Avaya Voice Portal 5.0 IVVR White Paper

History

Version	Date	Author	Comments
1.0	April 29, 2009	Ashfaque Noori, Amol Ghode	Initial release
2.0	August 3, 2009	Ashfaque Noori, Amol Ghode	The video cache size has been increased from 10 MB to 100 MB; added Appendix A and Appendix B: DD Session variables for video.

Contents

1. Introduction

Voice Portal 5.0 introduces video capabilities allowing users of Voice portal 5.0 to develop multimedia applications using video, audio, images and text.

VP 5.0 has added video support by implementing few tags of VXML 3.0 specs and implementing SMIL 3.0 Tiny profile specifications.

Following figure shows a high level architecture diagram of VP MPP. Video manager is a new component introduced in VP 5.0 to handle video related media processing.

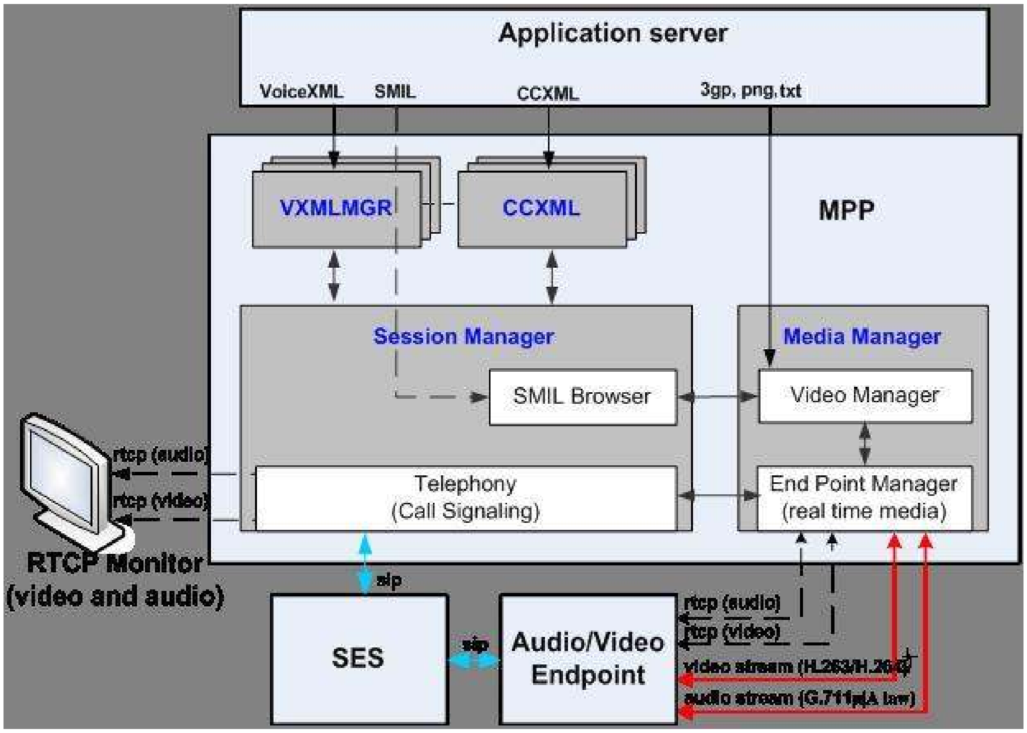
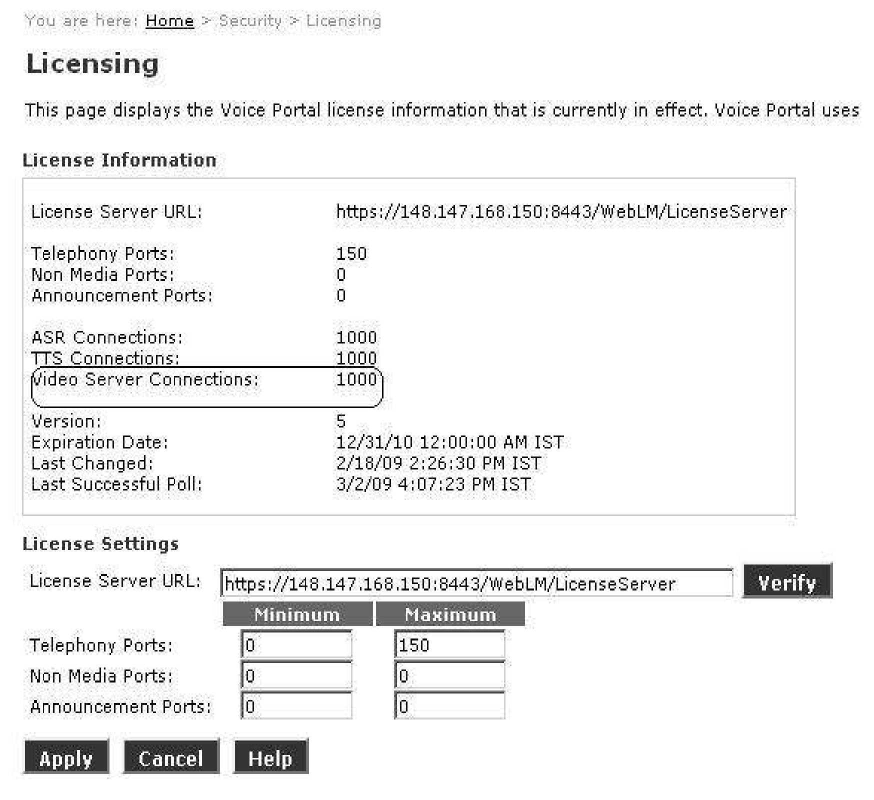


Figure 1. VP 5.0 MPP Architecture diagram

Video feature is implemented in VP 5.0 for SIP connections only & is not supported on H323 connections. It is also important to understand that VP 5.0 supports only video applications generated by Dialog Designer version 5.0 or higher.

