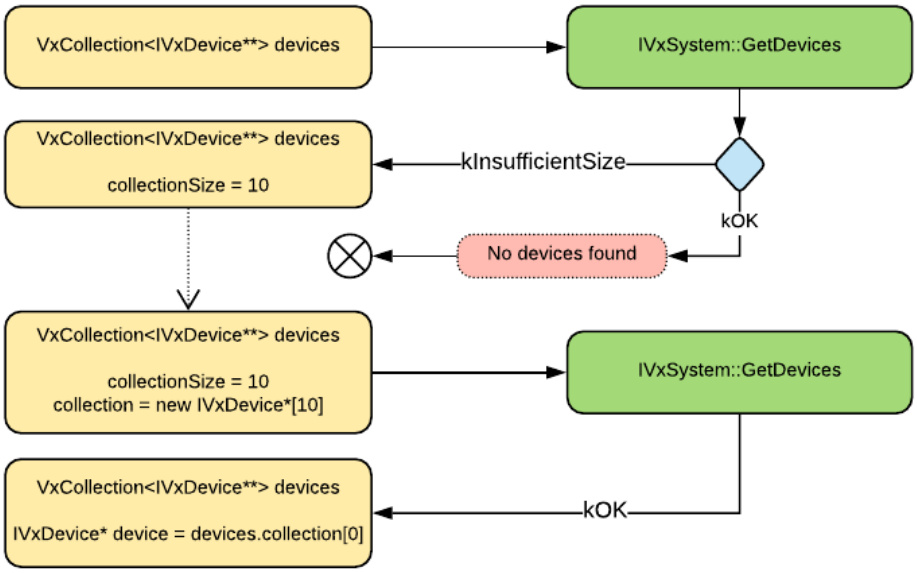

Figure 8.1: Collection Retrieval Process

8.3 Event State Transitions

An IVxEvent can transition through several different acknowledgement states (see: VxAckState). The figure below depicts these valid state transitions. "Ack" refers to the `Acknowledge` method while "Silence" refers to the `Silence` method. The two automated state transitions, "«wakeup»" and "«auto-ack»", refer to automatic transitions that occur on the server after a configured period of time has elapsed. The initial acknowledgement state of an IVxEvent is determined by its IVxSituation configuration.

Note that an IVxEvent configured to notify (`shouldNotify` attribute set to `True`) SHALL generate a notification for *every* state transition.

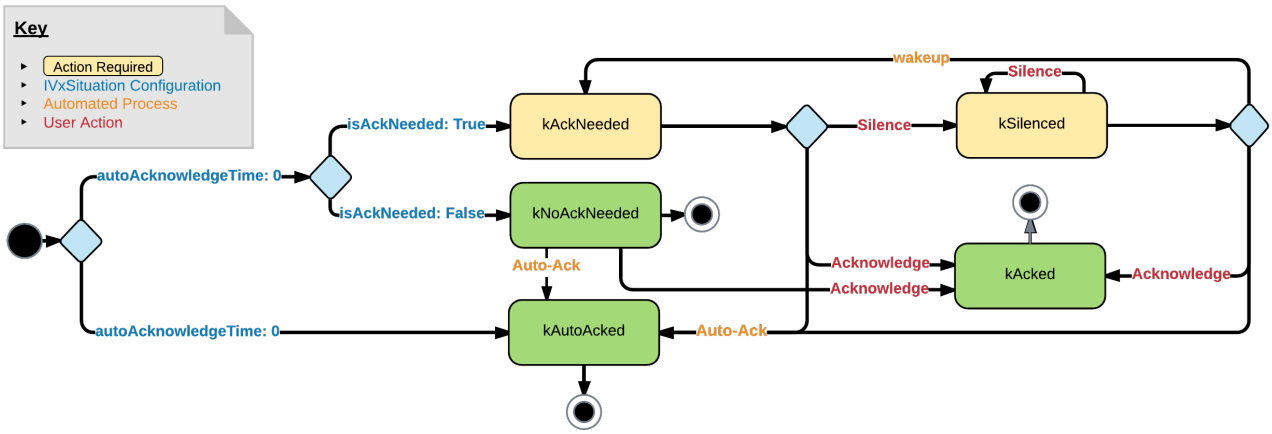

Figure 8.2: Event State Transitions

8.4 Event Workflow Examples

The figures below depict several example IVxEvent workflows for illustrative purposes.

