ESWIN

VPS Developer Manual

EIC7x series AI Digital SoC

Preliminary / Rev 0.7

Dec. 03, 2024

Notice

INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND DOES NOT REPRESENT A COMMITMENT ON THE PART OF ESWIN. UNLESS OTHERWISE SPECIFIED, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENT ARE PROVIDED "AS IS" WITHOUTH WARRANTIES, GUARANTEES, OR REPRESENTATIONS OF ANY KIND, EITHER EXPRESS OR IMPLIED.

"ESWIN" logo, and other ESWIN icons, are trademarks of Beijing ESWIN Computing Technology Co., Ltd. And(or) its parent company.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

If applicable, Open-Source Software and/or free software licensing notices are available in the product installation.

Change History

Version	Date	Descriptions
V0.6	Jun.26,2024	The initial version.
V0.6.1	Jul.25.2024	Revised VPS pixel format requirements.
V0.7	Aug.06.2024	Add description of stride for ES_VPS_CropResize API.
V0.7.1	Aug.13.2024	Revised VPS pixel format requirements.
V0.7.2	Sept.04.2024	Add description of Normalization parameters
V0.7.3	Oct.19.2024	Add content related to VPS Property
V0.7.4	Dec.03.2024	Add chapter of VPS Proc
V0.7.5	Apr.28.2025	2.4.3 Revised the VPS pixel format support matrix and updated the format supported by normalization, and vse added support for B8G8R8 output. 3.4 Updates the format supported by normalization

Content

Content of Tables

Content of Figures

1. Overview

The Video Process System (VPS) is a video image processing system that offers image processing functions in a time-sharing multiplexing mode. These functions include fisheye correction, lens distortions correction, cropping, scaling, color space conversion, fixed angle rotation, MIRROR and FLIP operations, line drawing, solid color fill, normalization, frame rate control, OVERLAY, aspect ratio and other functions.

VPS has no dedicated hardware, but based on DWE (De-Warp Engine), VSE (Video Scale Engine), HAE (Hardware Accelerated Engine) and other hardware, to achieve the image processing function of the software driver, support single frame and link two processing methods.

- Single: the image processing is called synchronously until it is complete.
- Link: complex operations are performed asynchronously based on the combination of multiple hardware components configured by users.

2. Function Description

2.1 Basic Concepts

- GROUP: The VPS provides the group concept for users. The maximum number is ES_VPS_MAX_GRP_NUM. The hardware used by VPS is multiplexed by various groups in time division multiplexing (TDM) mode, the hardware processes the tasks submitted by each group in sequence.
- CHANNEL: Channel of the VPS group. In VPS, channels generally correspond to data outputs. Both GROUP and CHANNEL have the crop function. The difference is that the crop of GROUP will affect all channels, while the crop of CHANNEL will only affect its own channel independently. The maximum number is ES_VPS_MAX_CHN_NUM.
- Frame rate control(FRC): There are two kinds of frame rate control in VPS: group frame rate control and channel frame rate control.
  - Group frame rate control: The group frame rate of receiving input images is controlled.
  - Channel frame rate control: The channel frame rate is used to control the image processing of each channel.
- CROP: The cropping modes include group cropping and channel cropping in VPS.
  - Group cropping: The VPS crops input images.
  - Channel cropping: The VPS crops the output image of each channel.
- COLOR SPACE CONVERSION(CSC): VPS supports format conversion of input and output images, such as from some RGB format to some YUV format.
- SCALE: The VPS can zoom out or zoom in on the image.
- MIRROR/FLIP: Mirror refers to horizontal mirroring, and flip refers to up and down inversion. Mirror and flip can be combined to achieve 180° rotation.
- ROTATION by a Fixed Angle: Rotate the input image 90°, 180°, 270° in a clockwise direction.
- OVERLAY: The VPS OVERLAY supports line drawing, rectangle drawing, and bitmap overlay. It is also divided into two types: group OVERLAY and channel OVERLAY.
  - Group OVERLAY: VPS OVERLAY on input image.
  - Channel OVERLAY: VPS OVERLAY the output image of each channel.
- ASPECT RATIO: The aspect ratio specifies the ratio of width and height of the input image on the target canvas, the coordinates, and the color filling outside the rectangle.
- NORMALIZATION: The VPS normalizes the input image.
- DEWARP: The VPS image distortions correction includes fisheye correction and fisheye expand function.

2.2 Function Implementation

The position of VPS in the entire ESSDK is shown in the figure below.

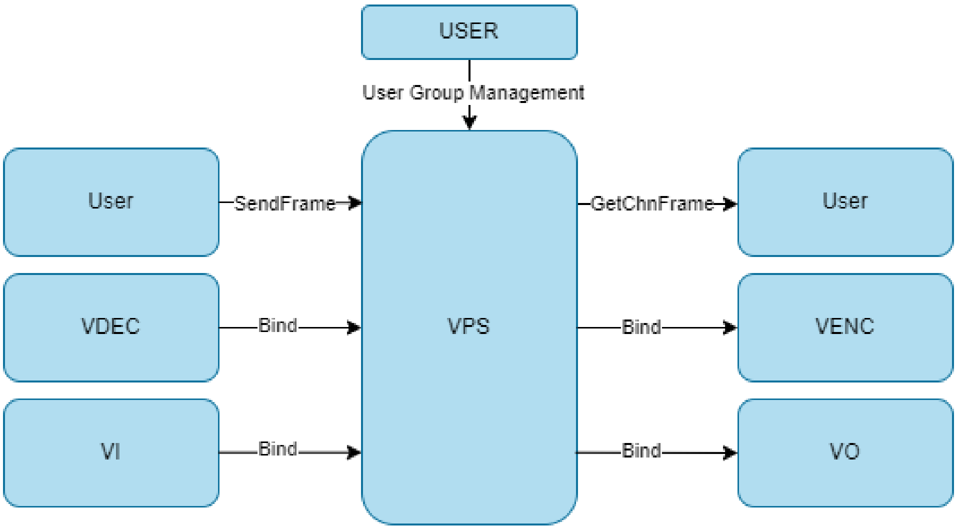


Figure 2-1 Position of VPS in ESSDK

By calling the binding interface of the System module, the user can bind other modules to the VPS. When VPS is the input source, VO and VENC can be bound to it as the receiver. When VPS is the receiver, VDEC and VI can be bound to it as input sources.

2.3 Processing Workflow

The VPS data processing flow is shown in the figure.

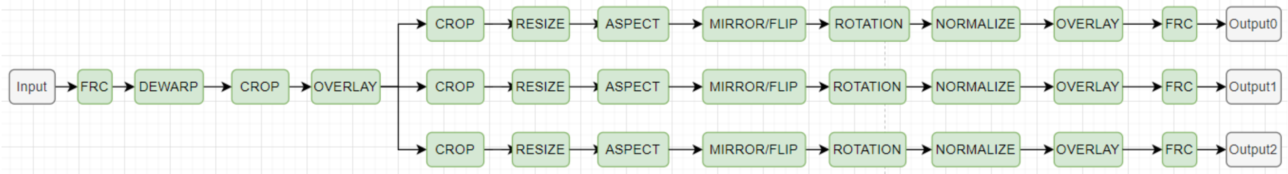


Figure 2-2 VPS longest link figure

The figure above shows the longest link of the VPS in link mode and the sequence of each operation.

- Users can set GROUP and CHANNEL parameters and attributes based on service requirements.
- The user can specify a hardware to perform an operation based on the actual situation.
- When the combination conditions are met, the VPS automatically merges adjacent operations belonging to the same hardware and submits them to the hardware at one time to improve performance.
- Normalization and OVERLAY cannot exist simultaneously in the same channel.

2.4 Hardware Limitation

The VPS uses DWE, VSE, HAE and other different hardware for image processing, different hardware has different capabilities and functions, the following is a brief introduction.

2.4.1 Alignment requirement

Table 2-1 VPS hardware alignment requirements

HW	IN/OUT	Width-Align	Height-Align
DWE	IN/OUT	16	8
VSE	IN/OUT	2	2
HAE	IN/OUT	64	2

- The alignment units are pixels.
- HAE supports the format of the RGB domain, and the alignment of RGB input and output are both 1 pixel.
- Only VSE can be used for one in multiple out feature.
- Only HAE can be used for multiple sources blit feature.
- It is recommended to use HAE for rotation, MIRROR, FLIP to get better performance.

2.4.2 Capability Matrix

Table 2-2 VPS capability matrix

ITEM	DWE	VSE	HAE	Work Mode
Maximum resolution	input：4096x3072 output：3840x2160	input：4096x2160 output： channel 0：3840x2160 channel 1：1920x1088 channel 2：1920x1088	7680x3840	
Minimum resolution	640x480	64x32	64x16	
One In Multiple Out	No	Yes	No	Single/Link
Multiple Source Blit	No	No	Yes	Single
Resize	Only upscale is supported and does not exceed 3840x2160	Yes	Yes	Single/Link
Crop	Yes	Yes	Yes	Single/Link
Alpha Blend	No	No	Yes	Single
Overlay	No	No	Yes	Link
Rotation	Yes	No	Yes	Single/Link
Mirror/Flip	Yes	No	Yes	Link
Aspect	No	No	Yes	Link
Normalization	No	No	Yes	Single/Link
Dewarp	Yes	No	No	Single/Link
CSC	Yes	Yes	Yes	Single/Link
Fill	No	No	Yes	Single
Line	No	No	Yes	Single/Link

2.4.3 Pixel format requirement

- Only the small-end mode is supported. In the format ending with BE, the significant bit is high; in the format ending with LE, the significant bit is low.
- General in HAE output refers to zoom in, zoom out, format conversion, FLIP, etc.
- OSD in HAE output refers to AlphaBlending, where two diagrams are superimposed and the format requirements of the target diagram are specified.
- VSE can output only one path in RGB format, and output multiple RGB channels at the same time.
- “Y” indicates support, and “-” indicates no support.

Table 2-3 VPS pixel format requirements

Pixel Format	HAE			Normalization		Line	Fill	DWE		VSE	
	Input	Output		Input	Output	Input	Input	Input	output	Input	output
		Normal	OSD								
PIXEL_FORMAT_A8R8G8B8	Y	Y	Y	Y	Y	Y	Y	-	-	-	-
PIXEL_FORMAT_A8B8G8R8	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_R8G8B8	Y	Y	Y	Y	Y	Y	Y	-	-	-	Y
PIXEL_FORMAT_B8G8R8_PLANAR	-	-	-	-	Y	-	-	-	-	-	Y
PIXEL_FORMAT_B8G8R8I	-	-	-	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_B8G8R8I_PLANAR	-	-	-	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_B16G16R16I	-	-	Y	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_B16G16R16I_PLANAR	-	-	-	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_B32G32R32F	-	-	Y	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_B32G32R32F_PLANAR	-	-	-	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_B16G16R16F	-	-	Y	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_B16G16R16F_P	-	-	-	-	Y	-	-	-	-	-	-

Pixel Format	HAE			Normalization		Line	Fill	DWE		VSE	
	Input	Output		Input	Output	Input	Input	Input	output	Input	output
		Normal	OSD								
LANAR											
PIXEL_FORMAT_GRAY16F	-	-	-	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_GRAY32F	-	-	-	-	Y	-	-	-	-	-	-
PIXEL_FORMAT_A2R10G10B10	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_A2B10G10R10	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_YUY2	Y	Y	Y	-	-	Y	Y	Y	Y	Y	Y
PIXEL_FORMAT_YVYU	Y	Y	Y	-	-	Y	Y	-	-	-	-
PIXEL_FORMAT_VYUY	Y	Y	Y	-	-	Y	Y	-	-	-	-
PIXEL_FORMAT_UYVY	Y	Y	Y	-	-	Y	Y	-	-	-	-
PIXEL_FORMAT_NV16	Y	Y	-	-	-	-	Y	Y	Y	Y	Y
PIXEL_FORMAT_NV61	Y	Y	-	-	-	-	Y	-	-	-	-
PIXEL_FORMAT_I420	Y	Y	-	Y	-	-	Y	-	-	-	-
PIXEL_FORMAT_YV12	Y	Y	-	-	-	-	Y	-	-	-	-
PIXEL_FORMAT_NV12	Y	Y	-	Y	-	-	Y	Y	Y	Y	Y
PIXEL_FORMAT_NV21	Y	Y	-	Y	-	-	Y	-	-	-	-
PIXEL_FORMAT_GRAY8	-	Y	Y	-	-	Y	Y	-	-	-	-
PIXEL_FORMAT_YUV422I010LE	-	-	-	-	-	-	-	Y	Y	Y	Y
PIXEL_FORMAT_YUV422SP010LE	-	-	-	-	-	-	-	Y	Y	Y	Y

Pixel Format	HAE			Normalization		Line	Fill	DWE		VSE	
	Input	Output		Input	Output	Input	Input	Input	output	Input	output
		Normal	OSD								
PIXEL_FORMAT_R5G6B5	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_A4R4G4B4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_X1R5G5B5	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_X4R4G4B4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_X8R8G8B8	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_X8B8G8R8	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_R4G4B4X4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_R4G4B4A4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_R5G5B5X1	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_R5G5B5A1	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_R8G8B8X8	Y	Y	Y	Y	--	Y	Y	-	-	-	-
PIXEL_FORMAT_R8G8B8A8	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_R10G10B10A2	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_B4G4R4X4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_B4G4R4A4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_B5G5R5X1	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_B5G5R5A1	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT	Y	Y	Y	Y	-	Y	Y	-	-	-	-

Pixel Format	HAE			Normalization		Line	Fill	DWE		VSE	
	Input	Output		Input	Output	Input	Input	Input	output	Input	output
		Normal	OSD								
_B8G8R8X8											
PIXEL_FORMAT_B8G8R8A8	Y	Y	Y	Y	Y	Y	Y	-	-	-	-
PIXEL_FORMAT_B10G10R10A2	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_B5G6R5	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_A1R5G5B5	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_X1B5G5R5	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_A1B5G5R5	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_X4B4G4R4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_A4B4G4R4	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_A8	Y	Y	Y	Y	-	Y	Y	-	-	-	-
PIXEL_FORMAT_B8G8R8	Y	Y	Y	Y	Y	Y	Y	-	-	-	Y
PIXEL_FORMAT_YUV420P010LE	Y	Y	-	Y	-	-	Y	-	-	-	-
PIXEL_FORMAT_YUV420P010BE	Y	Y	-	Y	-	-	Y	-	-	-	-
PIXEL_FORMAT_YUV420SP010BE	Y	Y	-	Y	-	-	Y	-	-	-	-
PIXEL_FORMAT_YUV420SP010LE	Y	Y	-	Y	-	-	Y	Y	Y	Y	Y
PIXEL_FORMAT_YUV444P	Y	Y	-	Y	-	-	Y	-	-	-	Y

3. User Guide

There are two ways to use VPS, single and link. These two methods need to call ES_VPS_Init before use, ES_VPS_Deinit after use, and these two global APIs need to be called only once in the application lifecycle. When using the APIs, it is recommended to write the code strictly in accordance with the process of the user guide, and reasonably determine the return value of the function.