2. Video Licenses

For executing video applications on VP 5.0, you need to have video licenses. The number of video licenses should be equivalent to the number of concurrent video channels that the customer plans to run.	

You are here: Home > Security > Licensing

Licensing

This page displays the Voice Portal license information that is currently in effect. Voice Portal uses

License Information

License Server URL:	https://148.147.168.150:8443/WebLM/LicenseServer
Telephony Ports:	150
Non Media Ports:	0
Announcement Ports:	0
ASR Connections:	1000
TTS Connections:	1000
Video Server Connections:	1000
Version:	5
Expiration Date:	12/31/10 12:00:00 AM IST
Last Changed:	2/18/09 2:26:30 PM IST
Last Successful Poll:	3/2/09 4:07:23 PM IST

License Settings

License Server URL: https://148.147.168.150:8443/WebLM/LicenseServer

	Minimum	Maximum
Telephony Ports:	0	150
Non Media Ports:	0	0
Announcement Ports:	0	0

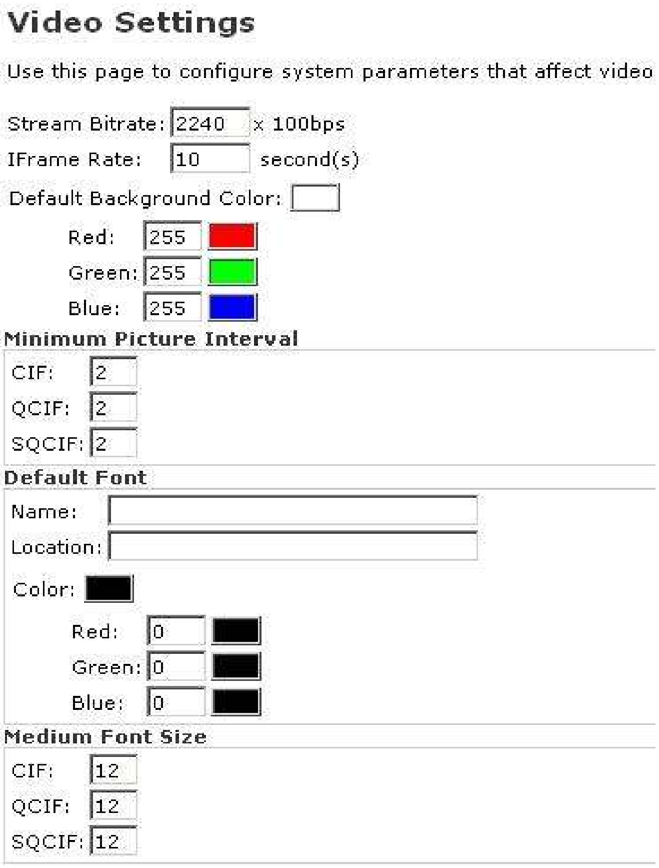
3. VPMS Configuration

This section will explain the various video related fields that can be configured through VPMS.

Global Settings

You can navigate to this page in VPMS by clicking on System Configuration-->MPP Server -->Video Setting.

Following figure shows a snapshot of a sample configuration



Video Settings

Use this page to configure system parameters that affect video

Stream Bitrate: 2240 x 100bps

IFrame Rate: 10 second(s)

Default Background Color: □

Red: 255 🟥

Green: 255 🟩

Blue: 255 🟦

Minimum Picture Interval

CIF: 2

QCIF: 2

SQCIF: 2

Default Font

Name:

Location:

Color: ⬛

Red: 0 ⬛

Green: 0 ⬛

Blue: 0 ⬛

Medium Font Size

CIF: 12

QCIF: 12

SQCIF: 12

Figure 2. Video Configuration in VPMS

Following table explains each of the configuration parameters related to video

Field Name	Remarks
Stream Bitrate	The term ‘bitrate’ refers to the amount of data that can be transferred per second. It represents number of bits processed per second. This has traditionally been the main way to control the quality of your stream, although there are other ways to improve the quality (for example: changing your framerate, source quality, etc.) Valid value range for this parameter is 0 – 19200. If the value is set to 0, then it is ignored by the platform & the default value 1920 * 100 bps is used. It is observed that If the value is less than 1200, quality of video is bad.

<table>
<tr><td></td><td>Recommended value: Set it to 0 & fine tune based on your network.

For endpoints sending stream bitrate in MaxBR parameter in SIP SDP, the session bitrate is the minimum of Bit rate from End point and bit rate configured on “Global settings” page.
For end points not sending any bit rate or sending the bit rate in parameter other than MaxBR, session bit rate is always the stream bitrate configured on VPMS global settings page.</td></tr>
<tr><td>IFrame Rate</td><td>This parameter sets the minimum time interval, in seconds, between two consecutive IFrames.

For example, to get one IFrame in every 20 seconds, enter 20 in the IFrame Rate field.

