INGENIC®
T31 FrameSource API Reference

Date: 2022-04
Viewer: Jason Xu

北京君正集成电路股份有限公司
Ingenic Semiconductor Co., Ltd.

INGENIC®

T31 FrameSource API Reference

Release history

Date	Revision	Change
2022-04	1.0	First release

Disclaimer

This documentation is provided for use with Ingenic products. No license to Ingenic property rights is granted. Ingenic assumes no liability, provides no warranty either expressed or implied relating to the usage, or intellectual property right infringement except as provided for by Ingenic Terms and Conditions of Sale.

Ingenic products are not designed for and should not be used in any medical or life sustaining or supporting equipment.

All information in this document should be treated as preliminary. Ingenic may make changes to this document without notice. Anyone relying on this documentation should contact Ingenic for the current documentation and errata.

Ingenic Semiconductor Co. Ltd

Add: Junzheng R&D Center, Phase II, Zhongguancun Software Park, Dongbeiwangxi Road, Haidian District, Beijing, China

Tel: 86-10-56345000

Fax: 86-10-56345001

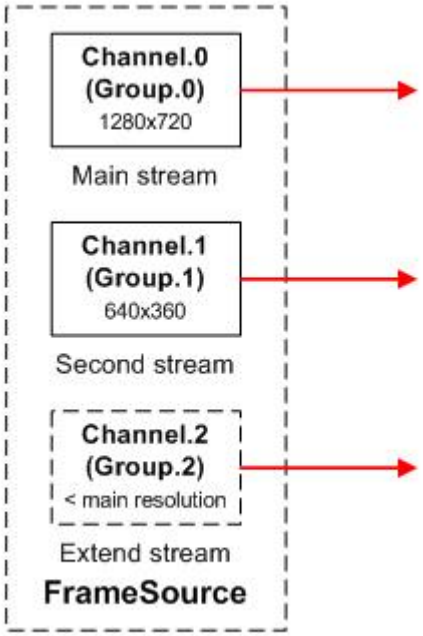
Http: //www.ingenic.com

Content

1 Description

1.1Function introduce

Video Source is the image source of IMP system, it includes functions such as setting the image resolution, cropping, scaling and other properties as well as the back-end noise reduction function. FrameSource is a data flow related to the concept, you can set the image resolution, format, etc., and provide the original image to the back-end. The scheme of FrameSource is as follow：



Pic 1-1 FrameSource

According to this image above, FrameSource has 3 outputs, all of them can be used for encoding

*Channel 0: High Definition (HD) video stream

*Channel 1: Standard Definition (SD) video stream, or IVS can only do the algorithm of this data source.

*Channel 2: Expands the channel, not recommended. It is used in special applications.

Note: the code stream of the extended Channel here is actually copied from the main code stream (Channel 0) or the secondary code stream (Channel 1). The purpose is to perform some other processing on the copied video, so as to achieve some functions (such as CSC) or parallel operation that are inconvenient for the main or secondary code stream. However, it is recommended that the size of the code stream in the expansion channel be not larger than or smaller than the size of the code channel to be copied. This will consume extra CPU and reduce CPU efficiency.

Pic 1-2 Framesource acquisition process

FrameSource is a software abstraction of ISP's mapping capabilities. Images with a certain resolution output by the Sensor are output after three levels of processing: fcrop, sclaer and crop. Fcrop is the first-level clipping function, sclaer is the scaling function, and crop is the post-level clipping function. These functions are mapped in the FrameSource channel parameters. Note: The difference between T30/T21 is that there is no Fcrop feature.