3.1 The Single Mode APIs Usage

Single APIs are synchronous interfaces, so calls are relatively simple.

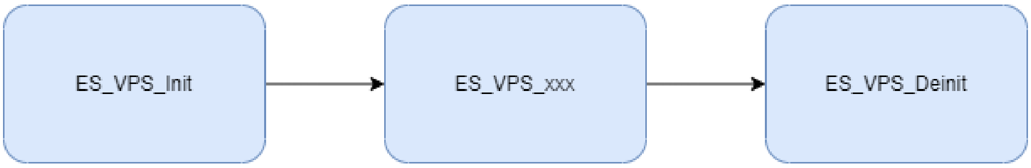


Figure 3-1 Single API call flow diagram

3.2 The Link Mode APIs Usage

Link APIs are asynchronous interfaces, more powerful, but also more complex to use. The following describes the use of link API in four scenarios: start, end, parameter update, and Overlay.

3.2.1 Start

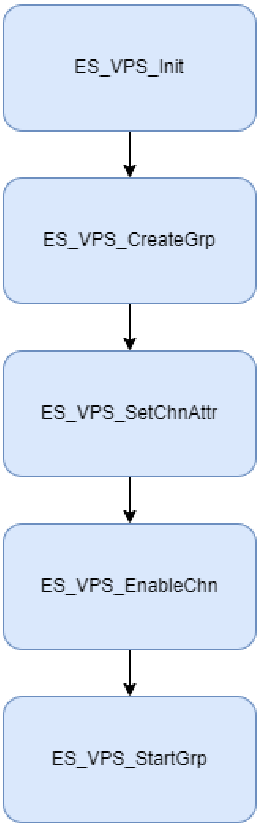


Figure 3-2 Link API start flow diagram

If you need to configure multiple outbound traffic, call ES_VPS_SetGrpMultiOutAttr before ES_VPS_EnableChn and ensure that the Enable status of the channels matches. Otherwise, an error message is displayed.

3.2.2 End

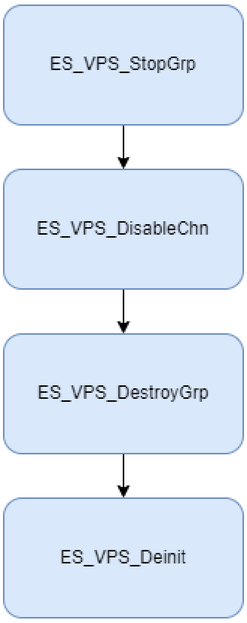


Figure 3-3 Link API end flow diagram

3.2.3 Parameter Update

When updating parameters, you are advised to Get parameters first and then Set parameters. Generally, the parameters of VPS cannot be dynamically updated. You need to Stop and then Start first. If Group doesn't have Start, you can directly update the parameters.

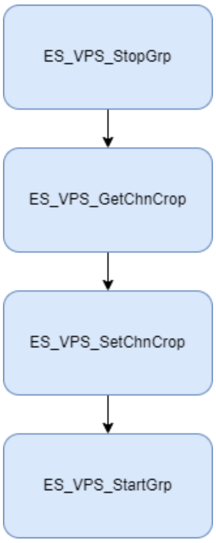


Figure 3-4 Flow diagram of updating link API parameters.

3.2.4 OVERLAY

VPS Overlay requires that VPS_OVERLAY_HANDLE be bound to a Group or Channel, and a Handle can be bound to multiple groups or channels.

Handle creation and release must be paired; otherwise, memory leaks may occur. The Attach operation for the Overlay must be before the Group Start and the Detach operation must be after the Group Stop. Attach and Detach must also be paired, or the Handle will fail to release.

Overlay layers support dynamic updating.

The following figure shows the process of attaching an Overlay Attach to a Group. If a user wants to Attach it to a Channel, just call ES_VPS_AttachChnOverlay.

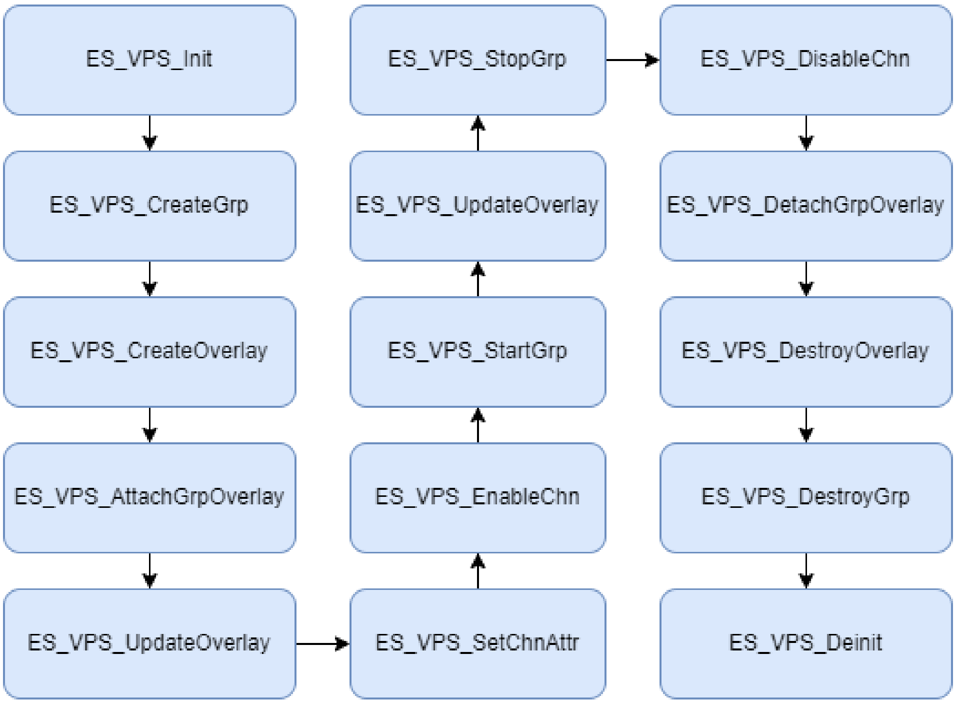


Figure 3-5 Link API Overlay usage flow diagram

3.3 Output Image

After the VPS processing is complete, the result is cached to the output queue of the corresponding channel. If the user or the later module doesn't remove the output image in time, it will cause a block, and the VPS will stop processing the image because there is no space to store the output image.

- Binding mode: In binding mode, the VPS is bound to the source end using the ES_SYS_Bind interface. In this case, the VPS automatically sends the processed image to the target. The target side should decide how to process the image in time according to its specific business, whether to receive or discard, and shouldn't block.
- User mode: In user mode, the user needs to obtain the output image through the ES_VPS_GetChnFrame interface. You are advised to obtain the FD of a channel through the ES_VPS_GetChnFd interface. Then, the epoll mechanism of the channel operating system listens for the FD. When FD is awakened, it means that the user can obtain the output image from this channel. At this time, the user calls ES_VPS_GetChnFrame interface to remove the image. After using the image, the user must call ES_VPS_ReleaseChnFrameinterface to release the memory space corresponding to the image. Otherwise, a lot of memory space will be wasted, and eventually memory will not be available elsewhere.

3.4 Normalization and Quantization

Normalization is a technique that scales data proportionally to fit within a smaller, specific interval, typically between 0 and 1 or -1 and 1. This technique is crucial for enhancing the performance of many algorithms as it eliminates the impact of differences in dimensions or magnitudes among various measurements.

VPS_NORMALIZATION_MIN_MAX: This refers to min-max normalization. It linearly transforms the original data into a specified interval using the formula:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Where X is the original data, X_{min} and X_{max} are the minimum and maximum values in the dataset, respectively, and X_{norm} is the normalized data.

VPS_NORMALIZATION_Z_SCORE: Also known as Z-score normalization or standard score normalization. This method transforms the data based on the mean (μ) and standard deviation (σ) of the original data using the formula:

$$\mathrm{Z} = \frac{X - \ \mu}{\sigma}$$

Where X is the original data, μ is the mean of the original data, σ is the standard deviation of the original data, and Z is the normalized data. This normalization technique transforms the data into a dataset with a mean of 0 and a standard deviation of 1, i.e., the data distribution becomes a standard normal distribution (or close to it).

Interfaces related to normalization and quantization are ES_VPS_Normalization, ES_VPS_SetChnNormalization, and ES_VPS_GetChnNormalization. They utilize the same structure, VPS_NORMALIZATION_PARAMS_S, where the data types for R, G, B are ES_U32. The difference between X_{max} and X_{min} must be provided in the form of its reciprocal, and the same applies to the standard deviation value.

To minimize precision loss within VPS and ensure consistency in hardware output, the interfaces must adhere to two key points:

- The data types for R, G, B cannot be ES_FLOAT. Users can resolve this by using C language's type casting.
- Users must calculate the reciprocals of the denominators in the transformation formulas. VPS performs only multiplication internally.

Below is an example code snippet for setting the R component, with similar logic applicable to G and B.

```
// min-max sample data
ES_FLOAT min = 100.0;    // The min value of red pixel training by model
ES_FLOAT max = 200.0;    // The max value of red pixel training by model
ES_FLOAT max_min_rec = 1.0 / (max - min);
// z-score sample data
ES_FLOAT mean = 98.5;    // The mean value of red pixel training by model
ES_FLOAT std = 100.3;    // The standard deviation value of red pixel training by model
ES_FLOAT std_rec = 1.0 / r_std;

PIXEL_FORMAT_E output_format;
ES_FLOAT step = 0.02;    // The step value training by model
ES_FLOAT step_rec = 1.0 / step_rec;

VPS_NORMALIZATION_PARAMS_S para;
memset(&para, 0, sizeof(para));
// Set them if mode is min-max
para.normalizationMode = VPS_NORMALIZATION_MIN_MAX;
para.minValue.r = *((ES_U32*)(&min));
para.minValue.g = …
para.minValue.b = …
para.maxMinReciprocal.r = *((ES_U32*)(&max_min_rec));
para.maxMinReciprocal.g = …
para.maxMinReciprocal.b = …
// Set them if mode is z-score
para.normalizationMode = VPS_NORMALIZATION_Z_SCORE;
para.meanValue.r = *((ES_U32*)(&mean));
para.meanValue.g = …
para.meanValue.b = …
para.stdReciprocal.r = *((ES_U32*)(&std_rec));
para.stdReciprocal.g = …
para.stdReciprocal.b = …
// For quantization
switch (output_format) {
    case PIXEL_FORMAT_R8G8B8:
    case PIXEL_FORMAT_A8R8G8B8:
    case PIXEL_FORMAT_B8G8R8:
    case PIXEL_FORMAT_B8G8R8_PLANAR:
    case PIXEL_FORMAT_B8G8R8I:
    case PIXEL_FORMAT_B8G8R8I_PLANAR:
    case PIXEL_FORMAT_B16G16R16I:
    case PIXEL_FORMAT_B16G16R16I_PLANAR:
    case PIXEL_FORMAT_B8G8R8A8:
        para.bByPassQuantization = ES_FALSE;
        break;
    case PIXEL_FORMAT_GRAY16F:
    case PIXEL_FORMAT_GRAY32F:
    case PIXEL_FORMAT_B16G16R16F:
    case PIXEL_FORMAT_B32G32R32F:
    case PIXEL_FORMAT_B16G16R16F_PLANAR:
    case PIXEL_FORMAT_B32G32R32F_PLANAR:
    para.bByPassQuantization = ES_TRUE;
        para.stepReciprocal = *((ES_U32*)(&step_rec));
        break;
    default:
        // Error. Output format not support by normalization
        break;
}
```

```
// VPS_NORMALIZATION_PARAMS_S is done!
```

3.5 CSC Array

This chapter describe the CSC table between YUV and RGB.