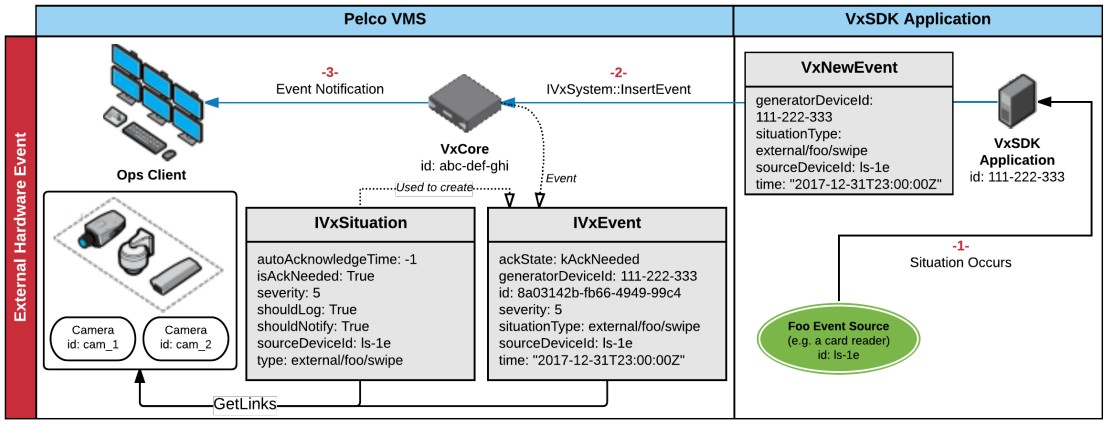

Figure 8.3: External Hardware Event

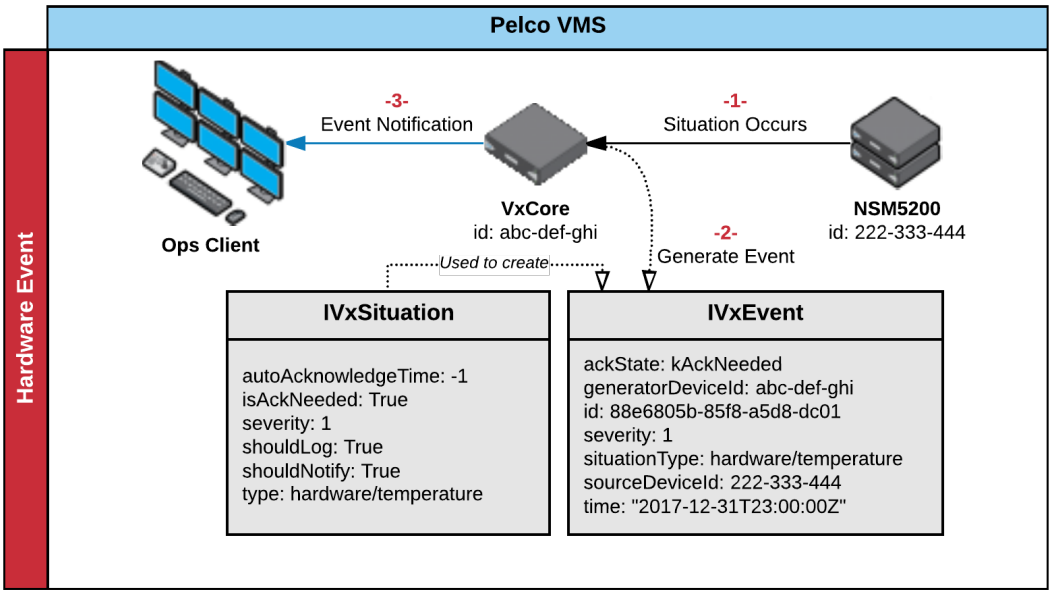

Figure 8.4: Internal Hardware Event

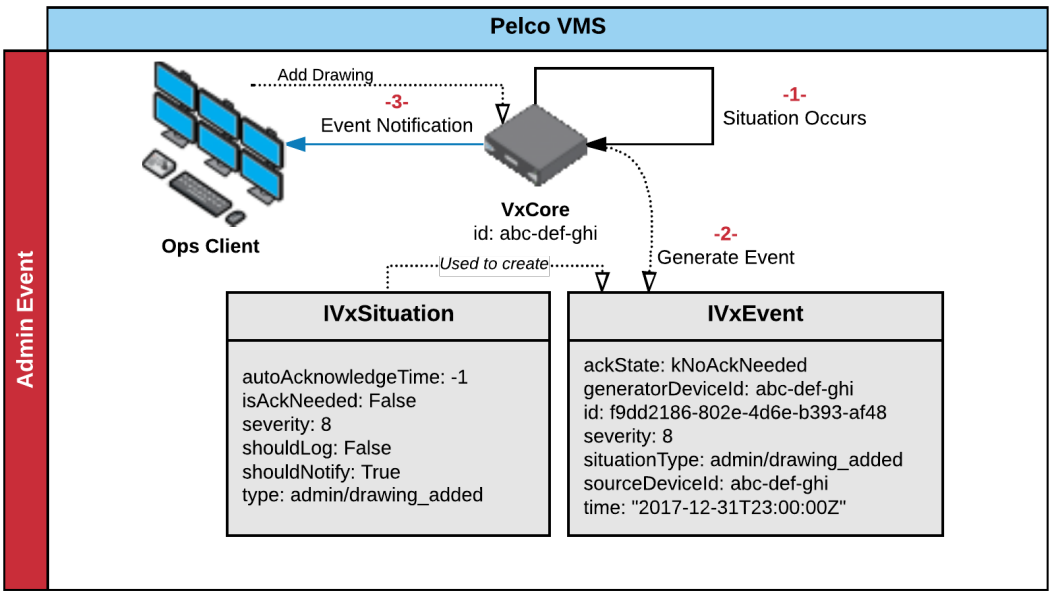

Figure 8.5: Internal Admin Event

8.5 Media Retrieval

Media data retrieval is accomplished using the RTSP control protocol [RFC 2326] or MJPEG (in which a IVxDataSession acts as the control protocol). Both live and recorded media is accessed using the same method; clients do not need to know where the media is located to retrieve it. See the Media Retrieval section for further details.