1.2Module operation process

1.2.1 Init process

1) Create a channel
2) Set the channel
3) Enable the channel

1.2.2 Exit process

1) Enable channel
2) Logout channel

The initialization step of FrameSource is as follows (take two output as an example)：

```
IMPFSChnAttr fs_chn_attr;
fs_chn_attr.pixFmt = PIX_FMT_NV12;
fs_chn_attr.outFrmRateNum = SENSOR_FRAME_RATE;
fs_chn_attr.outFrmRateDen = 1;
fs_chn_attr.nrVBs = 3;
fs_chn_attr.type = FS_PHY_CHANNEL;

fs_chn_attr.crop.enable = 1;
fs_chn_attr.crop.top = 0;
fs_chn_attr.crop.left = 0;
fs_chn_attr.crop.width = 960;
```

```
fs_chn_attr.crop.height = 640;

fs_chn_attr.scaler.enable = 1;
fs_chn_attr.scaler.outwidth = 320;
fs_chn_attr.scaler.outheight = 240;

ret = IMP_FrameSource_CreateChn(0, &fs_chn_attr);
//step.1 set up Channel 0
if (ret < 0) {
    printf("FrameSource_CreateChn(0) error\n");
    goto createchn_err;
}

ret = IMP_FrameSource_EnableChn(0);
//step.2 Enable Channel 0，channel 0Start outputting the image
if (ret < 0) {
    printf("EnableChn(0) error\n";
    return enablechn_err;
}

//At this point, FrameSource begins to pass the data to the back-end Group

ret = IMP_FrameSource_DisableChn(0);
//step.3 Disable channel 0，channel 0 Stop outputting the image
if (ret < 0) {
    printf("FrameSource_DisableChn(0) error\n");
return disablechn_err;
}

ret = IMP_FrameSource_DestroyChn(0); //step.4  destroy channel 0
if (ret < 0) {
    printf("FrameSource_DestroyChn error\n");
    return destorychn_err;
}
```

Referring to Samples for more methods.

2 API

2.1 FrameSource API

API Name	Function
IMP_FrameSource_CreateChn	Create a channel
IMP_FrameSource_DestroyChn	Destroy the channel
IMP_FrameSource_EnableChn	Enable the channel
IMP_FrameSource_DisableChn	Close the channel
IMP_FrameSource_SetSource	Specifies the extended channel source channel interface
IMP_FrameSource_GetChnAttr	Obtain channel properties
IMP_FrameSource_SetChnAttr	Set the channel properties
IMP_FrameSource_SetFrameDepth	Sets the maximum depth of the accessible image
IMP_FrameSource_GetFrameDepth	Maximum depth of the images acquired
IMP_FrameSource_GetFrame	Images acquired
IMP_FrameSource_GetTimedFrame	Get the image for the specified time
IMP_FrameSource_ReleaseFrame	Release the acquired image
IMP_FrameSource_SnapFrame	Get the image
IMP_FrameSource_SetMaxDelay	Sets the maximum number of delay frames
IMP_FrameSource_GetMaxDelay	Gets the maximum number of delay frames
IMP_FrameSource_SetDelay	Set the number of delay frames
IMP_FrameSource_GetDelay	Get the number of delay frames
IMP_FrameSource_SetChnFifoAttr	Sets the channel maximum cache FIFO attribute
IMP_FrameSource_GetChnFifoAttr	Gets the channel maximum cache FIFO attribute
IMP_FrameSource_SetPool	Bind chnnel to the memory pool where FrameSource application mem from pool application

IMP_FrameSource_GetPool	Get the poolID via the channel ID
IMP_FrameSource_ChnStatQuery	Get the creation status of the framesource channel via the channel ID

2.1.1 IMP_FrameSource_CreateChn

【Function】

Create a channel.

【Prototype】

int IMP_FrameSource_CreateChn(int chnNum, IMPFSChnAttr *chn_attr);

【Parameters】

Parameters	Description	Input/Output
chnNum	Id of frame channel	Input
chnAttr	the pointer of the frame channel's attribute	Input

【Return values】

0 means success; Other values means failure, its value is an error code.

【Dependencies】

Header file：imp_framesource.h

Library file：libimp.a / libimp.so

【NB】

Create a channel to provide a data source to the back-end module.

You can set the channel related attributes, including: the width of the picture, the height of the picture, the picture format, channel output frame rate, number of cache buf, cropping and scaling properties.

For T10, channels 0 and 1 can only be set as physical channels, and channels 2 and 3 can only be set as extended channels.

2.1.2 IMP_FrameSource_DestroyChn

【Function】

Destroy the channel.

【Grammar】

IMP_FrameSource_DestroyChn(int chnNum);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

If the program has called IMP_FrameSource_EnableChn, be sure to call IMP_FrameSource_DisableChn before using this function.

2.1.3 IMP_FrameSource_EnableChn

【Function】

Enable the channel.

【Grammar】

int IMP_FrameSource_EnableChn(int chnNum);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input

【Return value】

return 0 success； no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

Before using this function, you must ensure that the enabled channel has been created.