An I-frame is a single frame of digital content that the compressor examines independent of the frames that precede and follow it and stores all of the data needed to display that frame. Typically, a I-frames are interspersed with P-frames and B-frames in a compressed video. The more I-frames that are contained, the better quality the video will be; however, I-frames contain the most amount of bits and therefore take up more space on the storage medium.

Enter a value in the range 1 to 99999.

The default is 10 seconds.</td></tr>
<tr><td>Default background color</td><td>This parameter allows you to select the default background color of the layout.

For Media tag / Regions where no Background color specified, default background color will apply.</td></tr>
<tr><td colspan="2">Minimum Picture Interval</td></tr>
<tr><td colspan="2">This value determines video format and frames per second.
Frames per second is calculated as under
Frames per second (FPS) = 30 / MPI.
MPI value is typically driven by your application & media needs. For e.g. if you have fast moving videos, you might want to have a lower MPI (higher FPS).
Please note the video format and MPI heavily impact MPP’s CPU</td></tr>
</table>

Utilization & are major factor in determining the video port capacity. Please refer to Performance section for more information on video capacities.	
CIF	1 to 32 default =2. Actual MPI at which the video is streamed is determined based on the rules mentioned in the section Please see section MPI Determination rules for video for more info.
QCIF	1 to 32 default =2. Actual MPI at which the video is streamed is determined based on the rules mentioned in the section Please see section MPI Determination rules for video for more info.
SQCIF	1 to 32 default =2. Actual MPI at which the video is streamed is determined based on the rules mentioned in the section Please see section MPI Determination rules for video for more info.
Default Font	
Allows you to set global font related values. These font settings are used if no font related information is provided in the applications.	
Name	luximr
Location	Keep the font in $AVAYA_MPP_HOME/VideoMgr/fonts folder. By default VP has 12 fonts installed in the above directory. Please note VP 5.0 supports only TTF fonts. Keeping the font files at any other locations is not currently supported. Please note font names are case sensitive
Color	This value indicates the default font color.
Medium Font Size	
CIF	0-32
QCIF	0-32
SQCIF	Not tested

Application level Settings

Video Enable

- Should be set to Yes for video applications.

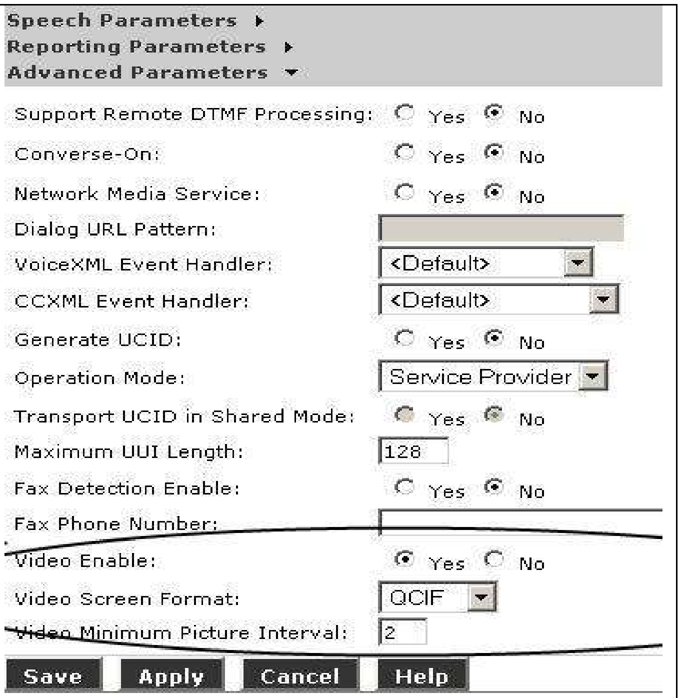
Video Screen Format

- Would be used only if the far end did not send the format.
- If the far end sent (e.g. QCIF= 2) but in the global setting is QCIF=0,CIF=0

Video Minimum Picture Interval

- Would be used only if far-end SDP does not have media attribute for video. E.g.: if SDP does not contain media attribute as a=fmtp:<vcodec_id> CIF=x QCIF=x MaxBR=1960

It is recommended that Video Screen format and MPI should match with the Application’s 3gp files video format and fps



Speech Parameters ▸
Reporting Parameters ▸
Advanced Parameters ▾

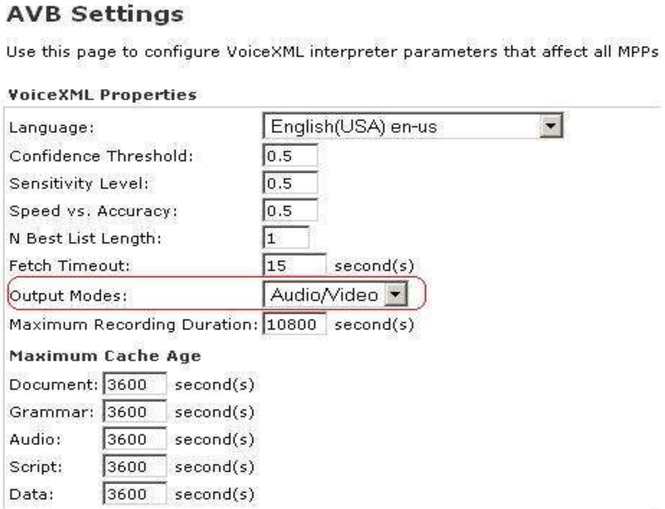
Support Remote DTMF Processing: ○ Yes ● No
Converse-On: ○ Yes ● No
Network Media Service: ○ Yes ● No
Dialog URL Pattern:
VoiceXML Event Handler: <Default>
CCXML Event Handler: <Default>
Generate UCID: ○ Yes ● No
Operation Mode: Service Provider
Transport UCID in Shared Mode: ○ Yes ● No
Maximum UUI Length: 128
Fax Detection Enable: ○ Yes ● No
Fax Phone Number:
Video Enable: ● Yes ○ No
Video Screen Format: QCIF
Video Minimum Picture Interval: 2

Figure 3. App level configurations for Video

AVB Settings

Output Modes

- Please ensure that output Modes parameter is set to ‘Audio/Video’.



