

How to search and download the video file via ISAPI

Please refer to the ISAPI document (**General Application**) below first.

ISAPI Description	Search Video Files
Security	
Login	
Device Management	
Device Configuration and Maintenance	
User Management	
Video and Audio	
Capture and Recording	
PTZ Control	
Storage Management	
File Management	
Video Files	
Audio Files	
Log Files	
Download Files	

Function	Request URI
Get search conditions	GET /ISAPI/ContentMgmt/search/profile
Search video files	POST /ISAPI/ContentMgmt/search Note: This URI is not only used to search video files, it can also be used to se resources stored in the device.
Search video files by time	POST /ISAPI/ContentMgmt/record/tracks/<ID>/dailyDistribution
Get capability of searching video files by VCA event	GET /ISAPI/ContentMgmt/SmartSearch/capabilities
Search video files by VCA event	POST /ISAPI/ContentMgmt/SmartSearch

Lock or Unlock Video Files

1. Search the video file

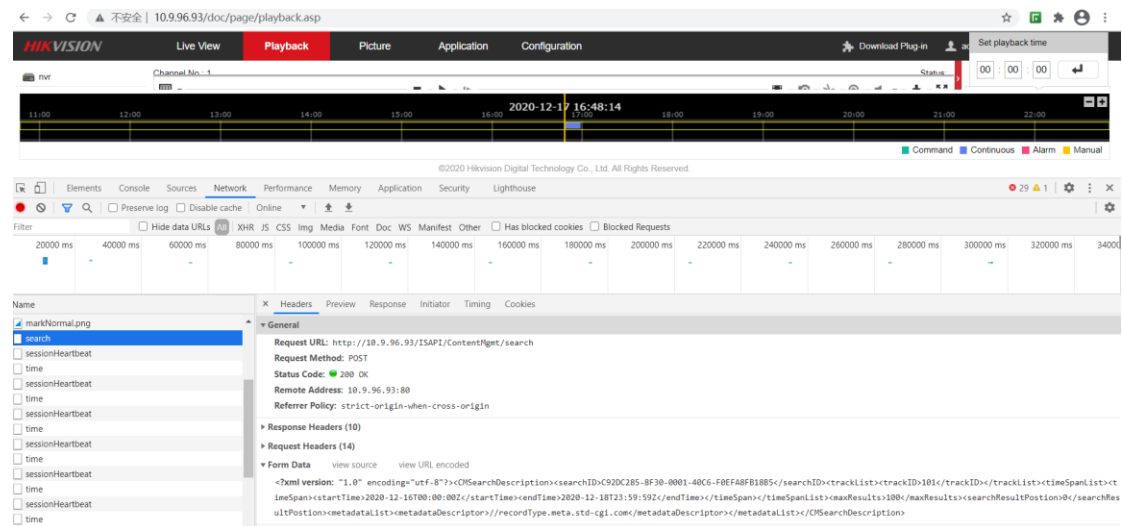
The Content Management 'Search' service is the primary component for conducting searches of content bases managed by a ISAPI recording device. Fundamentally, searches are initiated using a parameter-based criteria set which is conveyed by the initiator to the device via the 'CMSearchDescription' XML schema. *Since not all programming languages allow content bodies with HTTP GET methods, both 'GET' and 'POST' are supported as message types for initiating searches.* The responding device passes back the results in a 'CMSearchResult' XML schema instance for search requests that had valid syntax. If a search request is syntactically invalid (i.e. no payload, malformed schema instance, etc.), an HTTP response with status code 400 (Bad Request) and a corresponding 'Response Status' are returned to the requester. Please note that a syntactically correct search, that has no matching criteria, returns a 'CMSearchResult' schema instance with a 'NO MATCHES' status string.

Essentially, most searches are conducted based on time and/or track and/or source related search parameters. Full profile Content Management devices also support the potential for metadata search related parameters. More details, and examples, follow.

ISAPI command : /ISAPI/ContentMgmt/search

Method	GET or POST
Description	Search for specific resources.
Query	None.
Request	XML_VideoPic_CMSearchDescription
Response	Succeeded: XML_VideoPic_CMSearchResult Failed: XML_ResponseStatus

Please note the functions in the WEB interface are realized via ISAPI commands. Please login to the device via chrome with F12 debug mode, and you can see the details for the ISAPI command.