2.1.4 IMP_FrameSource_DisableChn

【Function】

Close the channel.

【Grammar】

int IMP_FrameSource_DisableChn(int chnNum);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input

【Return value】

return 0 success； no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

None.

2.1.5 IMP_FrameSource_SetSource

【Function】

Specify the extended channel source channel interface.

【Grammar】

int IMP_FrameSource_SetSource(int extchnNum, int sourcechnNum);

【Formal parameter】

Parameter name	Describe	Input/Output
extchnNum	Expand Channel number	Input
sourcechnNum	SourceChannel number	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

IMP_FrameSource_CreateChn is called after IMP_FrameSource_EnableChn.

2.1.6 IMP_FrameSource_GetChnAttr

【Function】

Get channel attributes.

【Grammar】

int IMP_FrameSource_GetChnAttr(int chnNum, IMPFSChnAttr *chnAttr);

【Formal parameter】

Parameter name	Describe	Input/Output

chnNum	Channel number	Input
chnAttr	Pointer to the channel property structure	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

You can obtain channel properties, including: image width, image height, image format, channel Output frame rate, cache BUF number, cropping and scaling properties.

2.1.7 IMP_FrameSource_SetChnAttr

【Function】

Setting channel Properties.

【Grammar】

int IMP_FrameSource_SetChnAttr(int chnNum,const IMPFSChnAttr *chnAttr);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
chnAttr	Pointer to the channel property structure	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

You can obtain channel properties, including: image width, image height, image format, channel Output frame rate, cache BUF number, cropping and scaling properties.

2.1.8 IMP_FrameSource_SetFrameDepth

【Function】

Sets the maximum depth of the image that can be retrieved.

【Grammar】

int IMP_FrameSource_SetFrameDepth(int chnNum, int depth);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
depth	Sets the maximum depth value of the image that can be obtained	Input

【Return value】

return 0 success； no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

This interface is used to set the number of video frames cached for a channel. When the user sets the cache of multi-frame video images, a certain number of continuous image data can be obtained.

If depth is set to 0, the system does not need to cache images for this channel. Therefore, users cannot obtain image data of this channel. The system does not cache images for channels by default, that is, depth is 0 by default.

The system will automatically update the oldest image data to ensure that users can get the latest image once they start to obtain it.

The system automatically stops caching new images because it cannot obtain images, and users cannot obtain new images. Therefore, users are advised to ensure that the access and release interfaces are used together.

The system will automatically update the oldest image data that the user has not obtained, and ensure that the image queue cached is the latest image. Because the user cannot guarantee the acquisition speed, the acquisition may not be continuous image;

This function can call location, no requirements. But you can only use it once.

2.1.9 IMP_FrameSource_GetFrameDepth

【Function】

Obtain the maximum depth of the image.

【Grammar】

int IMP_FrameSource_GetFrameDepth(int chnNum, int *depth);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
depth	Sets the maximum depth value of the image that can be obtained	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

None.

2.1.10 IMP_FrameSource_GetFrame

【Function】

Image acquired.

【Grammar】

int IMP_FrameSource_GetFrame(int chnNum, IMPFrameInfo **frame);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
frame	Image acquired	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

This interface can obtain the video image information of the specified channel. Image information mainly includes: image width, height, pixel format and image data starting address;

This interface takes effect only after the channel is enabled.

The interface can be obtained several times and then released. However, you are advised to pair the obtained and released interfaces.

The default timeout time of this interface is 2s, that is, if the image is still not obtained within 2s, it will be returned after timeout.

2.1.11 IMP_FrameSource_GetTimedFrame

【Function】

Gets the image at the specified time.

【Grammar】

int IMP_FrameSource_GetTimedFrame(int chnNum, IMPFrameTimestamp *framets, int block, void *framedata, IMPFrameInfo *frame);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
framets	Time information	Input
block	Blocking properties	Input
framedata	Memory pointer to copy image	Input
frame	Image information is obtained	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

This interface can obtain the video image information of the specified channel at the specified time. Image information mainly includes: image width, height, pixel format and image data;

This interface takes effect only after the channel is enabled.

This interface requires IMP_FrameSource_SetMaxDelay and IMP_FrameSource_SetDelay to be set.

2.1.12 IMP_FrameSource_ReleaseFrame

【Function】

Release the captured image.