Only when the property type is VPS_PROPERTY_ARRAY_CSC_YUV_TO_RGB, and the color mode is VPS_MODE_USER_DEFINED or VPS_MODE_USER_DEFINED_CLAMP, these tables will be valid. If HW is 10bit, this table also need to expand to 10bit.

```
R = ((C00*Y + C01*U + C02*V) + D0)>>10
G = ((C10*Y + C11*U + C12*V) + D1)>>10
B = ((C20*Y + C21*U + C22*V) + D2)>>10
// The array contains the coefficients in the sequence:
// C00, C01, C02, C10, C11, C12, C20, C21, C22, D0, D1, D2.
// For example:
yuv2rgb[] = {
        1196, 0, 1724,
        1196,-192, -668,
        1196, 2200,0,
        -958720, 364288, -1202432
};
D0=1196*(-64) + 1724 *(-512)+512 = -958720
D1=1196*(-64)+(-192)*(-512)+(-668)*(-512)+512 = 364288
D2=1196*(-64)+ 2200*(-512) +512 = -1202432

// For BT601 calculation (for 8bit)
R = 1.164(Y-16) + 1.596(Cr-128)
G = 1.164(Y-16) - 0.391(Cb-128) - 0.813(Cr-128)
B = 1.164(Y-16) + 2.018(Cb-128)

// Fixed-point to speed calculate (for 10bit)
R = 1.164(Y-64) + 1.596(Cr-512)
   = (1.164(Y-64) + 1.596(Cr-512))*1024/1024
   = 1192*Y+                  1634*V + (-913047)
G = 1.164(Y-64) - 0.391(Cb-512) - 0.813(Cr-512)
   = (1.164(Y-64) - 0.391(Cb-512) - 0.813(Cr-512))*1024/1024
   = 1192*Y+(-400)*U+(-833)*V + 554958
B = 1.164(Y-64) + 2.018(Cb-512)
   = (1.164(Y-64) + 2.018(Cb-512))*1024/1024
   = 1192*Y+    2066*U+               + (-1134297)
// So the table is:
{1192,0,1634,1192,-400,-833,1192,2066,0,-913047,554958,-1134297}
```

Only when the property type is VPS_PROPERTY_ARRAY_CSC_RGB_TO_YUV, and the color mode is VPS_MODE_USER_DEFINED or VPS_MODE_USER_DEFINED_CLAMP, these tables will be valid. If HW is 10bit, this table also need to expand to 10bit.

```
Y = ((C00*R + C01*G + C02*B) + 512 + D0)>>10
U = ((C10*R + C11*G + C12*B) + 512 + D1)>>10
V = ((C20*R + C21*G + C22*B) + 512 + D2)>>10
// The array contains the coefficients in the sequence:
// C00, C01, C02, C10, C11, C12, C20, C21, C22, D0, D1, D2.
// For example:
rgb2yuv[] = {
        230, 594, 52,
        -125, -323, 448,
        448, -412, -36,
        66048, 524800,524800
};
// For BT601 calculation
Y=0.299R+0.587G+0.114B
U=Cb=0.564(B-Y)+128=0.499704B??0.168636R??0.331068G+128(for 10bit, it should 512)
V=Cr=0.713(R-Y)+128=0.499813R??0.418531G??0.081282B+128(for 10bit, it should 512)


// Fixed-point to speed calculate (for 10bit)
Y=(0.299R+0.587G+0.114B)*1024/1024=(306*R+601*G+117*B+0)>>10
U=Cb=0.564(B-Y)+512=0.499704B-0.168636R-0.331068G+512
                    =(-0.168636R-0.331068G+0.499704B+512)*1024/1024
                    =((-173)*R+(-339)*G+512*B+524288)>>10
V=Cr=0.713(R-Y)+128=0.499813R-0.418531G-0.081282B+512
                    =(0.499813R-0.418531G-0.081282B+512)*1024/1024
                    =512*R+(-429)*G+(-83)*B+52488
// So the table is:
{ 306,601,117,-173,-339,512,512,-429,-83,0,52488,52488}
```

4. API Description

VPS interfaces are divided into three categories: global API, single API, and link API.

Globally, there are the following APIs:

- ES_VPS_Init: Initialize VPS.
- ES_VPS_Deinit: Deinitialize VPS
- ES_VPS_SetModParam: Set the module parameter of VPS.
- ES_VPS_GetModParam: Get the module parameter of VPS.
- ES_VPS_SetProperty: Set the property of VPS.
- ES_VPS_GetProperty: Get the property of VPS.

Single has the following APIs:

- ES_VPS_CropResize: Crop and Resize.
- ES_VPS_Rotation: Rotation.
- ES_VPS_Line: Draw lines.
- ES_VPS_Fill: Fill.
- ES_VPS_CSC: Color space convert.
- ES_VPS_AlphaBlending: Alpha blend.
- ES_VPS_Normalization: Normalization.
- ES_VPS_MultiSourcesBlit: Multiple source blit.
- ES_VPS_Dewarp: Dewarp, crop, upscale.

The link has the following APIs:

- ES_VPS_CreateGrp: Create a VPS Group.
- ES_VPS_DestroyGrp: Destroy a VPS Group.
- ES_VPS_StartGrp: Start a VPS Group.

- ES_VPS_StopGrp: Stop a VPS Group.
- ES_VPS_ResetGrp: Reset a VPS Group.
- ES_VPS_GetGrpAttr: Get Group attribute.
- ES_VPS_SetGrpAttr: Set Group attribute.
- ES_VPS_SetGrpCrop: Set Group crop.
- ES_VPS_GetGrpCrop: Get Group crop.
- ES_VPS_SendFrame: User send frame data to VPS.
- ES_VPS_SetGrpDewarp: Set Group dewarp.
- ES_VPS_GetGrpDewarp: Get Group dewarp.
- ES_VPS_QueryGrpStatus: Query VPS Group status.
- ES_VPS_GetGrpMultiOutAttr: Get VPS multi output attribute.
- ES_VPS_SetGrpMultiOutAttr: Set VPS multi output attribute.
- ES_VPS_SetChnAttr: Set VPS channel attribute.
- ES_VPS_GetChnAttr: Get VPS channel attribute.
- ES_VPS_EnableChn: Enable a VPS channel.
- ES_VPS_DisableChn: Disable a VPS channel.
- ES_VPS_SetChnCrop: Set channel crop.
- ES_VPS_GetChnCrop: Get channel crop
- ES_VPS_SetChnNormalization: Set channel normalization.
- ES_VPS_GetChnNormalization: Get channel normalization.
- ES_VPS_SetChnRotation: Set channel rotation.
- ES_VPS_GetChnRotation: Get channel rotation.
- ES_VPS_GetChnFrame: User get a frame data from VPS channel.
- ES_VPS_ReleaseChnFrame: Release channel video frame data.
- ES_VPS_AttachVbPool: Attach a VB pool to channel.
- ES_VPS_DetachVbPool: Detach VB pool from channel.
- ES_VPS_GetChnFd: Get FD of VPS channel.
- ES_VPS_CreateOverlay: Create an overlay handle.
- ES_VPS_DestroyOverlay: Destroy an overlay handle.
- ES_VPS_AttachGrpOverlay: Attach an overlay handle to group.
- ES_VPS_DetachGrpOverlay: Detach an overlay handle to group.
- ES_VPS_AttachChnOverlay: Attach an overlay handle to channel.
- ES_VPS_DetachChnOverlay: Detach an overlay handle to channel.
- ES_VPS_UpdateOverlay: Update the overlay group displays attribute for special overlay handle.

4.1 Global API

ES_VPS_Init

【Function body】

ES_S32 ES_VPS_Init()

【Description】

Init VPS. When use VPS module, this must be the first interface called.

【Parameters】

Parameter Name	Descriptions	Input/Output

【Return】

Return Value	Descriptions

ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_Deinit

【Function body】

ES_S32 ES_VPS_Deinit()

【Description】

Deinit VPS. When not use VPS module anymore, this must be the last interface called.

【Parameters】

Parameter Name	Descriptions	Input/Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetModParam

【Function body】

ES_S32 ES_VPS_SetModParam(const VPS_MOD_PARAM_S *pModParam)

【Description】

Set the module parameter of VPS. Module parameters are valid for the entire VPS module, when call this interface please make sure no groups are created, or interface will return ES_ERR_VPS_NOT_PERM.

- If VB Source is common VB pool, we need to follow the rules of common VB pool. For example call ES_VB_SetConfig firstly, call ES_VB_Init secondly to initialize common pool.
- If VB Source is user VB pool, before Group Started, user need call ES_VPS_AttachVbPool firstly to attach VB pool to Group, otherwise VPS will stop working because can't get output buffers.

【Parameters】

Parameter Name	Descriptions	Input/Output
pModParam	The value of module parameter	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetModParam

【Function body】

ES_S32 ES_VPS_GetModParam(VPS_MOD_PARAM_S *pModParam)

【Description】

Get the module parameter of VPS.

【Parameters】

Parameter Name	Descriptions	Input/Output
pModParam	The value of module parameter	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetProperty

【Function body】

ES_S32 ES_VPS_SetProperty(const VPS_PROPERTY_S *pVpsProperty)

【Description】

Set VPS property. Please initialize VPS first. The property set is bind with the thread call this interface. For different threads, even the same properties user needs to set again. After property set, the actual execution of other single interfaces will be affected, however link mode will not be affected by this interface.

- VPS_PROPERTY_RESIZE_METHOD: After this property set, all single interfaces contain resize feature will be affected, VPS will execute resize by the method user set.
- VPS_PROPERTY_ARRAY_CSC_YUV_TO_RGB, VPS_PROPERTY_ARRAY_CSC_RGB_TO_YUV: After this property set, all single interfaces contain CSC will be affected, VPS will execute CSC by the method user set.
- VPS_PROPERTY_YUV_COLOR_MODE_SRC, VPS_PROPERTY_YUV_COLOR_MODE_DST: After this property set, all single interfaces contain YUV color mode will be affected, VPS will affect the YUV color mode of source or destination image.
- VPS_PROPERTY_HAE_MULTICORE_MODE: After this property set, all single interfaces

with HAE will be affected, the HAE work mode will be changed to user set.

The default resize method of VPS is VPS_RESIZE_STRETCHBLIT, if user want to use a new method VPS_RESIZE_BICUBIC to do resize, user need to call ES_VPS_SetProperty interface, then call ES_VPS_CropResize interface.

【Parameters】

Parameter Name	Descriptions	Input/Output
pVpsProperty	The property of VPS	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetProperty

【Function body】

ES_S32 ES_VPS_GetProperty(VPS_PROPERTY_S *pVpsProperty)

【Description】

Get VPS property. Please initialize VPS first. The property get is bind with the thread call this interface. Different threads may get the different properties.

【Parameters】

Parameter Name	Descriptions	Input/Output
pVpsProperty	The property of VPS	Input/Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

4.2 Single Mode API

ES_VPS_CropResize

【Function body】

```
ES_S32 ES_VPS_CropResize (
              VIDEO_FRAME_S *pSrc,
              VIDEO_FRAME_S *pDst,
              RECT_S *pSrcRect,
              RECT_S *pDstRect,
              ES_S32 outputFlag,
              HWTYPE_E type)
```

【Description】

Crop and Resize. VPS must be initialized before using. When type is HW_TYPE_DWE or HW_TYPE_VSE, pDstRect is not supported. When output is YUV format, make sure the stride is 64 or above.

【Parameters】

Parameter Name	Descriptions	Input/Output
pSrc	The source image.	Input
pDst	The destination images, the count is defined by outputFlag.	Input/Output
pSrcRect	The rect of source surface.	Input
pDstRect	The rect of destination surface.	Input
outputFlag	The output flag. Output to channel 0: 0x0 or 0x1. Output to channel 1: 0x2. Output to channel 2: 0x4 Output to channel 1 and 2: 0x6	Input
type	Appoint the hardware type to execute crop resize work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_Rotation

【Function body】

```
ES_S32 ES_VPS_Rotation (
        VIDEO_FRAME_S *pSrc,
        VIDEO_FRAME_S *pDst,
        ROTATION_E rot,
        HWTYPE_E type)
```

【Description】

Rotation.

【Parameters】

Parameter Name	Descriptions	Input/Output
pSrc	The source image.	Input
pDst	The destination image.	Input/Output
rot	Support rotation angle: ROTATION_90, ROTATION_180, ROTATION_270.	Input
type	Appoint the hardware type to execute rotation work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_Line

【Function body】

```
ES_S32 ES_VPS_Line (
        VIDEO_FRAME_S *pDst,
        ES_U32 color,
        POINT_S *pStart,
        POINT_S *pEnd,
        ES_S32 lineCount,
        HWTYPE_E type)
```

【Description】

Draw lines.

【Parameters】

Parameter Name	Descriptions	Input/Output
pDst	The destination image.	Input/Output
color	The line ARGB color.	Input
pStart	Line start point, it is a pointer array.	Input
pEnd	Line end point, it is a pointer array.	Input
lineCount	Line count	Input
type	Appoint the hardware type to execute draw line work.	Input

【Return】

Return Value	Descriptions

ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_Fill

【Function body】

ES_S32 ES_VPS_Fill (

VIDEO_FRAME_S *pDst,
ES_U32 *pColor,
ES_S32 colorCnt,
RECT_S *pRect,
ES_S32 rectCount,
HWTYPE_E type)

【Description】

Fill.

【Parameters】

Parameter Name	Descriptions	Input/Output
pDst	The destination image.	Input/Output
pColor	Fill the color, ARGB format, it is an array of Pointers.	Input
colorCnt	The number of colors.	Input
pRect	Destination surface rect where will be filled. It is a pointer array.	Input
rectCount	Rect count	Input
type	Appoint the hardware type to execute fill work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_CSC

【Function body】

ES_S32 ES_VPS_CSC (

VIDEO_FRAME_S *pSrc,
VIDEO_FRAME_S *pDst,
HWTYPE_E type)

【Description】

Color space convertation.

【Parameters】

Parameter Name	Descriptions	Input/Output
pSrc	The source image with source format.	Input
pDst	The destination image with target format.	Input/Output
type	Appoint the hardware type to execute CSC work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_AlphaBlending

【Function body】

```
ES_S32 ES_VPS_AlphaBlending (
            VIDEO_FRAME_S *pSrc,
            VIDEO_FRAME_S *pDst,
            VPS_BLEND_MODE_E blendMod,
            ES_S32 srcGlobalAlpha,
            ES_S32 dstGlobalAlpha,
            HWTYPE_E type)
```

【Description】

Alpha blend.

【Parameters】

Parameter Name	Descriptions	Input/Output
pSrc	The source image with source format.	Input
pDst	The destination image with target format.	Input/Output
blendMod	Set the blend mode.	Input
srcGlobalAlpha	Set source surface global alpha value.	Input
dstGlobalAlpha	Set destination surface global alpha value.	Input
type	Appoint the hardware type to execute CSC work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_Normalization

【Function body】

ES_S32 ES_VPS_Normalization (

VIDEO_FRAME_S *pSrc,
VIDEO_FRAME_S *pDst,
VPS_NORMALIZATION_PARAMS_S *pParams,
RECT_S *pSrcRect,
RECT_S *pDstRect)

【Description】

Normalization.