AVB Settings

Use this page to configure VoiceXML interpreter parameters that affect all MPPs

VoiceXML Properties

Language: English(USA) en-us
Confidence Threshold: 0.5
Sensitivity Level: 0.5
Speed vs. Accuracy: 0.5
N Best List Length: 1
Fetch Timeout: 15 second(s)
Output Modes: Audio/Video
Maximum Recording Duration: 10800 second(s)

Maximum Cache Age

Document: 3600 second(s)
Grammar: 3600 second(s)
Audio: 3600 second(s)
Script: 3600 second(s)
Data: 3600 second(s)

Figure 4. AVB Configurations for Video Apps

Video Cache Size

The video cache size has been increased from 10 MB to 100 MB.

4. Differences between Simulator and Platform

Dialog Designer 5.0 uses Ambulant 2.1 for simulating video applications. There are certain differences between the way video apps play on Ambulant and on VP. This section documents summarizes some of the major differences between the simulator and platform

Feature	Platform behavior	DD behavior	Remarks
3gp and tts in VXML parallel node[+]	Only TTS heard	Audio from both elements are heard.	VP behavior is correct. DD Simulator and Ambulant player play the media concurrently.
Play a media tag with following setting			Ambulant takes the smallest of the 2 attributes (Duration / clipend) and the VP always takes the clipend.
Duration=10, clipend=5	Video stopped at 5s	Video stopped at 5s	
Duration=5 and clipend=10	Video stopped at 10s	Video stopped at 5s	
Play a media tag with following setting			
repeatCount =2	Video repeats	Video does not repeat	Ambulant does not repeat the video. It appears Ambulant honors repeatCount only if the duration attribute is set along with it.
repeatDur = 20	Video repeats	Video does not repeat	
Play media with following values set (media length - 16s)			

repeatDur=25,
repeatCount=2,
End=20
	Platforms repeats 3 times as End value overrides repeatDur
	(It's repeated twice and for the third time played for 8 seconds. So total play would be for 40 sec).
		Media length 16s or even more then 20s (End value) : Multiple instances of Haali Video Splitter and ffdshow video decoder keep getting created and CPU shoots to 100%.	

repeatDur=25,
repeatCount=2,
clipEnd=30
	Video played 4 times. 4th time for a few seconds. clipEnd and repeat count are honored.	Since the media length was 16, it played for 16s and silence for another 14s and then did not	

	ClipEnd value overrides repeatDur value. (Total for 30sec *2= 60 sec).	repeat at all.	
png file with transparency support (4 channel PNG files)	Playback is supported.	Playback is not supported in simulator.	
Video playing without any regions defined. But regions are added under SMIL Head- have some background color.	The background color of the region may show for an instant depending on the value of showbackground attribute.	For the entire duration of the video, a region of background color defined, blocks the video.	
2 videos in VXML parallel[+]	Only the first video and its audio plays.	Second video played twice and the execution does not proceed to the next prompt.	
2 audios in VXML parallel[+]	Only the first audio plays.	Second audio played twice and the execution does not proceed to the next prompt.	
2 videos in VXML Sequence [++]	Both play in order.	Only the second plays twice.	
2 audios in VXML Sequence[++]	Both play in order.	Only the second plays twice.	
Fill =”freeze” in SMIL parallel[+++]	If duration is specified, media freezes beyond the duration.	Media is removed.	
BargeIn	BargeIn of video prompts is supported.	Barge In functionality cannot be simulated with DD for video prompts. You need to stop Ambulant to enter digits.	
TTS and video in SMIL parallel[+++]	Not supported	Not supported	

[+] Use Coordination->Parallel tag on palette of a prompt

[++] Use Coordination->Sequence tag on palette of a prompt

[+++] Use Media->parallel tag on palette of a prompt

5. VXML 3.0 Compliance

Element	Attribute	Remarks
media	src	• file://, http are supported. • https is supported for audio, image and 3gp files only. • Currently https is NOT supported for referencing SMIL document inside a media element. • RTSP is NOT supported
	srcexpr, type, clipBegin, clipEnd, repeatCount, repeatDur, outputmodes	Supported
	soundLevel	Not Supported
	speed	Not Supported
seq		
par		
	endSync	Supported
record		
	recordmodes	Not supported for video

6. SMIL 3.0 Tiny Profile Compliance

Element	Attribute	Comments
<smil>		
<head>		
<body>		
<layout>	type	Supported
<region>		
	backgroundColor, bottom, fit, height, left, regionName, right, showBackground, top, width, z-index, textColor, textFontFamily, textFontSize	Supported
	textFontStyle	Not supported
	textFontWeight	Not supported
<root-layout>	backgroundColor , height, width	Supported
<meta>		Not supported
<metadata>		Not supported