Example for the parameters

```
<?xml version="1.0" encoding="utf-8"?>
<CMSearchDescription>
  <searchID>C92DC285-8F30-0001-40C6-F0EFA8FB18B5</searchID>
  <trackList>
    <trackID>101</trackID>
  </trackList>
  <timeSpanList>
    <timeSpan>
      <startTime>2020-12-16T00:00:00Z</startTime>
      <endTime>2020-12-18T23:59:59Z</endTime>
    </timeSpan>
  </timeSpanList>
  <maxResults>100</maxResults>
  <searchResultPosition>0</searchResultPosition>
  <metadataList>
```

```
<metadataDescriptor>//recordType.meta.std-cgi.com</metadataDescriptor>
</metadataList>
</CMSearchDescription>
```

Postman Test

Postman interface showing a POST request to `http://10.9.96.93/ISAPI/ContentMgmt/search`. The request body is an XML document. The response is also XML, showing search results.

Request Body:

```
<?xml version="1.0" encoding="utf-8"?>
<CMSearchDescription>
  <searchID>C92DC285-8F30-0001-40C6-F0EFA8FB18B5</searchID>
  <trackList>
    <trackID>101</trackID></trackList>
  <timeSpanList>
    <timeSpan>
      <startTime>2020-12-16T00:00:00Z</startTime>
    </timeSpan>
  </timeSpanList>
</CMSearchDescription>
```

Response Body:

```
<?xml version="1.0" encoding="UTF-8" ?>
<CMSearchResult version="2.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <searchID>{c92dc285-8f30-0001-40c6-f0efa8fb18b5}</searchID>
  <responseStatus>true</responseStatus>
  <responseStatusStrg>OK</responseStatusStrg>
  <numOfMatches>1</numOfMatches>
  <matchList>
    <searchMatchItem>
      <sourceID>{00000000-0000-0000-0000-000000000000}</sourceID>
      <trackID>101</trackID>
      <timeSpan>
        <startTime>2020-12-17T16:48:14Z</startTime>
      </timeSpan>
    </searchMatchItem>
  </matchList>
</CMSearchResult>
```

When you operate the video search via WEB interface, you can also capture the network packages at the same time. In the network packages ,you can also get the detailed information as below.

Wireshark interface showing a packet capture of an HTTP POST request to `http://10.9.96.93/ISAPI/ContentMgmt/search`. The packet details show the request structure and the raw data.

Filter: `http&&ip.addr==10.9.96.93`

No.	Time	Source	Destination	Protocol	Length	Info
286	4.13288700	10.25.100.110	10.9.96.93	HTTP/XML	1215	POST /ISAPI/ContentMgmt/search HTTP/1.1
287	4.14748900	10.9.96.93	10.25.100.110	HTTP/XML	1327	HTTP/1.1 200 OK
1156	20.2640820	10.25.100.110	10.9.96.93	HTTP	638	PUT /ISAPI/Security/sessionHeartbeat HTTP/1.1
1157	20.2688470	10.9.96.93	10.25.100.110	HTTP/XML	653	HTTP/1.1 200 OK
1184	20.7973030	10.25.100.110	10.9.96.93	HTTP	578	GET /ISAPI/System/time HTTP/1.1
1185	20.8084340	10.9.96.93	10.25.100.110	HTTP/XML	589	HTTP/1.1 200 OK
1335	23.6610090	10.25.100.110	10.9.96.93	HTTP	670	PUT /ISAPI/Security/sessionHeartbeat HTTP/1.1
1336	23.6681940	10.9.96.93	10.25.100.110	HTTP/XML	653	HTTP/1.1 200 OK

Frame 286: 1215 bytes on wire (9720 bits), 1215 bytes captured (9720 bits) on interface 0

Ethernet II, Src: f8:59:71:5d:a6 (f8:59:71:5d:74:a6), Dst: 38:ad:be:a1:4a:45 (38:ad:be:a1:4a:45)

Internet Protocol Version 4, Src: 10.25.100.110 (10.25.100.110), Dst: 10.9.96.93 (10.9.96.93)

Transmission Control Protocol, Src Port: 6295 (6295), Dst Port: http (80), Seq: 1, Ack: 1, Len: 1161