【Parameters】

Parameter Name	Descriptions	Input/Output
pSrc	The source image with source format.	Input
pDst	The destination image with target format.	Input/Output
pParams	Normalization parameter, support MIN_MAX, Z_SCORE mode.	Input
pSrcRect	Source surface rect.	Input
pDstRect	Destination surface rect.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_MultiSourcesBlit

【Function body】

ES_S32 ES_VPS_MultiSourcesBlit (

VIDEO_FRAME_S *pSrcs,
ES_S32 srcCount,
RECT_S *pSrcRects,
ROTATION_E *pRot,
RECT_S *pDstRects,

VIDEO_FRAME_S *pDst,
HWTYPE_E type)

【Description】

Multiple sources blit. Only HAE support this feature. Blit one or more source images into one destination image.

【Parameters】

Parameter Name	Descriptions	Input/Output
pSrcs	The source image with source Format.	Input
srcCount	The source image count.	Input
pSrcRects	The rect to be selected of source image.	Input
pRot	Rotation angle: ROTATION_90, ROTATION_180, ROTATION_270. It is a pointer array.	Input
pDstRects	The rect which source image locate at destination image.	Input
pDst	The destination image with target format.	Input/Output
type	Appoint the hardware type to execute CSC work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_Dewarp

【Function body】

ES_S32 ES_VPS_Dewarp (

VIDEO_FRAME_S *pSrc,
VIDEO_FRAME_S *pDst,
const VPS_DEWARP_PARAMS_S *pParams)

【Description】

Dewarp, crop, upscale.

【Parameters】

Parameter Name	Descriptions	Input/Output
pSrc	The source image with source Format.	Input
pDst	The destination image with target format.	Input/Output
pParams	Dewarp parameter.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

4.3 Link Mode API

ES_VPS_CreateGrp

【Function body】

```
ES_S32 ES_VPS_CreateGrp (
                VPS_GRP vpsGrp,
                DIE_IDX nId,
                const VPS_GRP_ATTR_S *pGrpAttr)
```

【Description】

Create a VPS Group within the specified DIE.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
nId	Hardware DIE id. Value range[0, 1].	Input
pGrpAttr	VPS group attribute.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_DestroyGrp

【Function body】

```
ES_S32 ES_VPS_DestroyGrp (VPS_GRP vpsGrp)
```

【Description】

Destroy a VPS Group.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_StartGrp

【Function body】

ES_S32 ES_VPS_StartGrp (VPS_GRP vpsGrp)

【Description】

Start a VPS Group.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_StopGrp

【Function body】

ES_S32 ES_VPS_StopGrp (VPS_GRP vpsGrp)

【Description】

Stop a VPS Group.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_ResetGrp

【Function body】

ES_S32 ES_VPS_ResetGrp (VPS_GRP vpsGrp)

【Description】

Reset a VPS Group.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetGrpAttr

【Function body】

```
ES_S32 ES_VPS_SetGrpAttr (
            VPS_GRP vpsGrp,
            const VPS_GRP_ATTR_S *pGrpAttr)
```

【Description】

Set Group attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pGrpAttr	VPS group attribute.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetGrpAttr

【Function body】

ES_S32 ES_VPS_GetGrpAttr (

VPS_GRP vpsGrp,
VPS_GRP_ATTR_S *pGrpAttr)

【Description】

Get Group attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pGrpAttr	VPS group attribute.	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetGrpCrop

【Function body】

ES_S32 ES_VPS_SetGrpCrop (

VPS_GRP vpsGrp,
const VPS_CROP_INFO_S *pCropInfo,
HWTYPE_E type)

【Description】

Set Group crop attributes.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input

Parameter Name	Descriptions	Input/Output
pCropInfo	VPS crop info.	Input
type	Appoint the hardware type to execute crop work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetGrpCrop

【Function body】

```
ES_S32 ES_VPS_GetGrpCrop (
            VPS_GRP vpsGrp,
            VPS_CROP_INFO_S *pCropInfo,
            HWTYPE_E *pType)
```

【Description】

Get Group crop attributes.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pCropInfo	VPS crop info.	Output
pType	Hardware type.	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SendFrame

【Function body】

```
ES_S32 ES_VPS_SendFrame (
            VPS_GRP vpsGrp,
            const VIDEO_FRAME_INFO_S *pVideoFrame,
```

ES_S32 milliSec)

【Description】

User send frame data to VPS.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pVideoFrame	Video frame data.	Input
milliSec	Timeout value. Time unit is ms. -1: Block wait. 0: Not wait. >0: Timeout value.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetGrpDewarp

【Function body】

```
ES_S32 ES_VPS_SetGrpDewarp (
            VPS_GRP vpsGrp,
            const VPS_DEWARP_PARAMS_S *pDewarp)
```

【Description】

Set Group dewarp attributes.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pDewarp	VPS dewarp parameter.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetGrpDewarp

【Function body】

```
ES_S32 ES_VPS_GetGrpDewarp (
                VPS_GRP vpsGrp,
                VPS_DEWARP_PARAMS_S *pDewarp)
```

【Description】

Get Group dewarp attributes.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pDewarp	VPS dewarp parameter.	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_QueryGrpStatus

【Function body】

```
ES_S32 ES_VPS_QueryGrpStatus (
                VPS_GRP vpsGrp,
                VPS_QUERY_STATUS_S *pQuery)
```

【Description】

Query VPS group processing status.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pQuery	VPS group query status.	Output

【Return】

Return Value	Descriptions

ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetGrpMultiOutAttr

【Function body】

ES_S32 ES_VPS_SetGrpMultiOutAttr (

VPS_GRP vpsGrp,
const VPS_MULTI_OUT_ATTR_S *pAttr)

【Description】

Set one in multi out attribute. If user want to use one group in and multiple channels out feature, this function is mandatory. If only one group in and one channel out, this function is optional. Once use this function, user need to make sure the channel enable status is same as the enable status in VPS_MULTI_OUT_ATTR_S. Otherwise StartGrp will be failed.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
pAttr	VPS Multi out attribute	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetGrpMultiOutAttr

【Function body】

ES_S32 ES_VPS_GetGrpMultiOutAttr (

VPS_GRP vpsGrp,
VPS_MULTI_OUT_ATTR_S *pAttr)

【Description】

Get One In Multiple Out attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input

Parameter Name	Descriptions	Input/Output
pAttr	VPS Multi out attribute	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetChnAttr

【Function body】

ES_S32 ES_VPS_SetChnAttr (

VPS_GRP vpsGrp,
VPS_CHN vpsChn,
const VPS_CHN_ATTR_S *pChnAttr)

【Description】

Set channel attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pChnAttr	VPS channel attribute	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetChnAttr

【Function body】

ES_S32 ES_VPS_GetChnAttr (

VPS_GRP vpsGrp,
VPS_CHN vpsChn,
VPS_CHN_ATTR_S *pChnAttr)

【Description】

Get channel attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pChnAttr	VPS channel attribute	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_EnableChn

【Function body】

```
ES_S32 ES_VPS_EnableChn (
                VPS_GRP vpsGrp,
                VPS_CHN vpsChn)
```

【Description】

Enable channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_DisableChn

【Function body】

ES_S32 ES_VPS_DisableChn (

VPS_GRP vpsGrp,
VPS_CHN vpsChn)

【Description】

Disable channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetChnCrop

【Function body】

ES_S32 ES_VPS_SetChnCrop (

VPS_GRP vpsGrp,
VPS_CHN vpsChn,
const VPS_CROP_INFO_S *pCropInfo,
HWTYPE_E type)

【Description】

Set channel crop attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pCropInfo	VPS crop information	Input

Parameter Name	Descriptions	Input/Output
type	Appoint the hardware type to execute crop work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetChnCrop

【Function body】

```
ES_S32 ES_VPS_GetChnCrop (
            VPS_GRP vpsGrp,
            VPS_CHN vpsChn,
            VPS_CROP_INFO_S *pCropInfo,
            HWTYPE_E *pType)
```

【Description】

Get channel crop attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pCropInfo	VPS crop information	Output
pType	Appoint the hardware type to execute crop work.	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_SetChnNormalization

【Function body】

```
ES_S32 ES_VPS_SetChnNormalization (
            VPS_GRP vpsGrp,
```

VPS_CHN vpsChn,
const VPS_NORMALIZATION_INFO_S *pNormInfo)

【Description】

Set channel normalization attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pNormalInfo	VPS normalization information	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetChnNormalization

【Function body】

ES_S32 ES_VPS_GetChnNormalization (

VPS_GRP vpsGrp,
VPS_CHN vpsChn,
VPS_NORMALIZATION_INFO_S *pNormInfo)

【Description】

Get channel normalization attribute.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pNormalInfo	VPS normalization information	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success

Others	Fail, refers to Error Code

ES_VPS_SetChnRotation

【Function body】

```
ES_S32 ES_VPS_SetChnRotation (
            VPS_GRP vpsGrp,
            VPS_CHN vpsChn,
            ROTATION_E rotation,
            HWTYPE_E type)
```

【Description】

Set channel rotation attributes.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
rotation	VPS rotation angle.	Input
type	Appoint the hardware type to execute rotation work.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetChnRotation

【Function body】

```
ES_S32 ES_VPS_GetChnRotation (
            VPS_GRP vpsGrp,
            VPS_CHN vpsChn,
            ROTATION_E *pRotation,
            HWTYPE_E *pType)
```

【Description】

Get channel rotation attributes.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pRotation	VPS rotation angle.	Output
pType	Hardware type.	Output

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetChnFrame

【Function body】

```
ES_S32 ES_VPS_GetChnFrame (
                VPS_GRP vpsGrp,
                VPS_CHN vpsChn,
                VIDEO_FRAME_INFO_S *pVideoFrame,
                ES_S32 milliSec)
```

【Description】

User get a frame data from VPS channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pVideoFrame	Video frame data.	Output
milliSec	Timeout value. Time unit is ms. -1: Block wait. 0: Not wait. >0: Timeout value.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success

Others	Fail, refers to Error Code

ES_VPS_ReleaseChnFrame

【Function body】

ES_S32 ES_VPS_ReleaseChnFrame (

VPS_GRP vpsGrp,
VPS_CHN vpsChn,
const VIDEO_FRAME_INFO_S *pVideoFrame)

【Description】

Release channel video frame data.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
pVideoFrame	Video frame data.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_AttachVbPool

【Function body】

ES_S32 ES_VPS_AttachVbPool (

VPS_GRP vpsGrp,
VPS_CHN vpsChn,
VB_POOL vbPool)

【Description】

Attach a VB pool to Channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input

Parameter Name	Descriptions	Input/Output
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
vbPool	VB pool ID.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_DetachVbPool

【Function body】

```
ES_S32 ES_VPS_DetachVbPool (
                VPS_GRP vpsGrp,
                VPS_CHN vpsChn)
```

【Description】

Detach VB pool from Channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_GetChnFd

【Function body】

```
ES_S32 ES_VPS_GetChnFd (
                VPS_GRP vpsGrp,
                VPS_CHN vpsChn)
```

【Description】

Get FD of VPS channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input

【Return】

Return Value	Descriptions
>= 0	The FD of channel
Others	Fail

ES_VPS_CreateOverlay

【Function body】

VPS_OVERLAY_HANDLE ES_VPS_CreateOverlay ()

【Description】

Create an overlay handle.

【Parameters】

Parameter Name	Descriptions	Input/Output

【Return】

Return Value	Descriptions
VPS_OVERLAY_HANDLE	The handle of overlay.
ES_NULL	Fail.

ES_VPS_DestroyOverlay

【Function body】

ES_S32 ES_VPS_DestroyOverlay (const VPS_OVERLAY_HANDLE handle)

【Description】

Destroy an overlay handle.

【Parameters】

Parameter Name	Descriptions	Input/Output
handle	VPS overlay handle.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_AttachGrpOverlay

【Function body】

```
ES_S32 ES_VPS_AttachGrpOverlay (
                VPS_GRP vpsGrp,
                const VPS_OVERLAY_HANDLE handle)
```

【Description】

Attach an overlay handle to group.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
handle	VPS overlay handle.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_DetachGrpOverlay

【Function body】

```
ES_S32 ES_VPS_DetachGrpOverlay (
                VPS_GRP vpsGrp,
                const VPS_OVERLAY_HANDLE handle)
```

【Description】

Detach an overlay handle to group.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
handle	VPS overlay handle.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_AttachChnOverlay

【Function body】

```
ES_S32 ES_VPS_AttachChnOverlay (
            VPS_GRP vpsGrp,
            VPS_CHN vpsChn,
            const VPS_OVERLAY_HANDLE handle)
```

【Description】

Attach an overlay handle to channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
handle	VPS overlay handle.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_DetachChnOverlay

【Function body】

```
ES_S32 ES_VPS_DetachChnOverlay (
VPS_GRP vpsGrp,
VPS_CHN vpsChn,
const VPS_OVERLAY_HANDLE handle)
```

【Description】

Detach an overlay handle to channel.

【Parameters】

Parameter Name	Descriptions	Input/Output
vpsGrp	VPS group ID. Value range[0, ES_VPS_MAX_GRP_NUM).	Input
vpsChn	VPS channel ID. Value range[0, ES_VPS_MAX_CHN_NUM).	Input
handle	VPS overlay handle.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

ES_VPS_UpdateOverlay

【Function body】

```
ES_S32 ES_VPS_UpdateOverlay (
        const VPS_OVERLAY_HANDLE handle,
        const VPS_OVERLAY_GROUP_S *pOverlays)
```

【Description】

Update the overlay group displays attribute for special overlay handle.