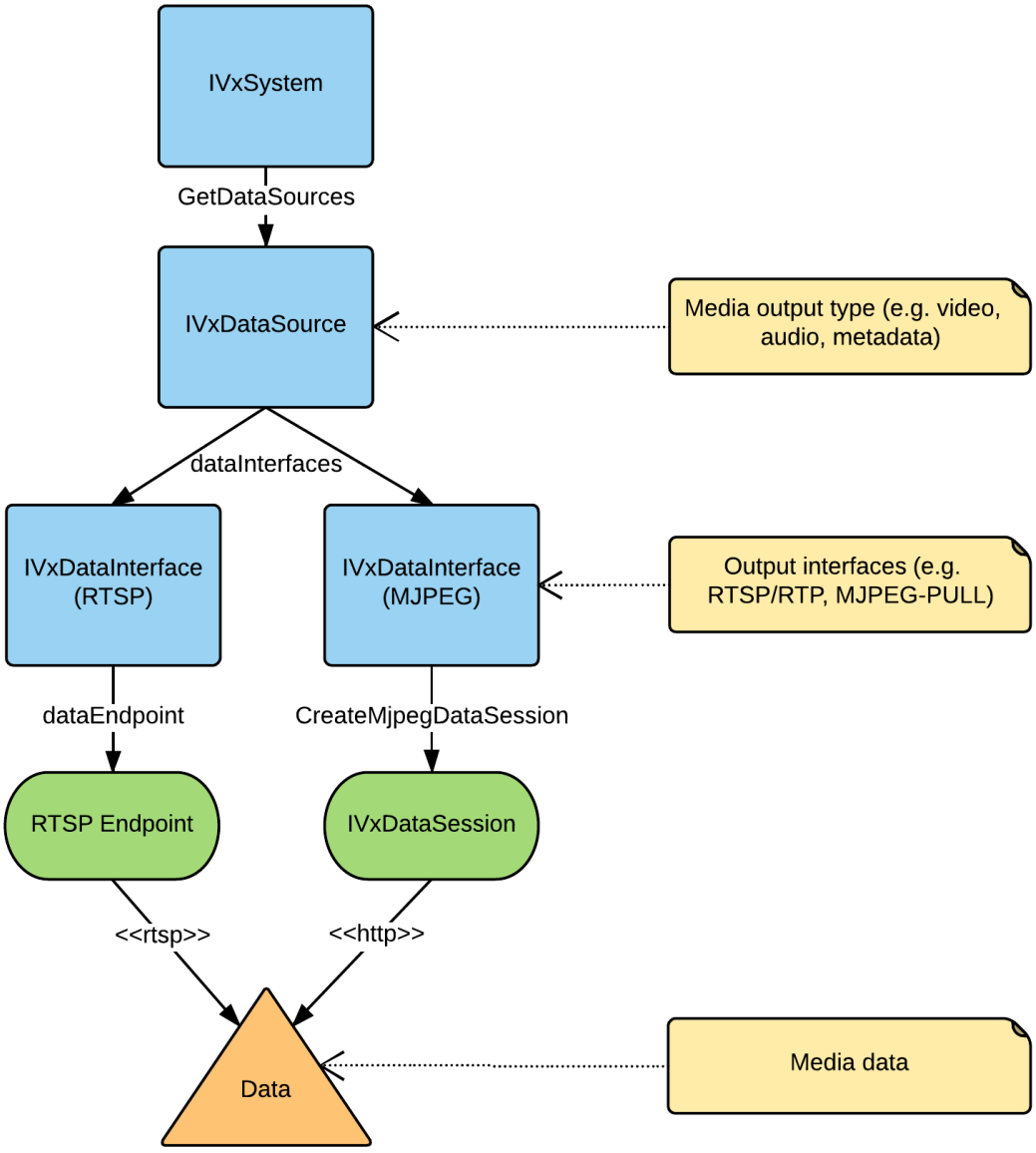

Figure 8.6: Media Retrieval

8.6 RTSP Streaming Sequence

Initiating and controlling an RTSP stream from the server is accomplished using the RTSP control protocol [RFC 2326]. See the RTSP Streaming section for further details.