Hypertext Transfer Protocol

extensible Markup Language

Raw Data:

```
000 38 ad be a1 4a 45 f8 59 71 5d 74 a6 08 00 45 00 8...JE.Y q]t...E.
010 04 b1 5f 0a 40 00 80 06 be 4f 0a 19 64 6e 0a 09 :...@... .O..dn..
020 60 5d 18 97 00 50 88 3b a7 52 0f af 79 4e 50 18 :...P.i .R..yNP.
030 00 fc fd 24 00 00 50 4f 53 54 20 2f 49 53 41 50 :...$.PO ST /ISAP
040 49 2f 43 6f 6e 74 65 6e 74 4d 67 6d 74 2f 73 65 I/Conten tMgmt/se
050 61 72 63 68 20 48 54 54 50 2f 31 2e 31 0d 0a 48 arch HTT P/1.1..H
060 6f 73 74 3a 20 31 30 2e 39 2e 39 36 2e 39 33 0d ost: 10. 9.96.93.
070 0a 43 6f 6e 65 63 74 69 6f 6e 3a 20 6b 65 65 .connect ion: kee
080 70 2d 61 6c 69 76 65 0d 0a 43 6f 6e 74 65 6e 74 p-alive. .Content
090 2d 4c 65 6e 67 74 68 3a 20 34 38 35 0d 0a 43 61 -length: 485..ca
nan 63 68 65 7d 43 6f 6e 74 72 6f 6e 3a 20 6d 61 78 che-cont rol: max
```

2. Download

ISAPI command: `/ISAPI/ContentMgmt/download`

Method	POST
Description	Download the file.
Query	None.
Request	XML_downloadRequest
Response	File

After the video search , you can get the **<playback URI>**.Please use <playback URI> to download the video file from the device.

The screenshot shows a REST client interface with a POST request to `http://10.9.96.93/ISAPI/ContentMgmt/search`. The request body is an XML document. The response status is 200 OK, and the response body is an XML document. A red box highlights the `<playbackURI>` element in the response XML.

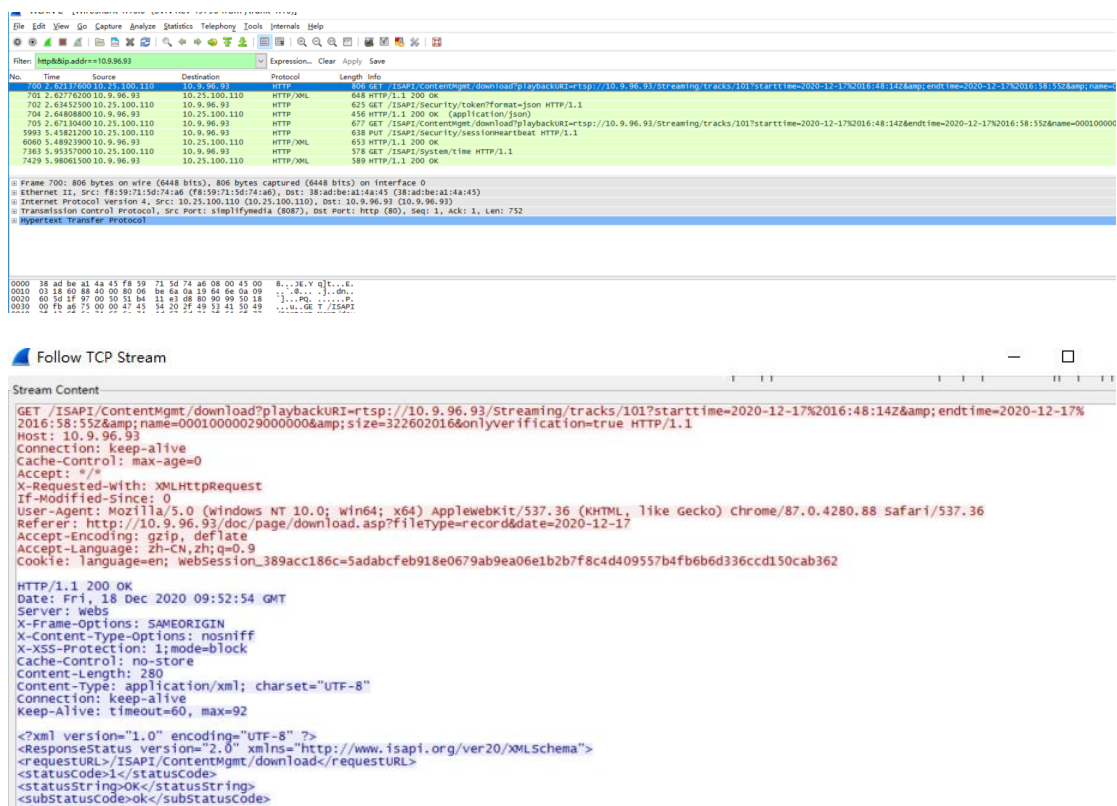
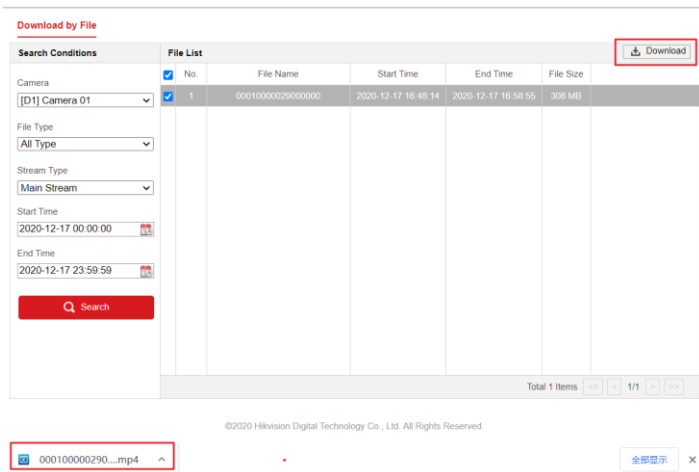
```

1  <?xml version="1.0" encoding="utf-8"?>
2  <CMSearchDescription>
3    <searchID>C92DC285-8F30-0001-40C6-F0EFA8FB18B5</searchID>
4    <trackList>
5      <trackItem>
6        <timeSpan>
7          <startTime>2020-12-17T16:48:14Z</startTime>
8          <endTime>2020-12-17T16:58:55Z</endTime>
9        </timeSpan>
10       <mediaSegmentDescriptor>
11         <contentType>video</contentType>
12         <codecType>H.264-BP</codecType>
13         <playbackURI>rtsp://10.9.96.93/Streaming/tracks/101/?starttime=20201217T164814Z&
14           amp;endtime=20201217T165855Z&name=00010000029000000&
15           amp;size=322602016</playbackURI>
16       </mediaSegmentDescriptor>
17       <metadataMatches>
18         <metadataDescriptor>recordType.meta.hikvision.com/timing</metadataDescriptor>
19       </metadataMatches>
20     </searchMatchItem>
21   </trackList>
22 </CMSearchDescription>
23 </trackList>
24 </matchList>