【Grammar】

int IMP_FrameSource_ReleaseFrame(int chnNum, IMPFrameInfo *frame);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
frame	Release the captured image	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

None.

2.1.13 IMP_FrameSource_SnapFrame

【Function】

Get the image.

【Grammar】

int IMP_FrameSource_SnapFrame(int chnNum, IMPPixelFormat fmt, int width, int height, void *framedata, IMPFrameInfo *frame);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
fmt	Image format	Input
width	The width of the image	Input
height	Height of the image	Input
framedata	Memory pointer to copy image	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

This interface can get a frame of the specified format and size of the image;

The current format supports NV12, YUYV422;

The size is consistent with channel resolution;

Do not call the IMP_FrameSource_SetFrameDepth interface;

This interface is valid only after the channel is enabled.

2.1.14 IMP_FrameSource_SetMaxDelay

【Function】

Set the maximum number of frames to delay.

【Grammar】

int IMP_FrameSource_SetMaxDelay(int chnNum, int maxcnt);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
maxcnt	Maximum Delay, Unit Frame	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

Call IMP_FrameSource_CreateChn between IMP_FrameSource_EnableChn and IMP_FrameSource_CreateChn.

2.1.15 IMP_FrameSource_GetMaxDelay

【Function】

Gets the maximum number of frames delayed.

【Grammar】

int IMP_FrameSource_GetMaxDelay(int chnNum, int *maxcnt);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
maxcnt	Maximum Delay, Unit Frame	Output

【Return value】

return 0 success； no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

The IMP_FrameSource_CreateChn function is used after IMP_FrameSource_CreateChn.

2.1.16 IMP_FrameSource_SetDelay

【Function】

Set the number of frames delayed.

【Grammar】

int IMP_FrameSource_SetDelay(int chnNum, int cnt);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
cnt	Delay, unit frame	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

The IMP_FrameSource_SetMaxDelay function is called after IMP_FrameSource_SetMaxDelay.

2.1.17 IMP_FrameSource_GetDelay

【Function】

Gets the number of delayed frames.

【Grammar】

int IMP_FrameSource_GetDelay(int chnNum, int *cnt);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
cnt	Delay, unit frame	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

The IMP_FrameSource_CreateChn function is used after IMP_FrameSource_CreateChn.

2.1.18 IMP_FrameSource_SetChnFifoAttr

【Function】

Set the channel maximum cache FIFO properties.

【Grammar】

int IMP_FrameSource_SetChnFifoAttr(int chnNum, IMPFSChnFifoAttr *attr);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
attr	FIFO properties，Includes FIFO maximum depth, unit frame; FIFO type	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

Call IMP_FrameSource_CreateChn between IMP_FrameSource_EnableChn and IMP_FrameSource_CreateChn.

2.1.19 IMP_FrameSource_GetChnFifoAttr

【Function】

Gets the channel maximum cache FIFO Properties.

【Grammar】

int IMP_FrameSource_GetChnFifoAttr(int chnNum, IMPFSChnFifoAttr *attr);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
attr	FIFO properties	Output

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

The IMP_FrameSource_CreateChn function is used after IMP_FrameSource_CreateChn.

2.1.20 IMP_FrameSource_SetPool

【Function】

Bind the CHnnel to the memory pool, that is, apply for the MEM from the pool.

【Grammar】

int IMP_FrameSource_SetPool(int chnNum, int poolID);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input
poolID	Memory Pool Number	Input

【Return value】

return 0 success；no 0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

To solve RMEM fragmentation, bind the channel FrameSource to the corresponding mempool. FramSource applies meM in mempool. If not called, The FramSource will apply in the rmem and there will be a possibility of fragmentation for the rmem.

The ChannelId must be greater than or equal to 0 and smaller than 32.

2.1.21 IMP_FrameSource_GetPool

【Function】

Obtain the poolID by channel ID.

【Grammar】

int IMP_FrameSource_GetPool(int chnNum);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input

【Return value】

return >=0 && < 32 success；<0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

The poolId can be obtained through ChannelId, which is not needed by the customer.

2.1.22 IMP_FrameSource_ChnStatQuery

【Function】

Get the creation status of the Framesource channel by channel ID; 0: the channel is not created. 1: the channel is created. 2: the channel is running.