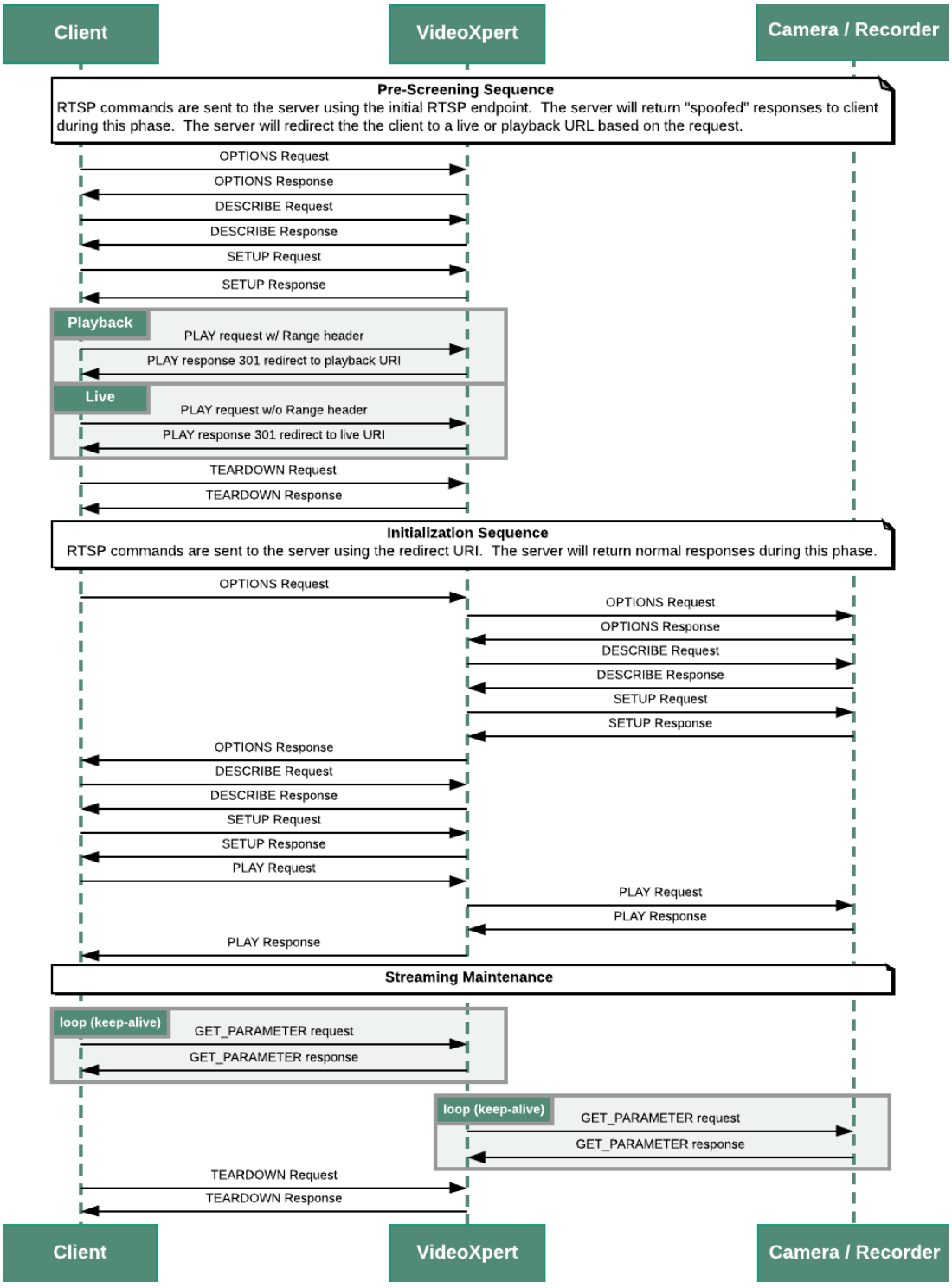

Figure 8.7: RTSP Streaming Sequence

8.7 MJPEG-Pull Streaming Sequence

Initiating and controlling an MJPEG-Pull stream from the server... See the MJPEG Streaming section for further details.

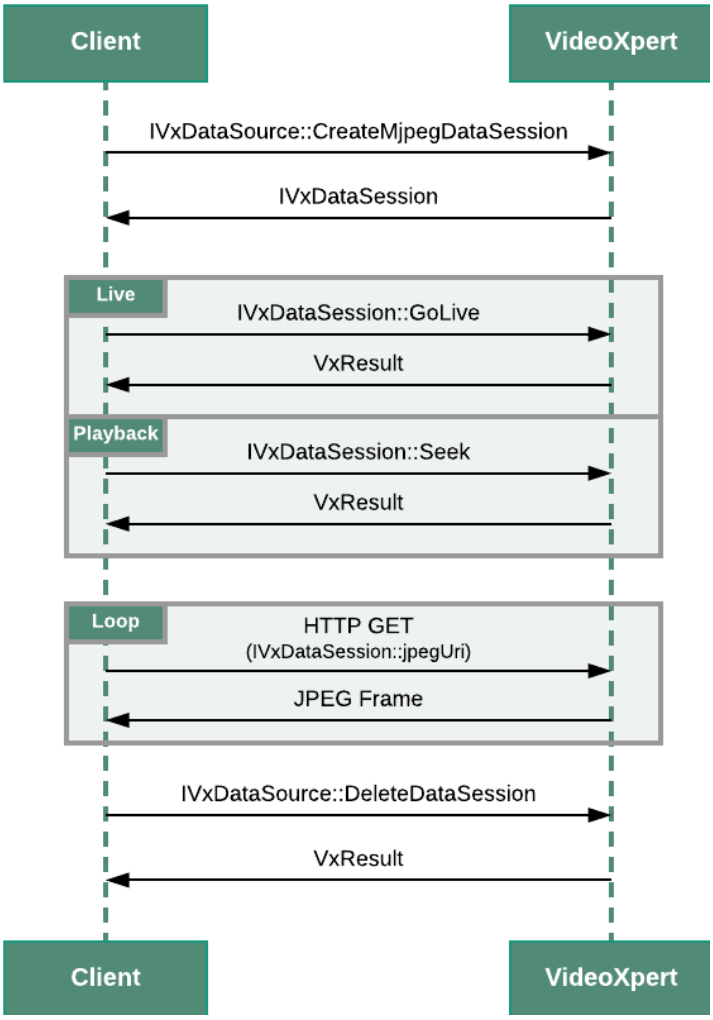

Figure 8.8: MJPEG-Pull Streaming Sequence

8.8 Restricted Users

VideoXpert systems utilize internal users–users that are hardcoded into the system–for special purposes. The following names are for internal users and are restricted from being used as a normal IVxUser `name`:

- **`admin`**: Used by Vx for administrative use.
- **`internal`**: Used internally for Vx Core to Core communication.
- **`rule_engine`**: Used by Vx for rules engine use.
- **`snmp`**: Used internally for SNMP communication.

Note that internal users have the following restrictions:

- They can not be edited.
- They can not be disabled.
- Their role configuration can not be changed.
- They can not be deleted.
- Their password can not be changed.

With the following special case:

1. The `admin` permission may change any account password.

8.9 Schedule Diagrams

A IVxSchedule represents a group of 0 or more resources associated with a set of time and/or event based IVxScheduleTrigger that, when *any* are active, cause the scheduled action to be performed.

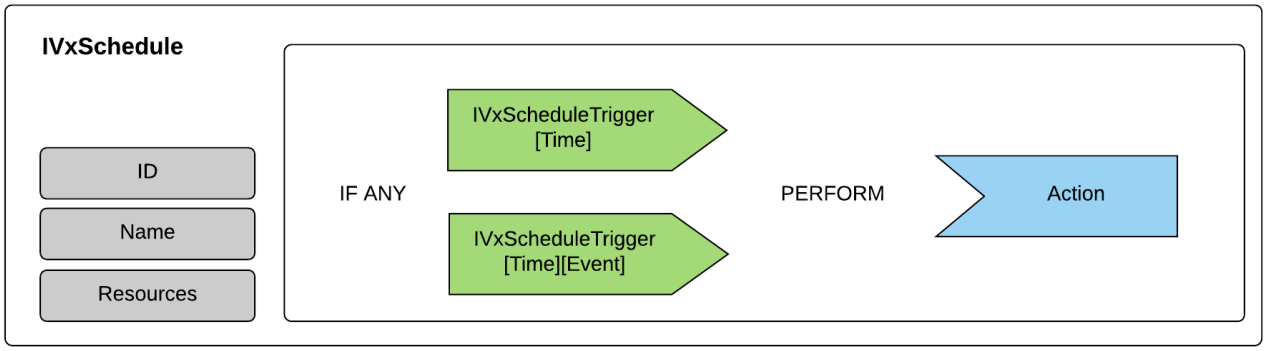

Figure 8.9: Schedule Overview

A IVxScheduleTrigger represents a time range and an optional event that together act as a trigger to activate a IVxSchedule and cause it to perform its action.