```

<playbackURI>rtsp://10.9.96.93/Streaming/tracks/101/?starttime=20201217T164814Z&endtime=20201217T165855Z&name=00010000029000000&size=322602016</playbackURI>

If you download the video file via WEB interface and get the Wireshark capture at the same time, you can see the detailed information as below.



Please note that some devices support GET and POST methods to download the vide file. So you can use the ISAPI commands below:

GET

/ISAPI/ContentMgmt/download?playbackURI=rtsp://10.9.96.93/Streaming/tracks/101?starttime=2020-12-17%2016:48:14Z&endtime=2020-12-17%2016:58:55Z&name=00010000029000000&size=322602016

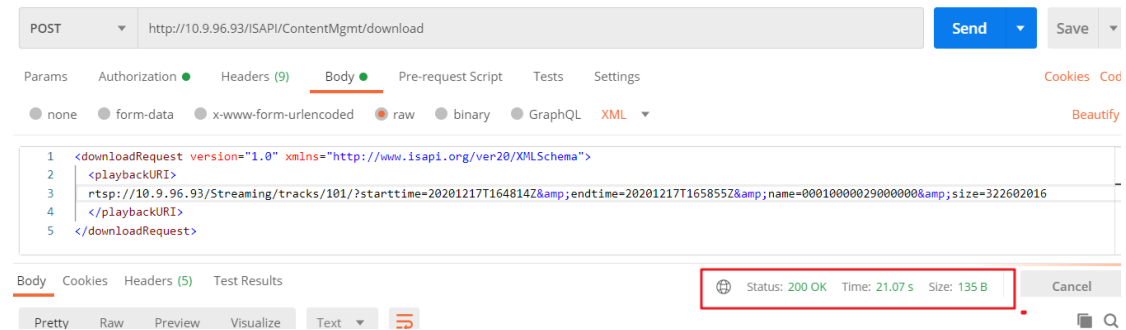
POST

/ISAPI/ContentMgmt/download

With request body parameters below:

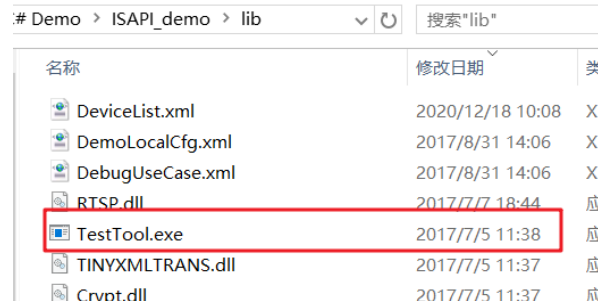
```
<downloadRequest version="1.0" xmlns="http://www.isapi.org/ver20/XMLSchema">
  <playbackURI>
    rtsp://10.9.96.93/Streaming/tracks/101/?starttime=20201217T164814Z&endtime=20201217T1658
55Z&name=00010000029000000&size=322602016
  </playbackURI>
</downloadRequest>
```