【Parameters】

Parameter Name	Descriptions	Input/Output
handle	VPS overlay handle.	Input
pOverlays	VPS overlay group attribute.	Input

【Return】

Return Value	Descriptions
ES_SUCCESS	Success
Others	Fail, refers to Error Code

5. Data Type

VPS related data types are defined as follows:

- ES_VPS_MAX_GRP_NUM: Define the max count of VPS group.
- ES_VPS_MAX_CHN_NUM: Define the max count of VPS channel.
- VPS_MIN_LINE_WIDTH: Define the min width of line.
- VPS_MAX_LINE_WIDTH: Define the max width of line.
- VPS_OVERLAY_MAX_LINE_NUM: Define the max number of lines.
- VPS_OVERLAY_MAX_QUADRANGLE_NUM: Define the max number of quadrangles.
- VPS_OVERLAY_MAX_DISPLAY_NUM: Define the max display number of overlay.
- VPS_GRP: Define VPS Group ID.
- DIE_IDX: Define the DIE index.
- VPS_CHN: Define VPS Channel ID.
- VPS_OVERLAY_HANDLE: Define the overlay handle.
- VPS_MOD_PARAM_S: Define the module parameter of VPS.
- HWTYPE_E: Define the hardware type of VPS engine.
- ROTATION_E: Define the rotation angle.
- VPS_BLEND_MODE_E: Define the blend mode.
- VPS_NORMALIZATION_MODE_E: Define the normalization mode.
- VPS_RGB_S: Define the RGB value.
- VPS_NORMALIZATION_PARAMS_S: Define the normalization parameters.
- VPS_DEWARP_MODEL_E: Define the dewarp model.
- VPS_DEWARP_DISTORTION_MAP_S: Define the dewarp distortion map.
- VPS_DEWARP_FOV_SETTING_S: Define the dewarp FOV setting.
- VPS_DEWARP_PARAMS_S: Define the VPS dewarp parameter.
- VPS_GRP_ATTR_S: Define the VPS GROUP attribute.
- VPS_CROP_COORDINATE_E: Define VPS crop coordinate mode.
- VPS_CROP_INFO_S: Define VPS crop information.
- VPS_CHN_FRAME_STATUS_S: Define the frame status of VPS channel.
- VPS_QUERY_STATUS_S: Define VPS query status.
- VPS_OUT_ATTR_S: Define the output attribute of channel.
- VPS_MULTI_OUT_ATTR_S: Define VPS multi output attribute.
- VPS_CHN_ATTR_S: Define VPS Channel attribute.
- VPS_NORMALIZATION_INFO_S: Define VPS normalization information.
- VPS_OVERLAY_DISP_LINE_S: Define the attribute of line.
- VPS_OVERLAY_DISP_LINE_GRP_S: Define the overlay line group.
- VPS_OVERLAY_DISP_QUADRANGLE_S: Define the attribute of quadrangle.
- VPS_OVERLAY_TYPE_E: Define the overlay type.
- VPS_OVERLAY_DISP_QUADRANGLE_GRP_S: Define the overlay quadrangle group.
- VPS_MOSAIC_BLK_SIZE_E: Define the block size of mosaic.
- VPS_OVERLAY_DISP_MOSAIC_S: Define the overlay mosaic.
- VPS_OVERLAY_DISP_BITMAP_S: Define the overlay bitmap.
- VPS_OVERLAY_DISPLAY_S: Define one display of overlay.
- VPS_OVERLAY_GROUP_S: Define VPS overlay group attribute.
- VPS_PROPERTY_TYPE_E: Define VPS property type.
- VPS_RESIZE_METHOD_E: Define VPS resize method.
- VPS_YUV_COLOR_MODE_E: Define YUV color mode.
- VPS_HAE_MULTICORE_MODE_E: Define HAE multi-core work mode.
- VPS_PROPERTY_S: Define VPS property.

ES_VPS_MAX_GRP_NUM

【Description】

Define the max count of VPS group.

【Define】

```
#define ES_VPS_MAX_GRP_NUM 256
```

ES_VPS_MAX_CHN_NUM

【Description】

Define the max count of VPS channel.

【Define】

```
#define ES_VPS_MAX_CHN_NUM 3
```

VPS_MIN_LINE_WIDTH

【Description】

Define the min width of line.

【Define】

```
#define VPS_MIN_LINE_WIDTH 2
```

VPS_MAX_LINE_WIDTH

【Description】

Define the max width of line.

【Define】

```
#define VPS_MAX_LINE_WIDTH 16
```

VPS_OVERLAY_MAX_LINE_NUM

【Description】

Define the max number of lines.

【Define】

```
#define VPS_OVERLAY_MAX_LINE_NUM 8
```

VPS_OVERLAY_MAX_QUADRANGLE_NUM

【Description】

Define the max number of quadrangles.

【Define】

```
#define VPS_OVERLAY_MAX_QUADRANGLE_NUM 8
```

VPS_OVERLAY_MAX_DISPLAY_NUM

【Description】

Define the max display number of overlay.

【Define】

```
#define VPS_OVERLAY_MAX_DISPLAY_NUM 4
```

VPS_CSC_TABLE_SIZE

【Description】

Define the size of CSC table.

【Define】

```
#define VPS_CSC_TABLE_SIZE 12
```

VPS_GRP

【Description】

Define VPS Group ID.

【Define】

```
typedef ES_S32 VPS_GRP;
```

DIE_IDX

【Description】

Define the DIE index.

【Define】

```
typedef ES_S32 DIE_IDX;
```

VPS_CHN

【Description】

Define VPS Channel ID.

【Define】

```
typedef ES_S32 VPS_CHN;
```

VPS_OVERLAY_HANDLE

【Description】

Define VPS overlay handle.

【Define】

```
typedef ES_VOID* VPS_OVERLAY_HANDLE;
```

VPS_MOD_PARAM_S

Members of data structure		
ES_U32	vpsVbSource	0: Common VB pool. 2: User VB pool.

HWTYPE_E

Values of enum	
HW_TYPE_HAE	HAE
HW_TYPE_3D	3D
HW_TYPE_DWE	DWE
HW_TYPE_VSE	VSE
HW_TYPE_BUTT	Invalid

ROTATION_E

Values of enum	
ROTATION_0	0
ROTATION_90	90
ROTATION_180	180
ROTATION_270	270
HW_TYPE_BUTT	Invalid

VPS_BLEND_MODE_E

Values of enum	
VPS_BLEND_CLEAR	0, 0
VPS_BLEND_SRC	1, 0
VPS_BLEND_DST	0, 1
VPS_BLEND_SRC_OVER_DST	1, 1 – Asrc
VPS_BLEND_DST_OVER_SRC	1 – Adst, 1
VPS_BLEND_SRC_IN_DST	Adst, 0
VPS_BLEND_DST_IN_SRC	0, Asrc
VPS_BLEND_SRC_OUT_DST	1 – Adst, 0
VPS_BLEND_DST_OUT_SRC	0, 1 – Asrc
VPS_BLEND_SRC_ATOP_DST	Adst, 1 – Asrc
VPS_BLEND_DST_ATOP_SRC	1 – Adst, Asrc
VPS_BLEND_ADD	1, 1
VPS_BLEND_SRC_XOR_DST	1 – Adst, 1 – Arc
VPS_BLEND_BUTT	Invalid

VPS_NORMALIZATION_MODE_E

Values of enum	
VPS_NORMALIZATION_MIN_MAX	Min-max
VPS_NORMALIZATION_Z_SCORE	Z score

VPS_RGB_S

Members of data structure		
ES_U32	r	Red
ES_U32	g	Green
ES_U32	b	Blue

VPS_NORMALIZATION_PARAMS_S

Members of data structure		
VPS_NORMALIZATION_MODE_E	normalizationMode	Normalization mode
VPS_RGB_S	minValue	Minimum value
VPS_RGB_S	maxMinReciprocal	The reciprocal of the difference of max minus min

Members of data structure		
VPS_RGB_S	stdReciprocal	The reciprocal of standard deviation
VPS_RGB_S	meanValue	Mean value
ES_U32	stepReciprocal	The reciprocal of step
ES_BOOL	bByPassQuantization	Quantization Enable

VPS_DEWARP_MODEL_E

Values of enum	
VPS_DEWARP_MODEL_LENS_DISTORTION_CORRECTION	Lens distortion correction.
VPS_DEWARP_MODEL_FISHEYE_EXPAND	Fisheye expand
VPS_DEWARP_MODEL_SPLIT_SCREEN	Split screen.
VPS_DEWARP_MODEL_FISHEYE_DEWARP	Fisheye correction
VPS_DEWARP_MODEL_PERSPECTIVE	Perspective.
VPS_DEWARP_MODEL_BYPASS	By pass.
VPS_DEWARP_MODEL_BUTT	Invalid

VPS_DEWARP_DISTORTION_MAP_S

Members of data structure		
ES_U32	userMapSize	0: Set camera matrix, calculate map at driver. !0: Set distortion map, calculate map by user.
ES_DOUBLE	cameraMatrix[9]	Camera Matrix
ES_DOUBLE	perspectiveMatrix[9]	Perspective Matrix
ES_DOUBLE	distortionCoeff[8]	Distortion Coefficient
ES_U32*	pUserMap	The address of user map file.

VPS_DEWARP_FOV_SETTING_S

Members of data structure		
ES_DOUBLE	offAngleUL	Offset Angle of 4
ES_DOUBLE	offAngleUR	Offset Angle of 1
ES_DOUBLE	offAngleDL	Offset Angle of 3
ES_DOUBLE	offAngleDR	Offset Angle of 2
ES_DOUBLE	fovUL	Field Of View of 4
ES_DOUBLE	fovUR	Field Of View of 1

Members of data structure		
ES_DOUBLE	fovDL	Field Of View of 3
ES_DOUBLE	fovDR	Field Of View of 2
ES_S32	panoAtWin	Panorama At Window
ES_DOUBLE	centerOffsetRatioUL	Center Offset Ratio of 4
ES_DOUBLE	centerOffsetRatioUR	Center Offset Ratio of 1
ES_DOUBLE	centerOffsetRatioDL	Center Offset Ratio of 3
ES_DOUBLE	centerOffsetRatioDR	Center Offset Ratio of 2
ES_DOUBLE	circleOffsetRatioUL	Circle Offset Ratio of 4
ES_DOUBLE	circleOffsetRatioUR	Circle Offset Ratio of 1
ES_DOUBLE	circleOffsetRatioDL	Circle Offset Ratio of 3
ES_DOUBLE	circleOffsetRatioDR	Circle Offset Ratio of 2

Below figure shows the concepts of Circle Offset Ratio, Offset Angle, Center Offset Ratio and Field Of View.

- 1 stands for UR, 2 stands for DR, 3 stands for DL, 4 stands for UL
- 0 <= Center Offset Ratio < 1
- 0 <= Circle Offset Ratio < 1
- Center Offset Ratio + Circle Offset Ratio < 1

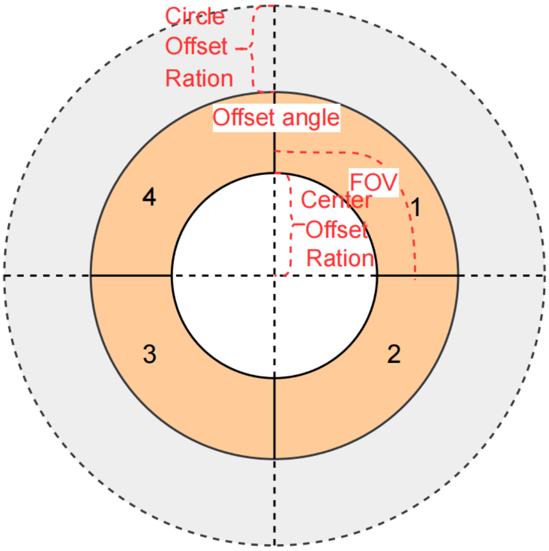


Figure 5-1 FOV parameters

VPS_DEWARP_PARAMS_S

Members of data structure		
ES_BOOL	bEnable	Dewarp Enable

Members of data structure		
VPS_DEWARP_MODEL_E	dewarpModel	Dewarp model
VPS_DEWARP_DISTORTION_MAP_S	map	Distortion Map
VPS_DEWARP_FOV_SETTING_S	fov	FOV setting
ES_DOUBLE	scaleFactor	Upscale factor: [1.0, 4.0]
ES_U32	splitHorizonLine	Horizon split line
ES_U32	splitVerticalLineUp	The up of Vertical split line
ES_U32	splitVerticalLineDown	The down of Vertical split line
ES_U32	roiX	The X of ROI
ES_U32	roiY	The Y of ROI

VPS_GRP_ATTR_S

Members of data structure		
ES_U32	maxW	Max width of source image.
ES_U32	maxH	Max height of source image.
PIXEL_FORMAT_E	pixelFormat	Pixel format of source image.
DYNAMIC_RANGE_E	dynamicRange	Dynamic range of source image. Reserved, not used.
FRAME_RATE_CTRL_S	frameRate	Group frame rate control.

VPS_CROP_COORDINATE_E

Values of enum	
VPS_CROP_RATIO_COOR	Ratio coordinate.
VPS_CROP_ABS_COOR	Absolute coordinate.

VPS_CROP_INFO_S

Members of data structure		
ES_BOOL	bEnable	CROP enable.
VPS_CROP_COORDINATE_E	cropCoordinate	Coordinate mode of the crop start point.
RECT_S	cropRect	CROP rectangular.

VPS_CHN_FRAME_STATUS_S

Members of data structure		
ES_U32	chnDropFrmCnt	The count of frame VPS process success but dropped by channel
ES_U32	chnOutputFrmCnt	The count of frame VPS process success and push to output queue.
ES_U32	chnFailFrmCnt	The count of frame VPS process failed.
ES_U32	chnConsumedFrmCnt	The count of user get frames from the channel.