【Grammar】

int IMP_FrameSource_ChnStatQuery(int chnNum, IMPFSChannelState *pstate);

【Formal parameter】

Parameter name	Describe	Input/Output
chnNum	Channel number	Input

【Return value】

return >=0 success；<0 fail.

【Dependence】

Head file：imp_framesource.h

Lib file：libimp.a / libimp.so

【NB】

None.

3 Date Type

3.1 FrameSource Date Type

Name	Definition
IMPFSChnCrop	The structure of cropping operation.
IMPFSChnScaler	The structure of scaling operation.
IMPFSChnType	The type of the frame channel
IMPFSChnFifoType	Channel FIFO type
IMPFSChannelState	Channel State
IMPFSChnFifoAttr	Channel FIFO properties construct
IMPFSChnAttr	Channel property structure body

3.1.1 IMPFSChnCrop

【Explain】

Channel-cropped constructs.

【Definition】

```
typedef struct {
        int enable;
        int left;
        int top;
        int width;
        int height;
} IMPFSChnCrop;
```

【Member】

Member name	Describe
enable	Enables the cropping of the Function
left	Cut the left starting point
top	Cut the starting point
width	Picture cutting width
height	Image cutting height

【NB】

None.

3.1.2 IMPFSChnScaler

【Explain】

The Channel scales the structure.

【Definition】

```
typedef struct {
        int enable;
        int outwidth;
        int outheight;
```

```
} IMPFSChnScaler;
```

【Member】

Member name	Describe
enable	Enables scaling Function
outwidth	Picture width after zoom
outheight	Picture height after the zoom

【NB】

None.

3.1.3 IMPFSChnType

【Explain】

The Channel scales the structure.

【Definition】

```
typedef enum {
        FS_PHY_CHANNEL,
        FS_EXT_CHANNEL,
} IMPFSChnType;
```

【Member】

Member name	Describe
FS_PHY_CHANNEL	physical channel
FS_EXT_CHANNEL	Expand the channel

【NB】

None.

3.1.4 IMPFSChnFifoType

【Explain】

Channel FIFO type.

【Definition】

```
typedef enum {
        FIFO_CACHE_PRIORITY = 0,
} IMPFSChnFifoType;
```

【Member】

Member name	Describe
FIFO_CACHE_PRIORITY	FIFO first cache and then Output the data
FIFO_DATA_PRIORITY	FIFO prioritizes Output data and then caches

【NB】

None.

3.1.5 IMPFSChannelState

【Explain】

Channel state.

【Definition】

```
typedef enum {
   IMP_FSCHANNEL_STATE_CLOSE,
   IMP_FSCHANNEL_STATE_OPEN,
   IMP_FSCHANNEL_STATE_RUN,
} IMPFSChannelState;
```

【Member】

Member name	Describe
FSCHANNEL_STATE_CLOSE	The fs channel was not created or was closed for destruction
FSCHANNEL_STATE_OPEN	The fs channel was created and was not enabled
FSCHANNEL_STATE_RUN	The fs channel has been created and is enabled

【NB】

None.

3.1.6 IMPFSChnFifoAttr

【Explain】

Channel FIFO properties construct.

【Definition】

```
typedef struct {
        int maxdepth;
        IMPFSChnFifoType type;
} IMPFSChnFifoAttr;
```

【Member】

Member name	Describe
maxdepth	FIFO, Maximum depth
IMPFSChnFifoType type	Channel FIFO type

【NB】

None.

3.1.7 IMPFSChnAttr

【Explain】

The structure of frame channel's attributes.

【Definition】

```
typedef struct {
        int picWidth;
        int picHeight;
        IMPPixelFormat pixFmt;
        IMPFSChnCrop crop;
        IMPFSChnScaler scaler;
        int outFrmRateNum;
        int outFrmRateDen;
        int nrVBs;
```

```
        IMPFSChnType type;
        IMPFSChnCrop fcrop;
} IMPFSChnAttr;
```

【Member】

Member name	Describe
picWidth	Picture width
picHeight	Image height
IMPPixelFormat pixFmt	picture format
IMPFSChnCrop crop;	Image clipping properties
IMPFSChnScaler scaler;	Image scaling property
outFrmRateNum	The Output frame-rate molecules of the channel
outFrmRateDen	The Output frame-rate denominator of the channel
nrVBs	Video buffer quantity
IMPFSChnType type	channel type
IMPFSChnCrop fcrop	Image clipping properties

【NB】

None.