Postman test

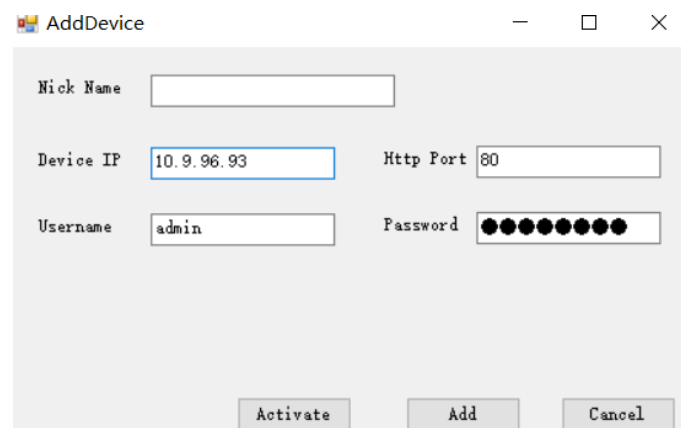


3. C# Demo Test

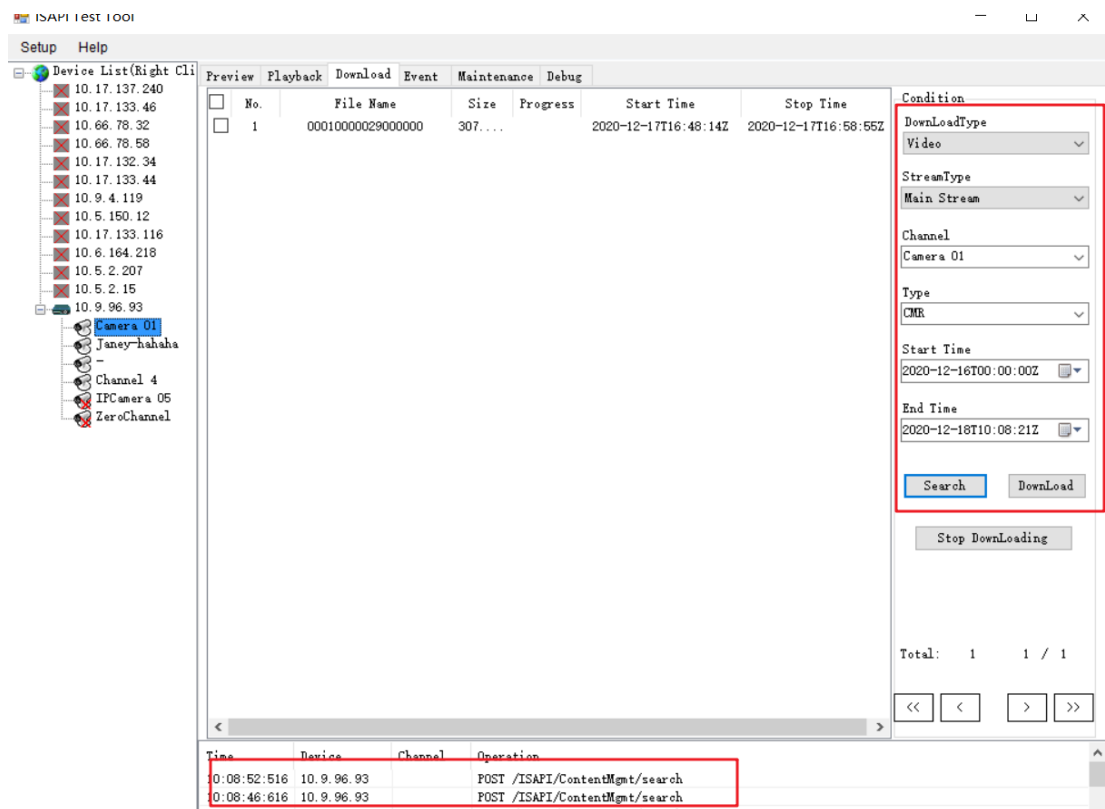
Step 1: Open the TestTool.exe



Step 2: Login to the device



Step 3: Search the video file via time



Step 4: Choose one video file for the download