<ref>	src, type, begin, end, dur, clipBegin, clipEnd, repeatDur, repeatCount, fill	Support for Audio, video, Image only.
<audio>	src, type, begin, end, dur, clipBegin, clipEnd, repeatDur, repeatCount	Supported
<img>	src, type, begin, end, dur, repeatDur, repeatCount, fill	Supported
<text>	src, type, begin, end, dur, repeatDur, repeatCount	Supported
<video>	src, type, begin, end, dur, clipBegin, clipEnd, repeatDur, repeatCount, fill	Supported Please note : clipEnd values override dur value.
<animation>		Not supported
<textStream>		Not supported
<smilText>	src, type, begin, end, dur, repeatDur, repeatCount	Supported Note: For newline “/n” is supported on the platform but is not supported.
	textMode	Only textMode="crawl" is supported.
	textWritingMode	supported values are "lr" and "rl"
<par>		
	endsync	Supported
	fill	Supported
<seq>		
	fill	Only “remove” Supported

7. Supported Media Types

Media Type	Supported file type
Video	3gp
Audio	Only Single channel 8000 Hz encoded wav file supported.
Image	png

Please note

- Media files can be accessed using file://, http & https.
- For https, you will need to install the certificates on mpp.
- Using https is currently NOT supported for SMIL documents.
- RTSP is currently NOT supported.

8. Supported Codecs

VP supports following H263 (RFC-2190 compliant video header) & H263-1998 (RFC-2429 compliant video header).Supported Codec configurations are kept in $AVAYA_MPP_HOME/config/mppconfig.xml file

```
<parameter name="sip.video.h264.fmtp">profile-level-id=42e00a; packetization-mode=1;
max-br=452; max-mbps=11880</parameter>
 <parameter name="sip.video.h263.fmtp">QCIF=1 CIF=1 I=1 J=1 T=1
MaxBR=4520</parameter>
 <parameter name="sip.video.h264">false</parameter>
 <parameter name="sip.video.h263.1998">true</parameter>
 <parameter name="sip.video.h263.2000">false</parameter>
<parameter name="sip.video.screen.size.pref">"QCIF=1;CIF=1;SQCIF=1"</parameter>
```

Figure 5. mppconfig.xml file

Please note:

- Parameter “sip.video.screen.size.pref”, the MPI value is overridden by the global settings done on voice portal admin page.
- Parameter “MaxBR” parameter is also overridden by global settings done on voice portal admin page.

Codec Negotiation scenarios

Final negotiated codecs in a video session depend on codec requests made by the end point and what codecs supported by the platform. Following sections describe how negotiation happens between the end point and VP in certain scenarios.

Scenario 1:

End point request

```
a=rtpmap:115 H263-1998/90000
a=rtpmap:34 H263/90000
```

MPP response

```
a=rtpmap:115 H263-1998/90000
a=fmtp:115 QCIF=2;MAXBR=4520
```

In above case, end point requested H263-1998 and H263, assuming both were configured on MPP, MPP responded back with the first supported codec it found i.e. H263-1998.

Scenario 2:

End point request

```
a=rtpmap:125 H264/90000
a=rtpmap:34 H263/90000
```

MPP response

```
a=rtpmap:34 H263/90000
a=fmtp:34 QCIF=2;MAXBR=19200
```

In above case, end point requested H264 and H263, since H264 is not supported by MPP, it negotiated using H263.

Scenario 3:

End point request

```
a=fmtp:125 profile-level-id=42e00a; packetization-mode=1; max-br=452; max-
mbps=11880
a=rtpmap:125 H264/90000
a=sendrecv
```

MPP response

```
a=rtpmap:101 telephone-event/8000
m=video 0 RTP/AVP 125
a=rtpmap:125 H264/90000
```

In the above case end point requested H264 which is not supported by MPP. In the MPP response ‘Video 0’ indicates failure of video negotiation.

9. Supported video formats

VP 5.0 supports CIF (352 x 288), QCIF (176 x 144) and SQCIF (128x96) as video formats.

When multiple video formats are requested by the end point, platform will select the first supported format and use it for establishing the video session.

Consider a typical end point request & response

End point request

```
a=fmtp:34 QCIF=1 CIF=1 MaxBR=4520
```

MPP response

```
a=fmtp:34 QCIF=2
```

As you can see in the above example, end point requested for both QCIF & CIF, since QCIF was encountered first, MPP honored this format and responded.

Now, if global settings in VPMS are set as QCIF = 0 (disable QCIF) and CIF = 2, then MPP response would be as under

MPP response

```
a=fmtp:34 CIF=2
```

Please note only If global settings are set as CIF=0 and QCIF=0, then app level configurations are used.

Following table illustrates few sample scenarios on how a screen format would be chosen.

End point requests	Global setting	App level setting	Actual format streamed
QCIF,CIF	QCIF=2,CIF=2	CIF =2	QCIF
QCIF,CIF	QCIF=0,CIF=2	QCIF=1	CIF
QCIF,CIF	QCIF=0,CIF=0	CIF=1	CIF
QCIF,CIF	QCIF=0,CIF=0	QCIF=2	QCIF

10. MPI determination rules for video

When a video session is established with an end point, MPI for the session is determined based on various factors. Following table explains the various permutations and combinations of MPI values and how the final MPI for negotiation is selected.