VPS_QUERY_STATUS_S

Members of data structure		
ES_U32	grpRecvFrmCnt	The count of frames that group received.
ES_U32	grpRejectFrmCnt	The count of frames that rejected by input queue full
ES_U32	grpAcceptFrmCnt	The count of frames that group accepted
ES_U32	grpDropFrmCnt	The count of frames that group accepted but finally dropped
ES_U32	grpFailFrmCnt	The count of frames that group accept failed frame count
VPS_CHN_FRAME_STATUS_S	chnFrmStatus[ES_VPS_MAX_CHN_NUM]	The statistics of VPS channel processing

VPS_OUT_ATTR_S

Members of data structure		
ES_BOOL	bEnable	Output Enable
PIXEL_FORMAT_E	pixelFormat	The pixel format of output
ES_U32	width	The width of output
ES_U32	height	The height of output

VPS_MULTI_OUT_ATTR_S

Members of data structure		
VPS_OUT_ATTR_S	outAttr[ES_VPS_MAX_CHN_NUM]	The attribute of one in multiple out

VPS_CHN_ATTR_S

Members of data structure		
ES_U32	width	Width of target image.
ES_U32	height	Height of target image.
VIDEO_FORMAT_E	videoFormat	Video format of target image.
PIXEL_FORMAT_E	pixelFormat	Pixel format of target image.
DYNAMIC_RANGE_E	dynamicRange	Dynamic Range of target image. Reserved, not used.
COMPRESS_MODE_E	compressMode	Compression mode of target image. Reserved, not used.
FRAME_RATE_CTRL_S	frameRate	Frame rate control info
ES_BOOL	bMirror	Mirror enable.
ES_BOOL	bFlip	Flip enable.
ES_U32	depth	Range: [0, 8]; User get list depth.
ASPECT_RATIO_S	aspectRatio	Aspect Ratio info.
HWTYPE_E	resizeHwType	Appoint the hardware to execute resize work.
HWTYPE_E	mirrorFlipHwType	Appoint the hardware to execute mirror and/or flip work.
HWTYPE_E	aspectHwType	Appoint the hardware to execute aspect work.

VPS_NORMALIZATION_INFO_S

Members of data structure		
ES_BOOL	bEnable	Normalization enable
VPS_NORMALIZATION_PARAMS_S	param	The parameters of Normalization

VPS_OVERLAY_DISP_LINE_S

Members of data structure		
POINT_S	startPoint	Start point of line. When use HAE, x and y must multiple of 2
POINT_S	endPoint	End point of line. When use HAE, x and y must multiple of 2
ES_U32	lineWidth	[VPS_MIN_LINE_WIDTH, VPS_MAX_LINE_WIDTH]. When use HAE, must multiple of 2
ES_U32	color	ARGB color.

VPS_OVERLAY_DISP_LINE_GRP_S

Members of data structure		
ES_U8	lineCnt	The valid count of lines, [0, VPS_OVERLAY_MAX_LINE_NUM]
VPS_OVERLAY_DISP_LINE_S	line[VPS_OVERLAY_MAX_LINE_NUM]	The attributes of lines

VPS_OVERLAY_DISP_QUADRANGLE_S

Members of data structure		
RECT_S	rect	Used when type is VPS_OVERLAY_TYPE_RECT. When use HAE, x, y, width, height must multiple of 2
POINT_S	points[4]	Used when type is VPS_OVERLAY_TYPE_POLYGON, clockwise
ES_U32	lineWidth	[VPS_MIN_LINE_WIDTH, VPS_MAX_LINE_WIDTH]. When use HAE, must multiple of 2
ES_U32	color	ARGB color.

VPS_OVERLAY_TYPE_E

Values of enum	
VPS_OVERLAY_TYPE_LINE	Line
VPS_OVERLAY_TYPE_RECT	Rectangle
VPS_OVERLAY_TYPE_POLYGON	Polygon
VPS_OVERLAY_TYPE_MOSAIC	Mosaic
VPS_OVERLAY_TYPE_BITMAP	Bitmap
VPS_OVERLAY_TYPE_BUTT	Invalid

VPS_OVERLAY_DISP_QUADRANGLE_GRP_S

Members of data structure		
ES_U8	quadrangleCnt	The valid count of quadrangles, [0, VPS_OVERLAY_MAX_QUADRANGLE_NUM]
VPS_OVERLAY_DISP_QUADRANGLE_S	quadrangle[VPS_OVERLAY_MAX_QUADRANGLE_NUM]	The attributes of quadrangles

Concave and convex polygons are not currently supported.

VPS_MOSAIC_BLK_SIZE_E

Values of enum	
VPS_MOSAIC_BLK_SIZE_8	8x8
VPS_MOSAIC_BLK_SIZE_16	16x16
VPS_MOSAIC_BLK_SIZE_32	32x32
VPS_MOSAIC_BLK_SIZE_64	64x64
VPS_MOSAIC_BLK_SIZE_BUTT	Invalid

Mosaic are not currently supported.

VPS_OVERLAY_DISP_MOSAIC_S

Members of data structure		
RECT_S	rect	The rectangle of mosaic
VPS_MOSAIC_BLK_SIZE_E	blkSize	The block size of mosaic

VPS_OVERLAY_DISP_BITMAP_S

Members of data structure		
PIXEL_FORMAT_E	format	pixel format of bitmap
ES_S32	alpha	global alpha for bitmap[-1, 255], -1: disable, 0: transparent, 255: opaque
ES_U16	x	postion x in pixel, must align to 16 pixel
ES_U16	y	postion y in pixel, must be even number
ES_U16	w	bmp width in pixel, must align to 16 pixel
ES_U16	h	bmp height in pixel, must be even number
ES_U64	fd	the fd of bmp buffer
ES_U32	size	bmp image size(byte)
ES_U32	stride	stride must be 256 align

VPS_OVERLAY_DISPLAY_S

Members of data structure		
VPS_OVERLAY_TYPE_E	type	Indicate the type of current overlay display
HWTYPE_E	hwType	Indicate the hardware type of current overlay display, not effected after grp started
VPS_OVERLAY_DISP_LINE_GRP_S	lineGrp	Overlay display is line

Members of data structure		
VPS_OVERLAY_DISP_QUADRANGLE_GRP_S	quadrangleGrp	Overlay display is quadrangle. Not support now.
VPS_OVERLAY_DISP_MOSAIC_S	mosaic	Overlay display is mosaic. Not support now.
VPS_OVERLAY_DISP_BITMAP_S	bitmap	Overlay display is bitmap

VPS_OVERLAY_GROUP_S

Members of data structure		
ES_U32	dispCnt	The valid count of displays, [0, VPS_OVERLAY_MAX_DISPLAY_NUM]
VPS_OVERLAY_DISPLAY_S	disp[VPS_OVERLAY_MAX_DISPLAY_NUM]	The attributes of overlay displays

VPS_PROPERTY_TYPE_E

Values of enum	
VPS_PROPERTY_RESIZE_METHOD	Set/Get resize method
VPS_PROPERTY_ARRAY_CSC_YUV_TO_RGB	Set/Get CSC Array of YUV to RGB
VPS_PROPERTY_ARRAY_CSC_RGB_TO_YUV	Set/Get CSC Array of RGB to YUV
VPS_PROPERTY_YUV_COLOR_MODE_SRC	Set/Get the YUV color mode of source image
VPS_PROPERTY_YUV_COLOR_MODE_DST	Set/Get the YUV color mode of destination image
VPS_PROPERTY_HAE_MULTICORE_MODE	Set/Get the Multi-core work mode of HAE.

VPS_RESIZE_METHOD_E

Values of enum	
VPS_RESIZE_STRETCHBLIT	Bresenham algorithm is used to generate corresponding coordinates for fast stretching. Fastest speed with lower quality. This is default value.
VPS_RESIZE_FILTERBLIT	Using an FIR re-sampling filter, block size is 5x5. Slower speed with high quality scaling up or down.
VPS_RESIZE_BILINEAR	Using Bilinear algorithm
VPS_RESIZE_BICUBIC	Using Bicubic algorithm
VPS_RESIZE_BUTT	Invalid

VPS_YUV_COLOR_MODE_E

Values of enum	
VPS_COLOR_MODE_BT601	BT601
VPS_COLOR_MODE_BT709	BT709, VPS default value
VPS_COLOR_MODE_BT2020	BT2020
VPS_COLOR_MODE_BT601_UNLIMITED	BT601 unlimited
VPS_COLOR_MODE_BT709_UNLIMITED	BT709 unlimited
VPS_COLOR_MODE_BT2020_UNLIMITED	BT2020 unlimited
VPS_COLOR_MODE_USER_DEFINED	User defined CSC coefficient unlimited
VPS_COLOR_MODE_USER_DEFINED_CLAMP	Clamped User defined CSC coefficient

VPS_HAE_MULTICORE_MODE_E

Values of enum	
VPS_HAE_MULTICORE_COMBINED_MODE	Combined mode, VPS default.
VPS_HAE_MULTICORE_INDEPENDENT_MODE_CORE0	Core 0 independent.
VPS_HAE_MULTICORE_INDEPENDENT_MODE_CORE1	Core 1 independent.

VPS_PROPERTY_S

Members of data structure		
VPS_PROPERTY_TYPE_E	type	The type of VPS property.
VPS_RESIZE_METHOD_E	resizeMethod	The method of resize
ES_S32	cscTable[VPS_CSC_TABLE_SIZE]	CSC table
VPS_YUV_COLOR_MODE_E	colorMode	YUV color mode
VPS_HAE_MULTICORE_MODE_E	haeMultiCoreMode	The work mode of HAE

6. Error Code

The following table lists the VPS-related APIs error codes.

Table 6-1 VPS API error code

Error Code	Macro Definition	Description
0XA0046001	ES_ERR_VPS_INVALID_GRPID	VPS group ID is invalid.

Error Code	Macro Definition	Description
0XA0046002	ES_ERR_VPS_INVALID_CHNID	VPS channel ID is invalid.
0XA0046003	ES_ERR_VPS_ILLEGAL_PARAM	VPS parameter setting is invalid.
0XA0046004	ES_ERR_VPS_EXIST	VPS group is already existed.
0XA0046005	ES_ERR_VPS_UNEXIST	VPS group is not existed.
0XA0046008	ES_ERR_VPS_NOT_SUPPORT	Operation is not supported.
0XA0046009	ES_ERR_VPS_NOT_PERM	Operation is not permitted.
0XA004600C	ES_ERR_VPS_NOMEM	Failed to allocate memory.
0XA004600D	ES_ERR_VPS_NOBUF	Failed to allocate buffer pool.
0XA004600E	ES_ERR_VPS_BUF_EMPTY	Image queue is empty.
0XA004600F	ES_ERR_VPS_BUF_FULL	Image queue is full.
0XA0046010	ES_ERR_VPS_NOTREADY	VPS is not initialized.
0XA0046012	ES_ERR_VPS_BUSY	VPS is busy.

7. VPS proc debugging information and description

The debugging information uses the proc file system in Linux to reflect the running status of the current system in real time. The recorded information can be used for issue location and analysis.

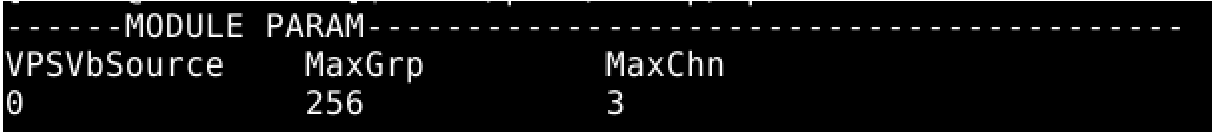
7.1 Method

Use the cat command on the console to review the information while the VPS-related program is running, such as cat /proc/esmap/vps.

7.2 Debugging information and description

7.2.1 MODULE PARAM

VPS module parameters.

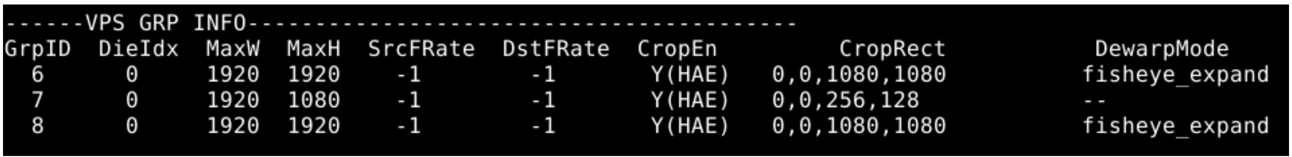


```
------MODULE PARAM-----------------------------------------
VPSVbSource    MaxGrp         MaxChn
0              256            3
```

Information Type	Parameter	Description
MODULE PARAM	VPSVbSource	The type of video buffer pool
	MaxGrp	The max count of VPS group
	MaxChn	The max count of channel in one VPS group

7.2.2 VPS GRP INFO

VPS group information.

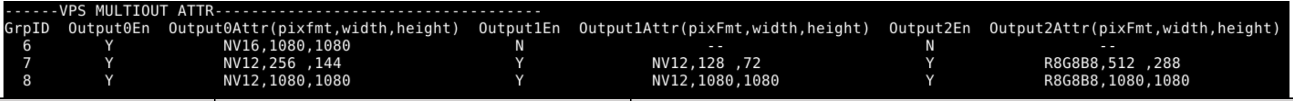


```
------VPS GRP INFO----------------------------------------
GrpID  DieIdx  MaxW  MaxH  SrcFRate  DstFRate  CropEn       CropRect          DewarpMode
  6      0     1920  1920    -1         -1      Y(HAE)   0,0,1080,1080        fisheye_expand
  7      0     1920  1080    -1         -1      Y(HAE)   0,0,256,128          --
  8      0     1920  1920    -1         -1      Y(HAE)   0,0,1080,1080        fisheye_expand
```

Information Type	Parameter	Description
VPS GRP INFO	GrpID	Group ID
	DieIdx	The DIE ID of this group
	MaxW	The max width of input in this group
	MaxH	The max height of input in this group
	SrcFRate	The source frame rate of group
	DstFRate	The destination frame rate of group
	CropEn	Group crop enable
	CropRect	The rectangle of group crop
	DewarpMode	The dewarp mode of group

7.2.3 VPS MULTIOUT ATTR

VPS one in multiple out attribute.