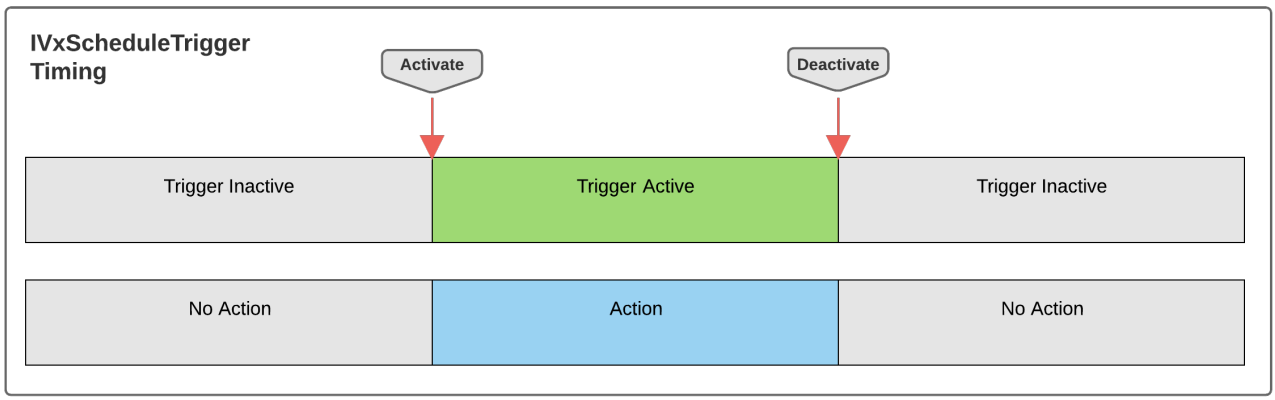

Figure 8.10: ScheduleTrigger Timing

The activation and deactivation of a IVxScheduleTrigger MAY be modified one or more of the following optional attributes: `preTrigger`, `timeout`, and `postTrigger`. The pre/post trigger attributes cause the IVxScheduleTrigger to be considered active before/after it actually is active (respectively). The timeout attribute causes the IVxScheduleTrigger to remain active (regardless of time and/or event activity) until the timeout period has elapsed, at which point the timeout causes the IVxScheduleTrigger to deactivate.

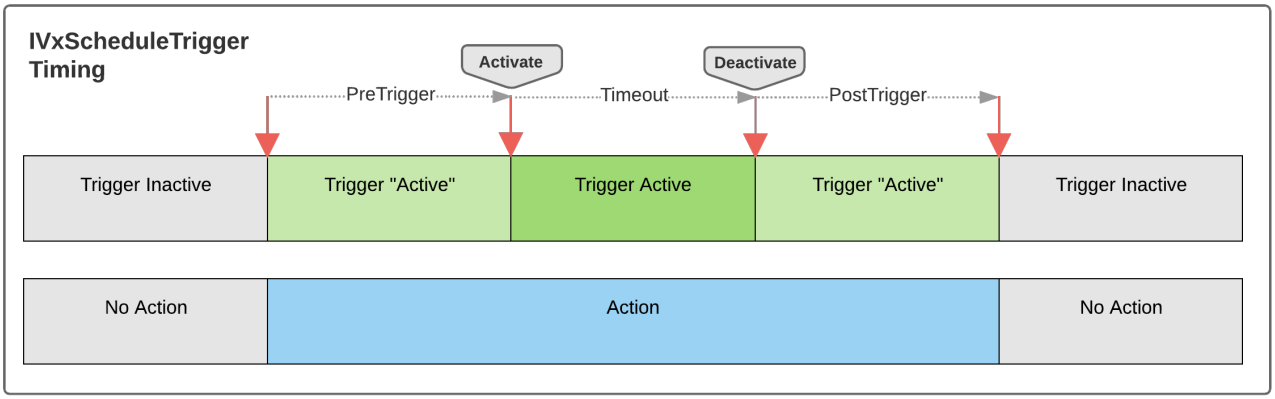

Figure 8.11: ScheduleTrigger Timing

A "bump-on-alarm" style IVxSchedule can be achieved by creating a IVxSchedule with both a "timed" IVxScheduleTrigger and an "event" IVxScheduleTrigger. If the event trigger is configured for a higher framerate than the timed trigger, you will wind up with bump-on-alarm:

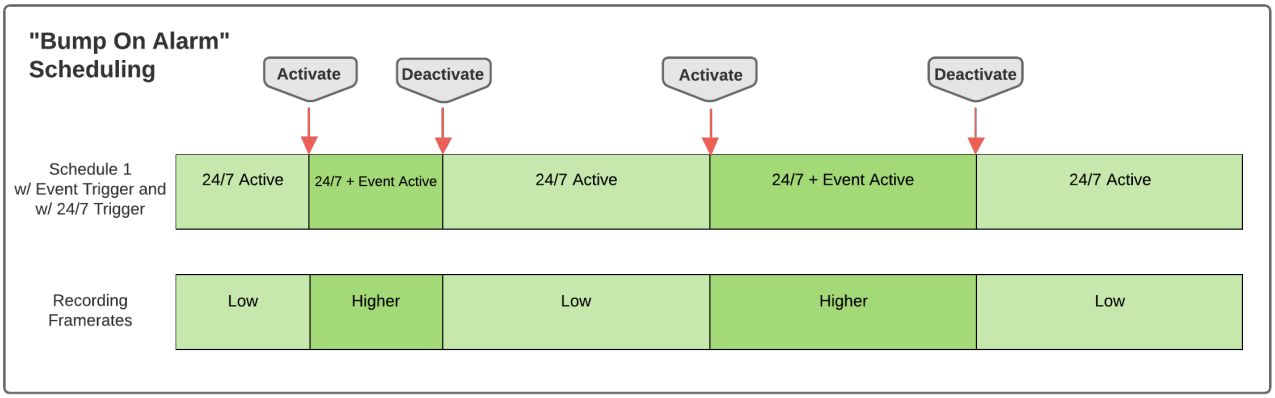

Figure 8.12: Bump-on-Alarm Schedule

8.10 Situation Defaults

IVxSituations provide the configuration for IVxEvent generation and handling. The table below specifies the defaults that shall be used for each IVxSituation prior to any custom configuration performed by clients or for attributes of VxNewSituation that are not specified. In the event of a reset to factory defaults, all IVxSituation configurations shall be returned to these defaults.

Field	Type	Default	Description
audibleLoopDelay	Integer	`0`	No delay.
audiblePlayCount	Integer	`1`	Play once.
autoAcknowledge	Integer	`-1`	No auto-acknowledgement.
isAckNeeded	Boolean	`False`	No acknowledgement needed.
name	String	[1]	[1] The Situation type (e.g. `admin/device_added`).
notificationIds	String[]	`[]`	No notification configuration.
severity	Integer	`5`	Medium severity.
shouldAudiblyNotify	Boolean	`False`	No audible notification.
shouldExpandBanner	Boolean	`True`	Display the in-cell banner.
shouldLog	Boolean	`True`	Log event.
shouldNotify	Boolean	`False`	Do not send notifications.
shouldPopupBanner	Boolean	`True`	Display the popup banner
snoozeIntervals	Integer[]	`[60,300,600]`	Default snooze intervals of 1 minute, 5 minutes, and 10 minutes.

Figure 8.13: Situation Defaults

8.11 Pixel Search Addendum

The pixel search grid is composed of rows and columns with zones indicating the areas of the grid to search. The grid uses a cartesian coordinate system (see image below) with 0-based indices.

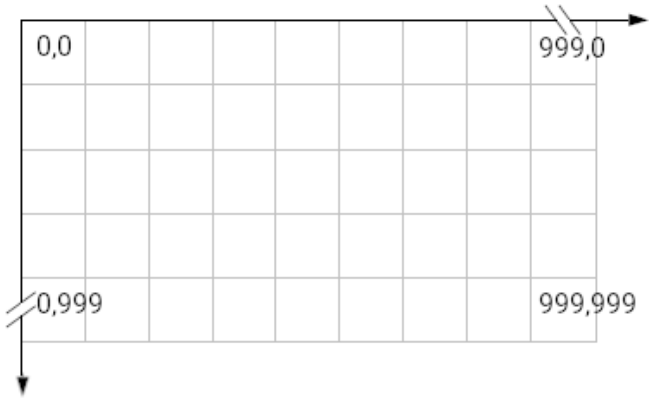

Figure 8.14: Pixel Search Grid

8.12 Continuous Move Velocity Diagram

Continuous positioning is used, for supporting devices, to continuously move the field of view at the given velocity speed until stopped. Velocity speeds are percentage based and can be both positive and negative (-100 to +100). Negative X values pan left and positive X values pan right while negative Y values tilt downward and positive Y values tilt upward (see image below).

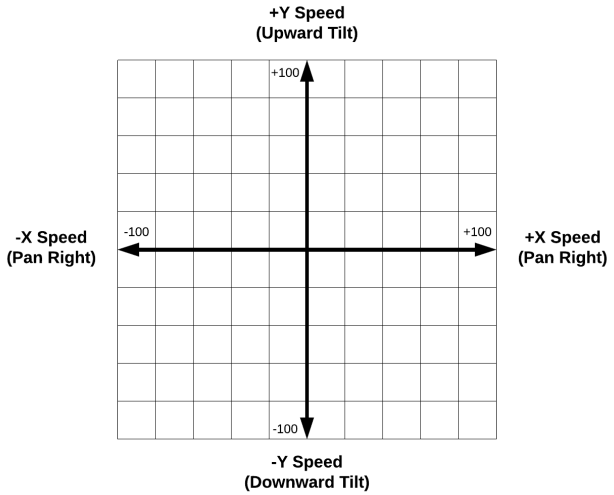

Figure 8.15: Continuous Move Velocity Directions