End point MPI	File MPI	App MPI	Global MPI	Actual MPI @ which video is streamed
QCIF =2	QCIF = 4	-	QCIF =1	QCIF =2 Session MPI = max MPI of (End point, global) , Actual MPI = min MPI of (Session MPI, File MPI) Video streamed at QCIF =2.
QCIF=2	QCIF=1	-	QCIF=1	QCIF=1 Session MPI = max MPI of (End point, global) , Actual MPI = min MPI of (Session MPI, File MPI) Video streamed at QCIF =1
QCIF=2	QCIF=1	-	QCIF=4	QCIF=1 Session MPI = max MPI of (End point, global) , Actual MPI = min MPI of (Session MPI, File MPI) Video streamed at QCIF =1
QCIF=1	QCIF=4	-	QCIF=2	QCIF=2 Session MPI = max MPI of (End point, global) , Actual MPI = min MPI of (Session MPI, File MPI) Video streamed at QCIF =2

Please note:

- The above table has used QCIF throughout for consistency reasons only. The above rules also apply for other formats such as CIF and SQCIF.
- If global MPI is set to 0, then App MPI is considered in calculation of session MPI.

11. Troubleshooting IVVR Apps

Video Logs

- Log Level for Video Manager can be set from VPMS configuration page

    - invoke MPP Server ->MPP Setting page and change trace level for category "Media Video Manager"
- Detailed log can be viewed in MPP log directory $AVAYA_MPP_HOME/logs/process/MediaMgr which contains following log files
    - VideoMgr.log – it contains global logs.
    - VideoMgr-SessionSlot-xxx.log – where xxx is sessionSlot ID. it contains logs specific to a video session.
    - SMIL Related Log file can be viewed at session manager log directory $AVAYA_MPP_HOME/logs/process/SessMgr in session specific log file.

Troubleshooting tips

Symptom	Tips
I hear 'Technical difficulties' prompt when I run video apps.	• Ensure your end point supports video. • "Verify" the application from the VPMS "Change application" page • Ensure that application is video enabled. • Check that the application server is running.
When I run video apps, only audio is heard.	• Check the output mode and ensure it is set appropriately.
Video Quality is not good	• Check network condition between end point and VP. • Fine tune the stream bit rate .
I see lip sync issues when playing 3gp files	• Ensure the 3gp file FPS matches with the negotiated FPS.

12. Performance

IVVR is supported only on high end Voice Portal Systems (Dell 1950 Dual CPU, quad core). VP 5.0 SP1 sizing tool enforces this by not allowing you to size video for any other low end systems.

Number of Video ports supported on platform is largely determined by the following factor –

Parameters	Remarks
CIF/QCIF	QCIF gives you good performance in comparison to CIF.
MPI	Frames per second (FPS) = 30 / MPI. Lesser the FPS, less the CPU Utilization which means high IVVR ports.
Overlays	More overlays you have in the app mean lesser concurrent ports. This is especially true if you have multiple video overlays.

	Overlays can be “Text or Image on Video”, “Video on Video and “Text/Image on Image”. For every additional video overlay, on an average port capacity would drop by 20% and for every additional text or image overlay, on an average port capacity would drop by 5%.

Let’s consider a case where a customer wants to know how many MPPs he needs to procure for 150 video ports

Depending on his choice of video format and MPI for an application which uses H263 & no overlay, one would need following number of MPPs on a high end Dell 1950 server.

QCIF=1 (i.e. 30 fps) - 3 MPPs
QCIF=2 (i.e. 15 fps) - 2 MPPs
 	
CIF=1 (i.e. 30 fps) - 10 MPPs
CIF=2 (i.e. 15 fps) - 5 MPPs

Now if we add one video overlay, the above numbers will change as follows

QCIF=1 (i.e. 30 fps) - 4 MPPs
QCIF=2 (i.e. 15 fps) - 3 MPPs
 	
CIF=1 (i.e. 30 fps) - 13 MPPs
CIF=2 (i.e. 15 fps) - 7 MPPs

Where as adding one text/image overlay to original configuration would result in

QCIF=1 (i.e. 30 fps) - 4 MPPs
QCIF=2 (i.e. 15 fps) - 3 MPPs
 	
CIF=1 (i.e. 30 fps) - 10 MPPs
CIF=2 (i.e. 15 fps) - 6 MPPs

Note: IVVR ports suggested by sizing tool are a guidance number, the actual number varies based on customer’s application, media contents etc. Given this it is important that we set the correct expectations on this when using the sizing tool.

Note: The above calculations are for taking video as a base media. The number of channels would increase by 8% for image as a base media.

13. Open issues/Limitations

- Video recording is not supported with Voice Portal 5.0.
- Video conferencing is not supported with Voice Portal 5.0.
- Configurable Application variables cannot be used with Media tags.
- In VXML par tags – same media types cannot be used.
- Timing attributes has to be explicitly used for media like images or text. Such media do not have any intrinsic duration so their simple duration is zero. So such media if used without any timing attribute will disappear immediately from the screen. This behavior is in compliance with the SMIL specs.
- If VXML form has TTS and SMIL in parallel and the raw SMIL is trying to play an image file or text without any duration, the image is removed as soon as it is shown. To show image for the duration of TTS play, timing attributes like dur, end etc has to be used explicitly with image or text. Please refer to Appendix A for more detailed explanation.
- Similarly if VXML form has TTS and SMIL in parallel and the raw SMIL is trying to play a video file which is of shorter length than TTS, the video will be removed as soon as it is over. In 5.0 GA the last video frame is frozen till TTS is completed. In order to keep the last video frame frozen, use explicit long duration for the video and use fill=”freeze”. Please refer to Appendix A for more detailed explanation.