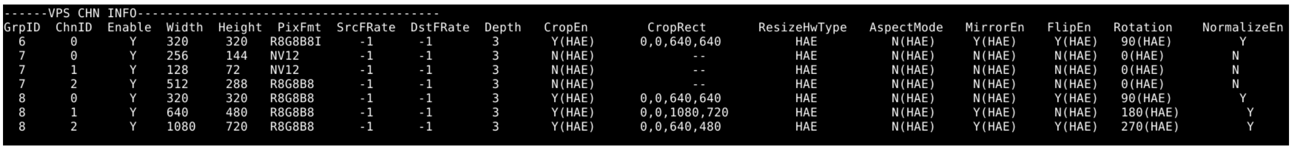


```
------VPS MULTIOUT ATTR-----------------------------------
GrpID  Output0En  Output0Attr(pixfmt,width,height)  Output1En  Output1Attr(pixFmt,width,height)  Output2En  Output2Attr(pixFmt,width,height)
  6        Y            NV16,1080,1080                  N                   --                        N                  --
  7        Y            NV12,256 ,144                   Y             NV12,128 ,72                    Y            R8G8B8,512 ,288
  8        Y            NV12,1080,1080                  Y             NV12,1080,1080                  Y            R8G8B8,1080,1080
```

Information Type	Parameter	Description
VPS MULTIOUT ATTR	GrpID	Group ID
	Output0En	Channel 0 output enable
	Output0Attr(pixfmt,width,height)	Channel 0 output attribute(pixel format, width, height)
	Output1En	Channel 1 output enable
	Output1Attr(pixfmt,width,height)	Channel 1 output attribute(pixel format, width, height)
	Output2En	Channel 2 output enable
	Output2Attr(pixfmt,width,height)	Channel 2 output attribute(pixel format, width, height)

7.2.4 VPS CHN INFO

VPS channel information.

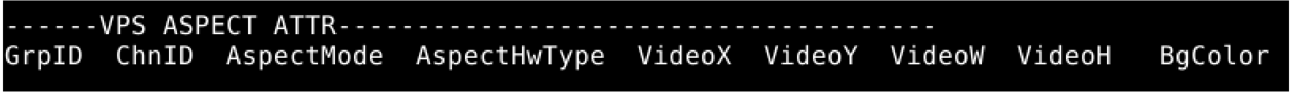


```
------VPS CHN INFO------------------------------------------
GrpID  ChnID  Enable  Width  Height  PixFmt   SrcFRate  DstFRate  Depth   CropEn      CropRect      ResizeHwType   AspectMode  MirrorEn  FlipEn   Rotation    NormalizeEn
  6      0       Y     320     320   R8G8B8I     -1        -1        3     Y(HAE)    0,0,640,640        HAE          N(HAE)    Y(HAE)   Y(HAE)   90(HAE)         Y
  7      0       Y     256     144   NV12        -1        -1        3     N(HAE)        --             HAE          N(HAE)    N(HAE)   N(HAE)   0(HAE)         N
  7      1       Y     128     72    NV12        -1        -1        3     N(HAE)        --             HAE          N(HAE)    N(HAE)   N(HAE)   0(HAE)         N
  7      2       Y     512     288   R8G8B8      -1        -1        3     N(HAE)        --             HAE          N(HAE)    N(HAE)   N(HAE)   0(HAE)         N
  8      0       Y     320     320   R8G8B8      -1        -1        3     Y(HAE)    0,0,640,640        HAE          N(HAE)    N(HAE)   Y(HAE)   90(HAE)         Y
  8      1       Y     640     480   R8G8B8      -1        -1        3     Y(HAE)    0,0,1080,720       HAE          N(HAE)    Y(HAE)   N(HAE)   180(HAE)         Y
  8      2       Y     1080    720   R8G8B8      -1        -1        3     Y(HAE)    0,0,640,480        HAE          N(HAE)    Y(HAE)   Y(HAE)   270(HAE)         Y
```

Information Type	Parameter	Description
VPS CHN INFO	GrpID	Group ID
	ChnId	Channel ID
	Enable	Channel enable
	Width	Channel output width
	Height	Channel output height
	PixFmt	Channel output pixel format
	SrcFRate	Channel output source frame rate
	DstFRate	Channel output destination frame rate
	Depth	The depth of channel output queue
	CropEn	Channel crop enable
	CropRect	Channel crop rectangle
	ResizeHwType	The hardware type of channel resize
	AspectMode	The mode of channel aspect
	MirrorEn	Channel mirror enable
	FlipEn	Channel flip enable
	Rotation	The angle of channel rotation
	NormalizeEn	Channel normalize enable

7.2.5 VPS ASPECT ATTR

VPS aspect attribute.



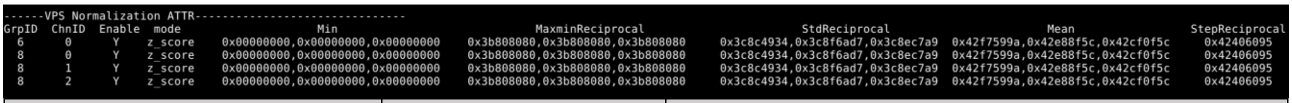
```
------VPS ASPECT ATTR-------------------------------------
GrpID  ChnID  AspectMode  AspectHwType  VideoX  VideoY  VideoW  VideoH   BgColor
```

Information Type	Parameter	Description
VPS ASPECT ATTR	GrpID	Group ID
	ChnId	Channel ID
	AspectMode	Aspect mode
	AspectHwType	The hardware type of aspect
	VideoX	The coordinate of X
	VideoY	The coordinate of Y

Information Type	Parameter	Description
	VideoW	The width of video
	VideoH	The height of video
	BgColor	The color of background

7.2.6 VPS NORMALIZATION ATTR

VPS normalization attribute.

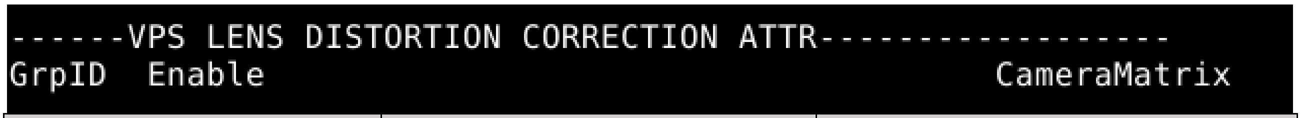


```
------VPS Normalization ATTR------------------------------
GrpID  ChnID  Enable  mode                  Min                              MaxminReciprocal                       StdReciprocal                            Mean                      StepReciprocal
  6      0      Y    z_score    0x00000000,0x00000000,0x00000000    0x3b808080,0x3b808080,0x3b808080     0x3c8c4934,0x3c8f6ad7,0x3c8ec7a9  0x42f7599a,0x42e88f5c,0x42cf0f5c     0x42406095
  8      0      Y    z_score    0x00000000,0x00000000,0x00000000    0x3b808080,0x3b808080,0x3b808080     0x3c8c4934,0x3c8f6ad7,0x3c8ec7a9  0x42f7599a,0x42e88f5c,0x42cf0f5c     0x42406095
  8      1      Y    z_score    0x00000000,0x00000000,0x00000000    0x3b808080,0x3b808080,0x3b808080     0x3c8c4934,0x3c8f6ad7,0x3c8ec7a9  0x42f7599a,0x42e88f5c,0x42cf0f5c     0x42406095
  8      2      Y    z_score    0x00000000,0x00000000,0x00000000    0x3b808080,0x3b808080,0x3b808080     0x3c8c4934,0x3c8f6ad7,0x3c8ec7a9  0x42f7599a,0x42e88f5c,0x42cf0f5c     0x42406095
```

Information Type	Parameter	Description
VPS NORMALIZATION ATTR	GrpID	Group ID
	ChnId	Channel ID
	Enable	Normalization enable
	mode	Normalization mode
	Min	Minimum value
	MaxminReciprocal	The reciprocal of the difference of max minus min
	StdReciprocal	The reciprocal of standard deviation
	Mean	Mean value
	StepReciprocal	The reciprocal of step

7.2.7 VPS LENS DISTORTION CORRECTION ATTR

VPS lens distortion correction attribute.

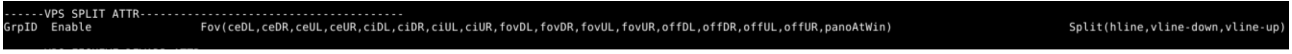


```
------VPS LENS DISTORTION CORRECTION ATTR-----------------
GrpID  Enable                                CameraMatrix
```

Information Type	Parameter	Description
VPS LENS DISTORTION CORRECTION ATTR	GrpID	Group ID
	Enable	Lens distortion correction enable
	CameraMatrix	Camera matrix

7.2.8 VPS SPLIT ATTR

VPS split attribute.

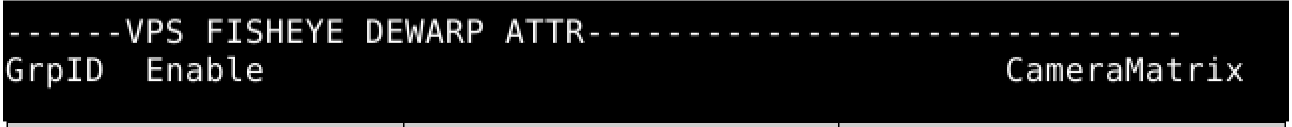


```
------VPS SPLIT ATTR--------------------------------------
GrpID  Enable              Fov(ceDL,ceDR,ceUL,ceUR,ciDL,ciDR,ciUL,ciUR,fovDL,fovDR,fovUL,fovUR,offDL,offDR,offUL,offUR,panoAtWin)                    Split(hline,vline-down,vline-up)
```

Information Type	Parameter	Description
VPS SPLIT ATTR	GrpID	Group ID
	Enable	Split enable
	Fov(ceDL,ceDR,ceUL,ceUR,ciDL,ciDR,ciUL,ciUR,fovDL,fovDR,fovUL,fovUR,offDL,offDR,offUL,offUR,panoAtWin)	FOV parameter
	Split(hline,vline-down,vline-up)	Split parameter

7.2.9 VPS FISHEYE DEWARP ATTR

VPS fisheye dewarp attribute.

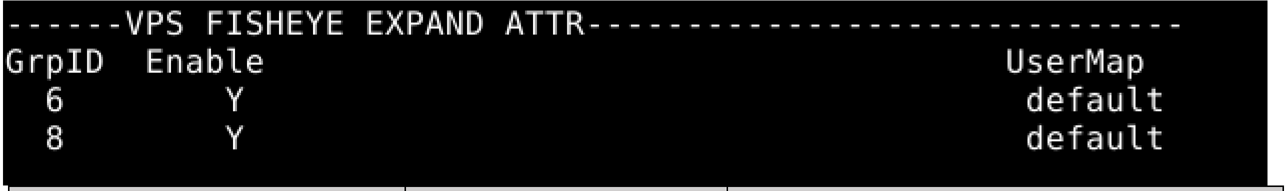


```
------VPS FISHEYE DEWARP ATTR------------------------------
GrpID  Enable                                   CameraMatrix
```

Information Type	Parameter	Description
VPS FISHEYE DEWARP ATTR	GrpID	Group ID
	Enable	Fisheye dewarp enable
	CameraMatrix	Camera matrix

7.2.10 VPS FISHEYE EXPAND ATTR

VPS fisheye expand attribute.

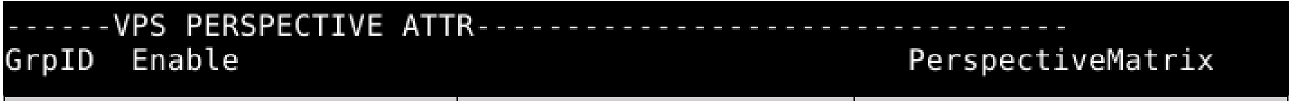


```
------VPS FISHEYE EXPAND ATTR------------------------------
GrpID  Enable                                   UserMap
  6        Y                                     default
  8        Y                                     default
```

Information Type	Parameter	Description
VPS FISHEYE EXPAND ATTR	GrpID	Group ID
	Enable	Fisheye expand enable
	UserMap	User map

7.2.11 VPS PERSPECTIVE ATTR

VPS perspective attribute.



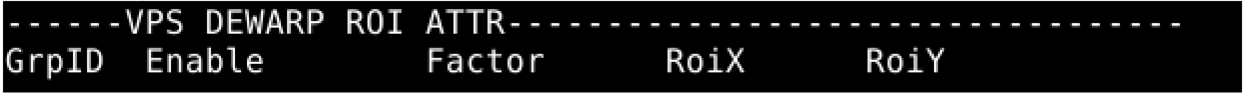
```
------VPS PERSPECTIVE ATTR------------------------------
GrpID  Enable                               PerspectiveMatrix
```

Information Type	Parameter	Description
VPS PERSPECTIVE ATTR	GrpID	Group ID
	Enable	Perspective enable

Information Type	Parameter	Description
	PerspectiveMatrix	Perspective Matrix

7.2.12 VPS DEWARP ROI ATTR

VPS dewarp ROI attribute.

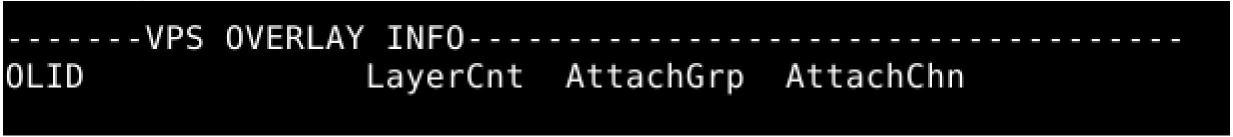


```
------VPS DEWARP ROI ATTR----------------------------------
GrpID  Enable        Factor      RoiX      RoiY
```

Information Type	Parameter	Description
VPS DEWARP ROI ATTR	GrpID	Group ID
	Enable	Dewarp ROI enable
	Factor	Factor
	RoiX	The X of ROI rectangle
	RoiY	The Y of ROI rectangle

7.2.13 VPS OVERLAY INFO

VPS overlay information.

