



CHAPTER 1

Overview

This chapter provides an overview of the Cisco VSM APIs and includes the following sections:

- [URL-Based Media Server API commands, page 1-1](#)
- [AxClient API Methods, page 1-2](#)

URL-Based Media Server API commands

Media server commands are URL-based API commands that are neither application-platform nor programming language specific. Commands are sent to dynamically loaded modules (for example, info.bwt, command.bwt, and event.bwt) on the media server using arguments in the form of name-value pairs. You can issue server API commands either manually or programmatically after an HTTP connection is established.

This section describes the four following server API command types:

- [VSMS Commands, page 1-1](#)
- [Proxy Commands, page 1-2](#)
- [Archive Commands, page 1-2](#)
- [Event Commands, page 1-2](#)

VSMS Commands

Cisco Video Surveillance Media Server (VSMS) commands perform the following media server-related functions:

- Retrieves the VSMS version
- Retrieves Session Description Protocol (SDP) information from a video stream
- Retrieves a Real Time Streaming Protocol (RTSP) stream for a third party video player
- Configures camera features, such as enabling or disabling backlight compensation or digital zoom
- Controls camera functions, such as PTZ movement, iris control, and focus

For more information on VSMS commands, see [Chapter 2, “VSMS Commands.”](#)

Text Part Number:

Proxy Commands

Proxy commands perform the following proxy-related functions:

- Starts, stops, or updates a proxy
- Views JPEG frames
- Lists all proxies
- Retrieves information about a proxy, such as its media source, media type, framerate or bitrate, video width or height, and status value.

For more information on proxy commands, see [Chapter 3, “Proxy Commands.”](#)

Archive Commands

Archive commands perform the following archive-related functions:

- Starts, stops, updates, renames, or removes an archive
- Lists all archives
- Creates an archive clip
- Retrieves information about an archive, such as its media type value, performance information, or details.

For more information on archive commands, see [Chapter 4, “Archive Commands.”](#)

Event Commands

Event commands perform the following event-related functions:

- Sets up, enables, disables, or removes an event
- Triggers an event
- Starts and stops an event-based clip
- Retrieves event information

For more information on event commands, see [Chapter 5, “Event Commands.”](#)

AxClient API Methods

VSM AxClient is an ActiveX-based API that is installed on a video surveillance client workstation, and it is used to control video operations (such as starting, pausing fast-forwarding, reversing, or stopping video streams), obtain information about AxClient or video streams, create clips and snapshots, control VMR display, and set up callbacks.

Developers can include API methods in their C# or JavaScript code to send commands to AxClient. For more information on AxClient API methods, see [Chapter 6, “AxClient API Methods.”](#)

Programming Notes for C#

The AXClient is written in C++ and inherently has different object types than C#. Many return values are different objects than referenced in the API documentation. Specifically:

- Return values of types date will return to C# as System.DateTime.
- Return values of types int, short and long will return to C# as System.Int32.

Programing Notes for JavaScript

The AXClient is written in C++ and inherently has different object types than JavaScript. Many return values are not native to JavaScript and require special consideration before they can be used. Specifically:

- Return values of type date are not native JavaScript Date objects. These return values will need to be recast to a JavaScript date object by passing it through the JavaScript Date constructor.
- Return values of types int, short, and long will return to JavaScript as an integer.
- Return values of types float and double return to JavaScript as a float.
- AxClient Parameter values that should be a Date must be case in JavaScript to a VT_DATE prior to passing to the AXClient. This is performed by using the Date object's getVarDate() method.

Sample Code:

```
// AXClient instantiation
<object codebase="AXClient.cab#version=5,0,17,0"
classid="clsid:41293422-93FD-443C-B848-E07EDBF866C3"
id="AXClient" name="AXClient" viewastext="true" width="350" height="280">
// Static Properties
<param name="name" value="AXClient" />
<param name="EnableDvrMode" value="false"/>
<param name="EnableVMRMode" value="false"/>
// Setup Callbacks
<param name="OnPlayerLoaded" value="userDefinedCallbackMethod"/>
// Default Values
<param name="timestamp" value="0"/>
</object>
```



Note

The AXClient version number will change from this documentation depending on which version of the AXClient is being used to develop against. The sample instantiation code above does not work around a known Microsoft Internet Explorer issue involving activating ActiveX Controls.