14. Support for End points

VP 5.0 IVVR functionality should be able to work with any SIP based Video enabled end points. Testing has been carried out with following end points in the lab.

End points	Remarks
Kapanga – 1.00.2173B	Has issues with H263-1998
XLite – 3.0 build 41150	
IBeam -	
Avaya one-X Communicator - R1.0.1 GA-14211	

IVVR should work with any SIP compliant gateways like Delithium or any other IMS gateways. Please note this has not been tested in the lab.

15. Integration with other Avaya Products

Supported SES version – 4.1 and above

IVVR has not been tested for inter operability with IC 7.1.2 or above

Appendix A

In VP 5.0 GA, if a VXML form has a TTS and SMIL in parallel and the raw SMIL tries to play image or text without any timing attributes (dur etc), the image/text just keeps frozen for the remaining duration of TTS play. However with VP 5.0 Service Pack 1, this behavior has changed, and it is in compliance with the SMIL specs. To keep the image or text visible for the duration of TTS play, explicit timing attributes must be used with image/text.

Similarly if a VXML form has a TTS and SMIL in parallel and the raw SMIL tries to play a video file which is of shorter duration than TTS , the last video frame just keeps frozen for the remaining duration of TTS play. However with VP 5.0 Service Pack 1, this behavior has changed. To keep the last video frame frozen for the duration of TTS play, use explicit timing and fill=”freeze”.

Following lines describe few such scenarios and the best possible workarounds for them.

1. A VXML form has a TTS and a SMIL in parallel. The raw SMIL is trying to play a video file.
The duration of the video is 15 seconds and the duration of TTS is 25 seconds.
Earlier Behavior: The last frame of the video is frozen till the TTS completes.

Workaround: Do <par> endsync "first". Specify a large duration in video and make it freeze at the end.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
<form>
<block>
    <prompt>
      <par endsync="first">
        <media type="application/smil">
          ...
          <body>
              <par fill=”freeze”>
                  <video src="video.3gp" dur="3600.0" />
                  <smilText region="reg1" dur="3600s">
                      When can we contact you?
                  </smilText>
              </par>
          </body>
          ...
        </media>
       Can we make it to the end of the game?
   </par>
    </prompt>
</block>
</form>
</vxml>
```

2. A VXML form has a TTS and a SMIL in parallel. The raw SMIL is trying to play a image file without any duration.
Earlier Behavior: The image is frozen till the TTS completes.

Workaround: Do <par> endsync "first". Specify a large duration in image and make it freeze at the end.

```
<?xml version="1.0" encoding="UTF-8"?>
<vxml xmlns="http://www.w3.org/2001/vxml"
<form>
<block>
   <prompt>
     <par endsync="first">
        <media type="application/smil">
          ...
          <body>
               <par fill=”freeze”>
                    <img src="image.png" dur="3600.0" />
                    <smilText region="reg1" dur="3600s">
                         When can we contact you
                    </smilText>
               </par>
          </body>
          ...
        </media>
      Can we make it to the end of the game?
  </par>
   </prompt>
</block>
</form>
</vxml>
```

3. For non-bargeable prompts, if input does not start until after the TTS is finished, the last video frame MUST not disappear while waiting for user input. Add a little silence to TTS, so the smil image will stay for a little while after TTS completes. If necessary, user can break a prompt into two, making the important part non-bargeable.