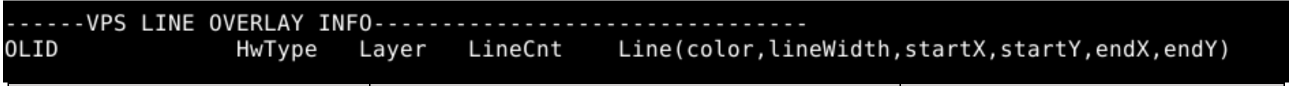


```
-------VPS OVERLAY INFO-----------------------------------
OLID             LayerCnt  AttachGrp  AttachChn
```

Information Type	Parameter	Description
VPS OVERLAY INFO	OLID	Overlay ID
	LayerCnt	The layer count of overlay
	AttachGrp	The group ID overlay attached to
	AttachChn	The channel ID overlay attached to

7.2.14 VPS LINE OVERLAY INFO

VPS line overlay information.



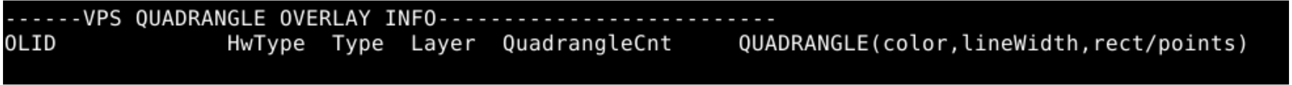
```
------VPS LINE OVERLAY INFO------------------------------
OLID             HwType   Layer   LineCnt    Line(color,lineWidth,startX,startY,endX,endY)
```

Information Type	Parameter	Description
VPS LINE OVERLAY INFO	OLID	Overlay ID
	HwType	The hardware type of draw line
	Layer	Layer ID

Information Type	Parameter	Description
	LineCnt	Line count
	Line(color,lineWidth,startX,startY,endX,endY)	The line attribute

7.2.15 VPS QUADRANGLE OVERLAY INFO

VPS quadrangle overlay information.

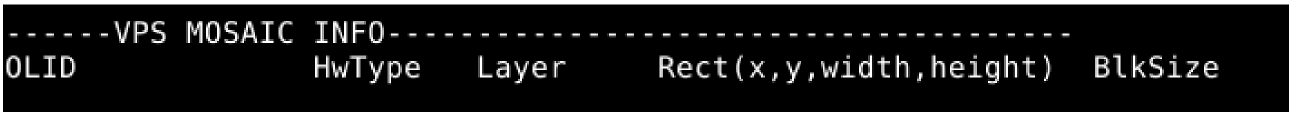


```
------VPS QUADRANGLE OVERLAY INFO-------------------------
OLID            HwType  Type  Layer  QuadrangleCnt     QUADRANGLE(color,lineWidth,rect/points)
```

Information Type	Parameter	Description
VPS QUADRANGLE OVERLAY INFO	OLID	Overlay ID
	HwType	The hardware type of draw quadrangle
	Type	The type of quadrangle
	Layer	Layer ID
	QuadrangleCnt	Quadrangle count
	QUADRANGLE(color,lineWidth,rect/points)	The quadrangle attribute

7.2.16 VPS MOSAIC INFO

VPS mosaic information.

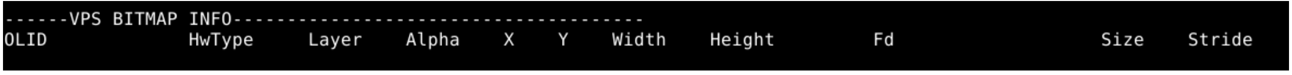


```
------VPS MOSAIC INFO------------------------------------
OLID            HwType   Layer     Rect(x,y,width,height)  BlkSize
```

Information Type	Parameter	Description
VPS MOSAIC INFO	OLID	Overlay ID
	HwType	The hardware type of draw mosaic
	Layer	Layer ID
	Rect(x,y,width,height)	The position of mosaic
	BlkSize	The block size of mosaic

7.2.17 VPS BITMAP INFO

VPS bitmap information.

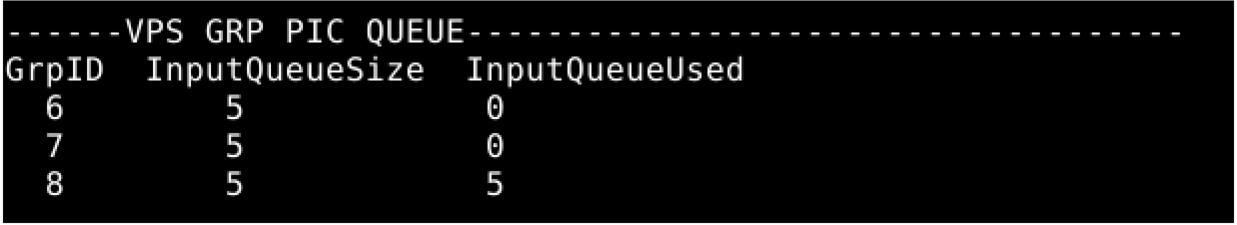


```
------VPS BITMAP INFO-------------------------------------
OLID            HwType     Layer     Alpha     X     Y     Width     Height         Fd                 Size     Stride
```

Information Type	Parameter	Description
VPS BITMAP INFO	OLID	Overlay ID
	HwType	The hardware type of bitmap
	Layer	Layer ID
	Alpha	The alpha value of bitmap
	X	The X of display coordinate of bitmap
	Y	The Y of display coordinate of bitmap
	Width	Bitmap width
	Height	Bitmap height
	Fd	Bitmap buffer fd
	Size	Bitmap size
	Stride	stride

7.2.18 VPS GRP PIC QUEUE

VPS group picture queue status.

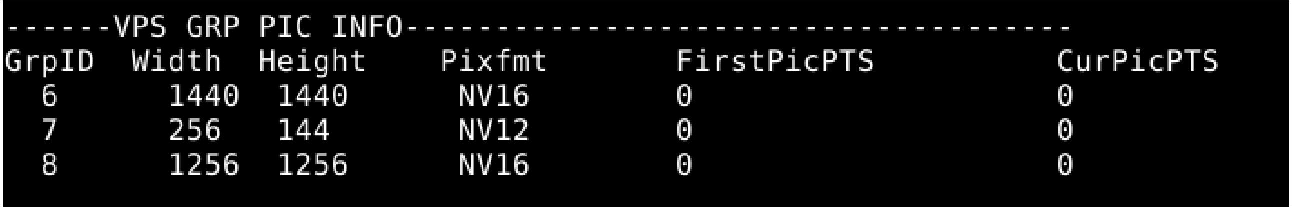


```
------VPS GRP PIC QUEUE-----------------------------------
GrpID  InputQueueSize  InputQueueUsed
  6        5             0
  7        5             0
  8        5             5
```

Information Type	Parameter	Description
VPS GRP PIC QUEUE	GrpID	Group ID
	InputQueueSize	The depth of input queue
	InputQueueUsed	The used depth of input queue

7.2.19 VPS GRP PIC INFO

VPS group picture information.

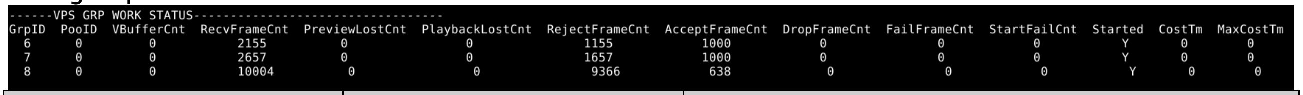


```
------VPS GRP PIC INFO------------------------------------
GrpID  Width  Height    Pixfmt      FirstPicPTS         CurPicPTS
  6      1440  1440      NV16        0                   0
  7      256   144       NV12        0                   0
  8      1256  1256      NV16        0                   0
```

Information Type	Parameter	Description
VPS GRP PIC INFO	GrpID	Group ID
	Width	The width of input picture
	Height	The height of input picture
	Pixfmt	The pixel format of input picture
	FirstPicPTS	The PTS of first picture
	CurPicPTS	The PTS of current picture

7.2.20 VPS GRP WORK STATUS

VPS group work status.

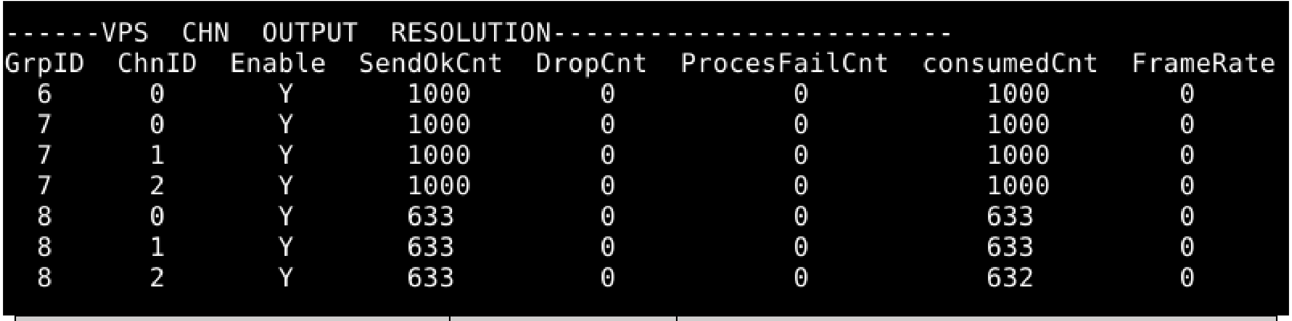


```
------VPS GRP WORK STATUS---------------------------------------
GrpID  PooID  VBufferCnt  RecvFrameCnt  PreviewLostCnt  PlaybackLostCnt  RejectFrameCnt  AcceptFrameCnt  DropFrameCnt  FailFrameCnt  StartFailCnt  Started  CostTm  MaxCostTm
  6      0         0           2155           0                0               1155           1000            0               0            0           Y       0        0
  7      0         0           2657           0                0               1657           1000            0               0            0           Y       0        0
  8      0         0           10004           0                0               9366           638             0               0            0           Y       0        0
```

Information Type	Parameter	Description
VPS GRP WORK STATUS	GrpID	Group ID
	PooID	The pool ID group used
	VBufferCnt	VB count
	RecvFrameCnt	The frame count group received
	PreviewLostCnt	The lost frame count in preview mode
	PlaybackLostCnt	The lost frame count in playback mode
	RejectFrameCnt	The frame count group rejected
	AcceptFrameCnt	The frame count group accepted
	DropFrameCnt	The frame count group dropped
	FailFrameCnt	The frame count group processed failed
	StartFailCnt	The count of started failed
	Started	The group is started or not
	CostTm	Total cost time
	MaxCostTm	The max cost time

7.2.21 VPS CHN OUTPUT RESOLUTION

VPS channel output information.

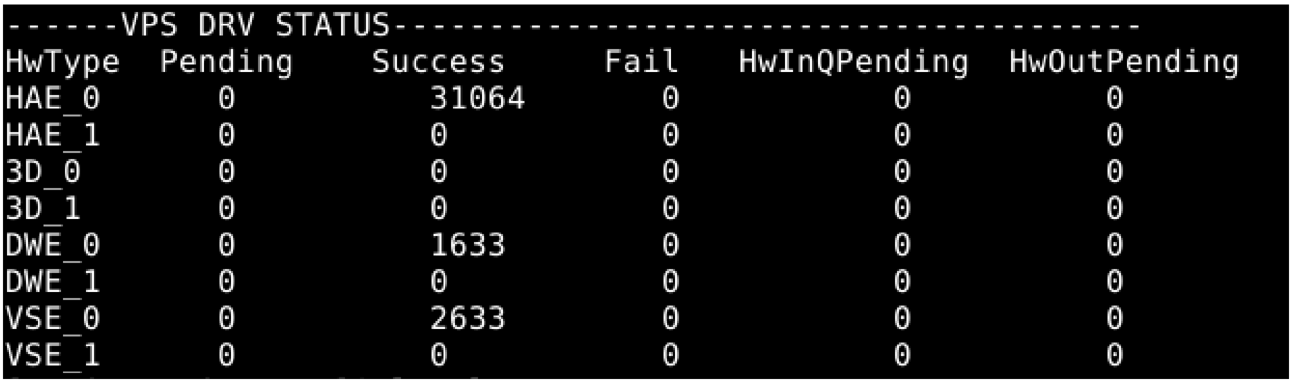


```
------VPS  CHN  OUTPUT  RESOLUTION------------------------
GrpID  ChnID  Enable  SendOkCnt  DropCnt  ProcesFailCnt  consumedCnt  FrameRate
  6      0       Y       1000       0           0           1000        0
  7      0       Y       1000       0           0           1000        0
  7      1       Y       1000       0           0           1000        0
  7      2       Y       1000       0           0           1000        0
  8      0       Y       633        0           0           633         0
  8      1       Y       633        0           0           633         0
  8      2       Y       633        0           0           632         0
```

Information Type	Parameter	Description
VPS CHN OUTPUT RESOLUTION	GrpID	Group ID
	ChnID	Channel ID
	Enable	Channel output enable
	SendOkCnt	The frame count of channel sent OK
	DropCnt	The frame count of channel dropped
	ProcesFailCnt	The frame count of channel processed failed
	consumedCnt	The frame count of user consumed
	FrameRate	The frame rate of channel output

7.2.22 VPS DRV STATUS

VPS driver status.



```
------VPS DRV STATUS--------------------------------------
HwType  Pending    Success     Fail   HwInQPending  HwOutPending
HAE_0      0          31064       0         0           0
HAE_1      0          0           0         0           0
3D_0       0          0           0         0           0
3D_1       0          0           0         0           0
DWE_0      0          1633        0         0           0
DWE_1      0          0           0         0           0
VSE_0      0          2633        0         0           0
VSE_1      0          0           0         0           0
```

Information Type	Parameter	Description
VPS DRV STATUS	HwType	The hardware type
	Pending	The frame count of pending
	Success	The frame count of process success
	Fail	The frame count of process failed

Information Type	Parameter	Description
	HwInQPending	The frame count in hardware queue which pending to process
	HwOutPending	The frame count in hardware queue which waiting for fetching out