```
<vxml>
...
<prompt bargein = "true">
       <par endsync="first">
          <media src = "long_smil.smil"/>
            <seq>
                This is the TTS to play.
                <media src="silence.wav"/>
            </seq>
       </par>
</prompt>
...
</vxml>
```

Following table summarizes the above scenarios.

Behavior	Modification
White screen between prompts	Use coordination parallel prompt elements to play audio/TTS in parallel with video. For each prompt, make sure that there is a video playing from the beginning of the prompt to the end. If the video is shorter that the audio/TTS set the duration attribute for the videos media tag.
White screen between reprompts	Specify small prompt timeout values to minimize the white screen between reprompts and make sure the duration of the video is set long enough.
Synchronization problems with audio/video	Make use of coordination parallel and sequence to get the desired behavior. It is recommended to set the “endsync” attribute for the coordination parallel.
Application not working for a particular screen size	Use different prompts for different screen sizes or use a single prompt and add if/else statements to it.
Non-bargeable prompts looses video when getting to input state	Insert a silent bargeable audio file after the last non-bargeable audio prompt and increase the duration of the video.

Appendix B: DD Session variables for video

Dialog Designer 5.0 SP1 exposes following session variables for video.

videobitrate, videocodec, videoenabled, videofarfmtp, videoformat, videofps, videoheight, videonearfmtp, and videowidth. Ideally these variables are READ-ONLY and should not be set in DD application.

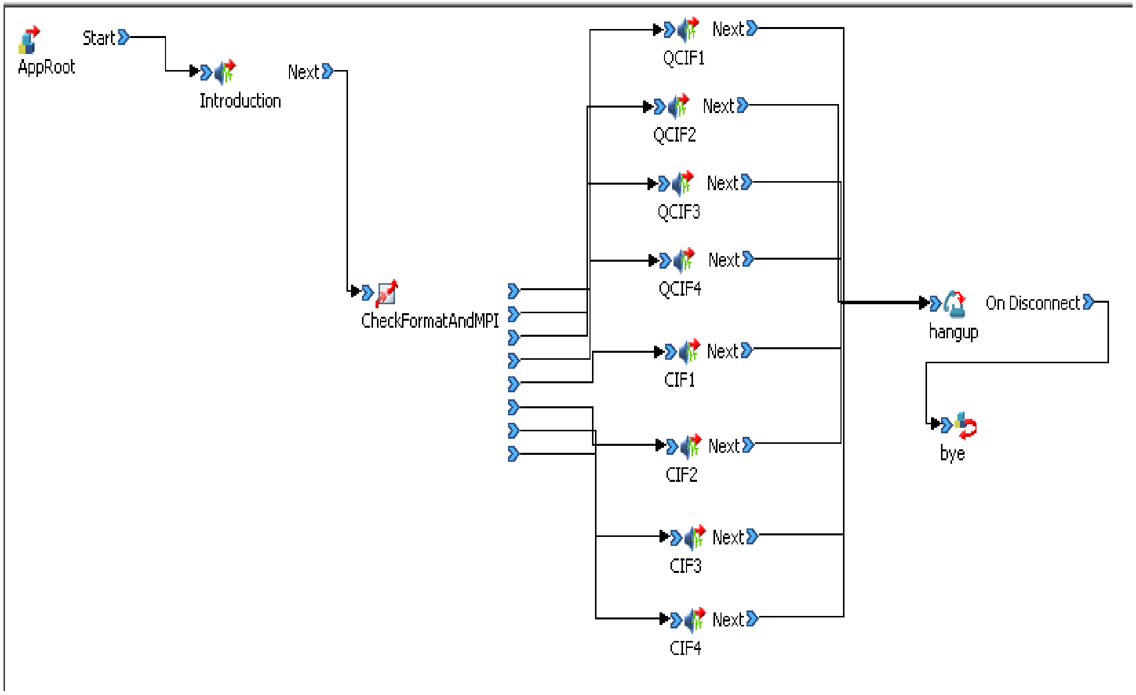
videoenabled: Possible values are true or false
videocodec: Possible values are H263or H263+ (or H263-1998). This parameter comes from phone. If mpp can not support the codec, call is dropped.

videoformat: Possible values are QCIF or CIF. This value is negotiated.
videowidth: Possible values are 176(for QCIF) or 352(for CIF)
videoheight: Possible values are 144(for QCIF) or 288(for CIF) .These values are interpreted using videoformat parameter.
videofps: Possible values are 15 or 30. This value depends on end device, and can be changed. This value is negotiated.
videobitrate: 1920 is default, this can be changed by end user according to network requirement. This value is negotiated.
videofarfmtp: "QCIF=2;CIF=3;MAXBR=1960".This depends on the far end's capabilities .This value represent what we get in Invite SDP. Hence comes from phone.
videonearfmtp: "QCIF=4".This comes from what MPP negotiate.

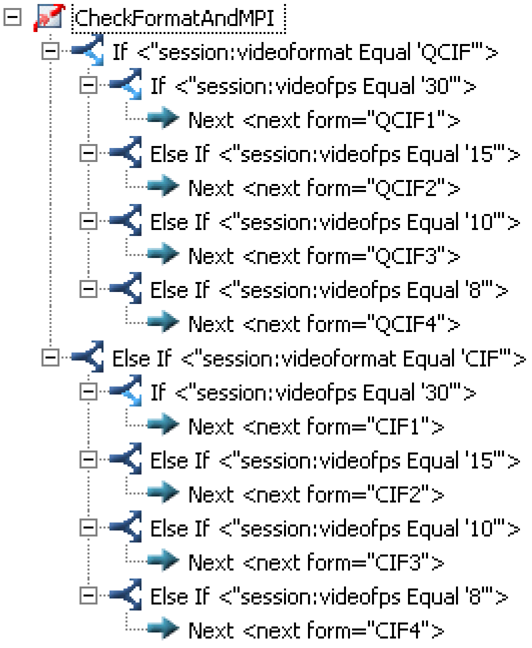
Application developer can use if/else statement in prompts to set different media pages according to different screen sizes. Thus based on the video format (CIF, QCIF etc), the prompt displays images and video text blocks suitable for that screen size.

As discussed earlier It is recommended that Video Screen format and MPI should match with the Application's 3gp files video format and fps, an application developer can check what is the session fps, and based on that can select a suitable media file to play.

Following figure shows a sample from DD application.



Following shows how one can do if/else to select a proper media file based on the session fps value.



```
CheckFormatAndMPI
    If <"session:videoformat Equal 'QCIF'">
        If <"session:videofps Equal '30'">
            Next <next form="QCIF1">
        Else If <"session:videofps Equal '15'">
            Next <next form="QCIF2">
        Else If <"session:videofps Equal '10'">
            Next <next form="QCIF3">
        Else If <"session:videofps Equal '8'">
            Next <next form="QCIF4">
    Else If <"session:videoformat Equal 'CIF'">
        If <"session:videofps Equal '30'">
            Next <next form="CIF1">
        Else If <"session:videofps Equal '15'">
            Next <next form="CIF2">
        Else If <"session:videofps Equal '10'">
            Next <next form="CIF3">
        Else If <"session:videofps Equal '8'">
            Next <next form="CIF4">
```

Refer to Dialog Designer documentation for more details on how to use session variables for the platform and in the simulator.